

# **Wide-Baseline Keypoint Detection and Matching with Wide-Angle Images for Vision Based Localisation**

Peter Ian Hansen

B.Eng. (mechanical) (Hons I)  
Queensland University of Technology

Faculty of Built Environment and Engineering  
School of Engineering Systems  
Queensland University of Technology  
GPO Box 2434, Brisbane, QLD 4001, Australia

Submitted as a requirement for the award of  
Doctor of Philosophy  
Queensland University of Technology

2010



# Keywords

Image processing, computer vision, keypoints, features, wide-angle, fisheye, catadioptric, scale-space, scale-invariant, wide-baseline, spherical diffusion, spherical Gaussian, visual odometry, localisation, visual SLAM, place recognition, loop-closure.





# Abstract

This thesis addresses the problem of detecting and describing the same scene points in different wide-angle images taken by the same camera at different viewpoints. This is a core competency of many vision-based localisation tasks including visual odometry and visual place recognition.

Wide-angle cameras have a large field of view that can exceed a full hemisphere, and the images they produce contain severe radial distortion. When compared to traditional narrow field of view perspective cameras, more accurate estimates of camera egomotion can be found using the images obtained with wide-angle cameras. The ability to accurately estimate camera egomotion is a fundamental primitive of visual odometry, and this is one of the reasons for the increased popularity in the use of wide-angle cameras for this task. Their large field of view also enables them to capture images of the same regions in a scene taken at very different viewpoints, and this makes them suited for visual place recognition. However, the ability to estimate the camera egomotion and recognise the same scene in two different images is dependent on the ability to reliably detect and describe the same scene points, or ‘keypoints’, in the images. Most algorithms used for this purpose are designed almost exclusively for perspective images.

Applying algorithms designed for perspective images directly to wide-angle images is problematic as no account is made for the image distortion. The primary contribution of this thesis is the development of two novel keypoint detectors, and a method of keypoint description, designed for wide-angle images. Both reformulate the Scale-Invariant Feature Transform (SIFT) as an image processing operation on the sphere. As the image captured by any central projection wide-angle camera can be mapped to the sphere, applying these variants to an image on the sphere enables keypoints to be detected in a manner that is invariant to image distortion. Each of the variants is required to find the scale-space representation of an image on the sphere, and they differ in the approaches they used to do this. Extensive experiments using real and synthetically generated wide-angle images are used to validate the two new keypoint detectors

and the method of keypoint description. The best of these two new keypoint detectors is applied to vision based localisation tasks including visual odometry and visual place recognition using outdoor wide-angle image sequences. As part of this work, the effect of keypoint coordinate selection on the accuracy of egomotion estimates using the Direct Linear Transform (DLT) is investigated, and a simple weighting scheme is proposed which attempts to account for the uncertainty of keypoint positions during detection. A word reliability metric is also developed for use within a visual ‘bag of words’ approach to place recognition.

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Fundamentals and Challenges of Vision-Based Localisation . . . . .	3
1.1.1	Visual Odometry . . . . .	3
1.1.2	Visual Place Recognition . . . . .	18
1.2	Wide-Angle Image Processing . . . . .	20
1.3	Research Questions and Methodology . . . . .	20
1.4	Contributions . . . . .	23
1.5	Structure of Thesis . . . . .	24
<b>2</b>	<b>Wide-Angle Image Formation and Calibration</b>	<b>27</b>
2.1	Introduction . . . . .	28
2.2	Image Formation . . . . .	29
2.2.1	Perspective Cameras . . . . .	29
2.2.2	Wide-Angle Cameras . . . . .	31
2.2.2.1	Central vs Non-central Cameras . . . . .	32
2.2.2.2	Catadioptric . . . . .	34
2.2.2.3	Wide-Angle Dioptric (fisheye) . . . . .	44
2.3	Review of Camera Calibration . . . . .	54
2.3.1	Full-range . . . . .	55
2.3.2	Auto-calibration . . . . .	56
2.3.3	Plumb-line . . . . .	58

2.3.4	Discussion . . . . .	60
2.4	Camera Calibration Algorithm and Results . . . . .	62
2.4.1	Preliminaries . . . . .	62
2.4.2	Objective Function . . . . .	65
2.4.3	Parametrisation and initialisation of camera extrinsic rotation $R_f$ . . . . .	69
2.4.4	Implementation . . . . .	70
2.4.5	Grid Point Detection . . . . .	71
2.4.6	Calibration Example . . . . .	75
2.4.7	Experiments and Results . . . . .	81
2.4.7.1	Initial Model Estimates . . . . .	82
2.4.7.2	Results and Discussion . . . . .	83
2.5	Conclusions . . . . .	86
<b>3</b>	<b>Keypoint Detection, Description and Matching with Applications to Wide-Angle Images</b>	<b>89</b>
3.1	Introduction . . . . .	90
3.2	Classical (small-baseline) Techniques . . . . .	92
3.2.1	Sparse Optical Flow . . . . .	92
3.2.1.1	Keypoint Detection . . . . .	92
3.2.1.2	Finding Keypoint Correspondences: Registration and Matching . . . . .	99
3.2.2	Dense Optical Flow . . . . .	101
3.2.3	Limitations for Wide-baseline Motion . . . . .	103
3.3	Wide-Baseline Techniques . . . . .	105
3.3.1	Scale-Space Approaches . . . . .	107
3.3.1.1	Scale-Space Primal Sketch and Automatic Scale Selection . . . . .	111
3.3.1.2	Multi-scale Harris, Harris-Laplace, Hessian-Laplace, Harris-Affine and Hessian-Affine . . . . .	115

---

3.3.1.3	The Scale-Invariant Feature Transform (SIFT) . . . .	117
3.3.1.4	Speeded up Robust Features (SURF) - ‘Fast Hessian’	122
3.3.2	Alternate Approaches . . . . .	123
3.3.2.1	Scale-Saliency . . . . .	123
3.3.2.2	Tuytelaars and Van Gool . . . . .	125
3.3.2.3	Maximally Stable Extremal Regions (MSER’s) . . . .	127
3.3.3	Keypoint Descriptors . . . . .	127
3.3.3.1	SIFT descriptor . . . . .	129
3.3.3.2	Gradient Location-Orientation Histogram (GLOH) .	131
3.3.3.3	SURF descriptor . . . . .	132
3.3.4	Keypoint Matching . . . . .	134
3.3.5	Suitability for vision-based localisation . . . . .	136
3.4	Wide-baseline Keypoint Matching with Wide-angle Images . . . . .	142
3.4.1	Existing Approaches . . . . .	143
3.4.2	Methods Designed for Wide-Angle Images . . . . .	145
3.4.3	Proposed Approach to Wide-Baseline Matching with Wide- Angle Images . . . . .	148
3.5	Conclusions . . . . .	150
<b>4</b>	<b>Wide-Baseline Keypoint Detection, Description and Matching with Wide- Angle Images</b>	<b>153</b>
4.1	Introduction . . . . .	153
4.2	Scale-Space for Wide-Angle Images . . . . .	155
4.2.1	Spherical Harmonics . . . . .	156
4.2.2	Spherical Gaussian Function . . . . .	158
4.2.3	Spherical Diffusion by Convolution . . . . .	160
4.2.3.1	Spatial Domain . . . . .	161
4.2.3.2	Spherical Fourier Domain . . . . .	164

4.3	Scale-Invariant Keypoint Detection: spherical SIFT (sSIFT) . . . . .	165
4.3.1	Spherical Fourier Transform (spectrum) of a Wide-Angle Image	166
4.3.1.1	Sample Scheme . . . . .	166
4.3.1.2	Spherical Bandwidth of a Wide-Angle Image . . . . .	167
4.3.1.3	Anti-aliasing . . . . .	171
4.3.2	Scale Selection . . . . .	174
4.3.2.1	Input and Initial Scale . . . . .	176
4.3.2.2	Scales per octave and number of octaves . . . . .	178
4.3.3	Obtaining the Scale-Space Images . . . . .	179
4.3.4	Keypoint Detection . . . . .	180
4.3.5	Keypoint Support Region . . . . .	183
4.3.6	Experiments . . . . .	185
4.3.6.1	Input Data . . . . .	187
4.3.6.2	Keypoints . . . . .	189
4.3.6.3	Performance Metrics . . . . .	192
4.3.6.4	Results . . . . .	193
4.3.6.5	Discussion . . . . .	196
4.3.7	Conclusions . . . . .	204
4.4	Scale-Invariant Keypoint Detection: parabolic SIFT (pSIFT) . . . . .	206
4.4.1	Conversion to a Stereographic Image . . . . .	206
4.4.2	Approximate Spherical Diffusion using Stereographic Projection	208
4.4.2.1	Scale correction . . . . .	210
4.4.2.2	Approximation Error . . . . .	212
4.4.3	Scale-selection . . . . .	213
4.4.4	Efficient Computation of Scale-Space Images . . . . .	213
4.4.4.1	Separable Convolution . . . . .	215
4.4.4.2	Cascade Filtering . . . . .	216

---

4.4.4.3	Octave-based Approach . . . . .	217
4.4.5	Keypoint Detection and Support Region . . . . .	218
4.4.6	Implementation . . . . .	219
4.4.7	Experiments: percentage correlation and number of correspondences . . . . .	221
4.4.7.1	Results . . . . .	221
4.4.7.2	Discussion . . . . .	226
4.4.8	Experiments: performance versus image position . . . . .	227
4.4.8.1	Results . . . . .	228
4.4.8.2	Discussion . . . . .	228
4.4.9	Conclusions . . . . .	231
4.5	sSIFT and pSIFT Keypoint Descriptors . . . . .	231
4.5.1	Experiments . . . . .	234
4.5.1.1	Data . . . . .	234
4.5.1.2	Performance Metric . . . . .	235
4.5.1.3	Results . . . . .	235
4.5.1.4	Discussion and Conclusions . . . . .	236
4.6	Experiments: Keypoint Detection, Description and Matching . . . . .	236
4.6.1	Keypoint Types . . . . .	236
4.6.2	Producing Perspective Images . . . . .	238
4.6.3	Wide-Angle Image Sequences . . . . .	239
4.6.3.1	Fisheye: Fisheye Camera . . . . .	239
4.6.3.2	Hyperion: Equiangular Catadioptric Camera . . . . .	240
4.6.3.3	Tractor: Equiangular Catadioptric Camera . . . . .	244
4.6.4	Performance metrics . . . . .	247
4.6.5	Similarity Metrics . . . . .	248
4.6.6	Selecting Correct Correspondences . . . . .	248

4.6.7	Results . . . . .	249
4.6.8	Discussion and Conclusions . . . . .	249
4.7	Conclusions . . . . .	262
<b>5</b>	<b>Applications to Vision Based Localisation</b>	<b>265</b>
5.1	Introduction . . . . .	265
5.2	Visual Odometry . . . . .	267
5.2.1	Keypoint Detection and Matching . . . . .	270
5.2.2	Frame-Rate Selection for Egomotion Estimation . . . . .	270
5.2.3	Visual Odometry Trials . . . . .	272
5.2.4	Measuring Visual Odometry Accuracy . . . . .	273
5.2.5	Ground Plane Visual Odometry with Coplanar World Points . . . . .	274
5.2.6	Ground Plane Visual Odometry: Euclidean . . . . .	278
5.2.6.1	Weighted DLT . . . . .	285
5.2.6.2	Iterative Weighted DLT . . . . .	287
5.2.6.3	Experiments: Linear Estimate . . . . .	288
5.2.6.4	Iterative Refinement . . . . .	293
5.2.6.5	Experiments: Iterative Refinement . . . . .	294
5.2.7	Ground Plane Visual Odometry: Triggs' Method . . . . .	298
5.2.7.1	Experiments: Linear Estimate . . . . .	300
5.2.7.2	Iterative Refinement . . . . .	303
5.2.7.3	Experiments: Iterative Refinement . . . . .	305
5.2.8	Discussion and Conclusions: Ground Plane Visual Odometry . . . . .	308
5.2.8.1	Direct Linear Transforms (DLT's) . . . . .	308
5.2.8.2	Euclidean versus Triggs . . . . .	309
5.2.8.3	Fixed versus Variable Frame-rates . . . . .	311
5.2.9	Generalised (Unconstrained) Visual Odometry . . . . .	313



---

5.2.9.1	Epipolar Geometry and the Essential Matrix . . . . .	314
5.2.9.2	Linear estimates: fixed versus variable frame-rate . . . . .	316
5.2.9.3	Visual Odometry Algorithm . . . . .	317
5.2.9.4	Experiments: Hyperion Sequence . . . . .	322
5.2.9.5	Experiments: Fisheye Sequence . . . . .	324
5.2.10	Conclusions . . . . .	324
5.3	Visual place recognition . . . . .	326
5.3.1	Establishing the visual-words . . . . .	327
5.3.2	Word reliability . . . . .	327
5.3.3	Visual Word Vector . . . . .	328
5.3.4	Image Similarity . . . . .	329
5.3.5	Results and Discussion . . . . .	330
5.4	Conclusions . . . . .	333
<b>6</b>	<b>Conclusions</b>	<b>335</b>
6.1	Answers to questions posed . . . . .	337
6.2	Contributions of the Thesis . . . . .	339
6.3	Further directions . . . . .	341
<b>A</b>	<b>Calibration Results for Fisheye Images</b>	<b>343</b>
<b>B</b>	<b>Spherical Harmonic Expansion of the Spherical Dirac Function</b>	<b>355</b>
<b>C</b>	<b>Computation of Discrete First Order Derivatives and Hessian Matrix for Keypoint Interpolation</b>	<b>357</b>



# List of Figures

1.1	Image of the same scene obtained with a narrow field of view perspective camera and a wide-angle (fisheye) camera. . . . .	3
1.2	Visual Odometry from an image sequence. . . . .	6
1.3	Spherical flow fields for camera rotation and translation. . . . .	8
1.4	Optical flow fields for rotation and translation. . . . .	9
1.5	Appearance of the theoretical spherical flow fields for pure rotation and translation with a perspective and wide-angle camera. . . . .	16
1.6	Loop closure for a mobile robot in an outdoor environment . . . . .	18
1.7	The large field of view of wide-angle cameras makes them suited to visual place recognition tasks. . . . .	19
2.1	Image formation with a perspective camera. . . . .	30
2.2	Central versus non-central catadioptric projection. . . . .	34
2.3	Wide-angle cameras, the images they produce, and their conversion to perspective images. . . . .	35
2.4	Spherical coordinates . . . . .	36
2.5	Inability to reconstruct geometrically correct images from non-central cameras without known caustic of viewpoint and depth of points. . . . .	36
2.6	The three practical central catadioptric cameras using elliptical, hyperbolic and parabolic reflective surfaces. . . . .	39
2.7	Geometric equivalence of image formation using a parabolic catadioptric camera and the unified image model. . . . .	40

2.8	Equivalent image formation using a parabolic catadioptric camera and the unified image model. . . . .	41
2.9	Nomenclature for the unified image model. . . . .	41
2.10	Points of projection for central catadioptric cameras under the unified image model. . . . .	42
2.11	Equiangular catadioptric camera model. . . . .	43
2.12	Pinhole-based and ray-based fisheye camera models. . . . .	45
2.13	Example image of the planar checkerboard calibration target obtained with the fisheye camera used for calibration. . . . .	63
2.14	Grid point nomenclature. . . . .	64
2.15	Camera calibration nomenclature for grid points on the sphere. . . . .	65
2.16	General procedure of the semi-supervised grid point detection operating in parallel with camera calibration. . . . .	72
2.17	Example grid point detection for each of the four outermost grid points given the local patch surrounding each. . . . .	74
2.18	The 10 input fisheye images used for calibration. Each image is of size $1024 \times 768$ pixels. . . . .	76
2.19	Calibration results for image 6. . . . .	77
2.20	Unified image model function obtained from calibration. . . . .	78
2.21	Plot of the $x, y$ reprojection errors in each original fisheye image after calibration with the unified image model. . . . .	80
2.22	Distribution of reprojection errors versus radius $r$ from the camera's principal point, and the probability distribution of the reprojection errors. . . . .	81
2.23	Boxplot comparison of the reprojection errors in the fisheye image plane for the unified and unified (affine) camera models. . . . .	81
2.24	Initial estimates of the camera model functions obtained prior to calibration. . . . .	84
2.25	Box-plot of the calibration results for each ray-based fisheye camera model considered. . . . .	85
3.1	Example keypoint detection using the Harris/Plessey detector. . . . .	93

---

3.2	Cross-correlation for a corner, edge and homogeneous point. . . . .	94
3.3	Keypoint detection and matching with Harris corners (ZNCC) and SIFT. 104	
3.4	Set of derived scale-space images $L(u, v; t)$ obtained from a primal image $I(u, v)$ . . . . .	107
3.5	Detection of blob like structures via binary thresholding. . . . .	112
3.6	Automatic scale-selection using a scale normalised differential metric. . . . .	114
3.7	Local extrema for the Harris or Hessian measure (space), and the scale-normalised Laplacian (scale). . . . .	116
3.8	SIFT selects candidate keypoints as local extrema compared to the 26 nearest pixels in the current and adjacent DoG images. . . . .	118
3.9	Octave approach used by SIFT for efficient image processing. . . . .	120
3.10	Box filter approximates of second order derivative of Gaussian used by SURF. . . . .	122
3.11	Automatic scale selection for the scale-saliency keypoint detector. . . . .	124
3.12	Geometry based method. A parallelogram is used to enclose an affine invariant region based on the greyscale intensity function along line segments originating from corner points. . . . .	125
3.13	Intensity-based algorithm used for scale and affine invariant keypoint detection. . . . .	126
3.14	MSER keypoint detection. . . . .	127
3.15	Mapping of local image content within a keypoint's support regions to fixed sized patch. . . . .	130
3.16	General procedure used to evaluate a SIFT descriptor. . . . .	131
3.17	SIFT and GLOH index cell arrays. . . . .	132
3.18	First order Haar wavelets used to compute a response $d_u$ and $d_v$ in orthogonal directions. . . . .	133
3.19	Epipolar geometry used for guided matching. . . . .	136
3.20	A wide-angle catadioptric image converted to a log-polar panoramic image. . . . .	141

3.21	'Rectified' log-polar and cylindrical panoramic images produced from a calibrated parabolic catadioptric camera. . . . .	143
3.22	A fisheye image converted to a perspective image. . . . .	144
3.23	Rotational shift invariance for different image representations. . . . .	147
3.24	Shift-invariant convolutions on the image and sphere. . . . .	149
4.1	Magnitude of the real components of the spherical harmonic functions $Y_l^m$ . . . . .	158
4.2	The spherical Gaussian function versus angle of colatitude $\theta$ . . . . .	160
4.3	Notations for the spherical Gaussian function and the associated kernels on the image plane. . . . .	163
4.4	The sampling scheme used by s2kit ( $b=8$ ) . . . . .	167
4.5	Coordinate system of image plane. The vector $dP$ represents a small shift at angle $\alpha$ from a point on the image at radius $r$ from the image centre. . . . .	168
4.6	The estimated image bandwidths $b_I(r, \alpha)$ of the fisheye and parabolic catadioptric cameras. . . . .	172
4.7	Camera model for the fisheye and parabolic catadioptric cameras . . .	173
4.8	Values of the anti-aliasing filter versus angle of colatitude for bandwidths $b = 128, 256, 512$ . . . . .	173
4.9	The (a) window function and (b) the ideal filter before and after the application of the window function. . . . .	175
4.10	Theoretical zonal harmonic coefficients of the windowed anti-aliasing filter. . . . .	175
4.11	Effect of the anti-aliasing interpolation filter on the magnitude of the image spectrum. . . . .	176
4.12	Fitted line obtained via least squares minimisation . . . . .	178
4.13	Comparison of $G(\cdot; \sigma)$ and $G_{\mathbb{S}^2}(\cdot; kt = (\sigma \theta_s)^2/2)$ for the parabolic catadioptric and fisheye cameras. . . . .	179
4.14	General procedure used by sSIFT to find the scale-space images $\mathcal{L}_{\mathbb{S}^2}(\cdot; kt)$ . 181	
4.15	Example keypoint detection using sSIFT. . . . .	184

4.16	The boundary of the support region for an sSIFT keypoint is defined as a circle on the sphere. . . . .	185
4.17	The support regions for a set of sSIFT keypoints as they appear on a fisheye image and on the unit sphere. . . . .	186
4.18	The data set consisting of 40 input images of size $2272 \times 1704$ pixels. . . . .	188
4.19	The set of synthetically generated parabolic catadioptric wide-angle images for one of the reference images. . . . .	190
4.20	The set of synthetically generated wide-angle fisheye images for one of the reference images. . . . .	191
4.21	Keypoint correspondences found between two synthetic wide-angle fisheye images. . . . .	196
4.22	Median percentage correlation and number of correspondences for images subject to change in rotation for SIFT and sSIFT. . . . .	197
4.23	Median percentage correlation and number of correspondences for images subject to change in both rotation and scale for SIFT and sSIFT. . . . .	198
4.24	Percentage increase in percentage correlation and number of correspondences using the anti-aliasing filter. . . . .	199
4.25	Percentage increase in the percentage correlation of keypoints and the overall number of keypoint correspondences of sSIFT( $512^*$ ) over sSIFT( $b = 256^*$ ). . . . .	201
4.26	Variation in global scale overlap for different sample rates and image transformations. . . . .	202
4.27	Percentage increase in the percentage correlation of keypoints and the overall number of keypoint correspondences for spherical SIFT with bandwidth $b = 512^*$ relative to SIFT. . . . .	203
4.28	Conversion of a fisheye image to a stereographic image. . . . .	207
4.29	Appearance of the spherical Gaussian kernels $G_{S^2}(\cdot; kt)$ and $G_{S^2(\eta')}(\cdot; kt)$ as they would appear on a stereographic image. . . . .	209
4.30	Corrected scale at a given radius on the image plane. The value $r_{\theta=\pi/2}$ is the radius on the image plane which projects via inverse stereographic projection to a point $\eta$ on the equator of the sphere. . . . .	212

4.31	pSIFT approximation error as a function of the radius $r$ from the principal point. . . . .	213
4.32	Difference between $\tilde{\mathcal{G}}_{\mathbb{S}^2}(\cdot; kt)$ and $\mathcal{G}_{\mathbb{S}^2(\eta')}(\cdot; kt(r))$ with and without scale correction. . . . .	214
4.33	Estimates of the separable kernels which approximate $\tilde{\mathcal{G}}_{\mathbb{S}^2}(\cdot; kt)$ . . . . .	217
4.34	Change in appearance of a local region for a change in camera rotation. . . . .	220
4.35	Median percentage correlation and number of correspondences for images subject to change in rotation for SIFT, sSIFT and pSIFT. . . . .	224
4.36	Median percentage correlation and number of correspondences for images subject to change in both rotation and scale for SIFT, sSIFT and pSIFT. . . . .	225
4.37	Percentage increase in the percentage correlation of keypoints and the overall number of correspondences for sSIFT(512*) and pSIFT relative to SIFT. . . . .	226
4.38	PDF and reliability of SIFT, sSIFT and pSIFT versus angle of colatitude for the parabolic catadioptric camera. . . . .	229
4.39	PDF and reliability of SIFT, sSIFT and pSIFT versus angle of colatitude for the parabolic catadioptric camera. . . . .	230
4.40	Angles used to define the projection of the image content within a keypoint's support region to a fixed sized patch. . . . .	232
4.41	Recall versus 1-precision results for each camera using a perspective, stereographic and equiangular projection. . . . .	235
4.42	Appearance of greyscale intensity values within a keypoint's support region mapped to a fixed sized patch using a perspective, stereographic and equiangular projection. . . . .	237
4.43	Overview of the fisheye image sequence used in recall versus 1-precision experiments. . . . .	241
4.44	Overview of the Hyperion image sequence used in recall versus 1-precision experiments. . . . .	243
4.45	Estimated bandwidth of the equiangular catadioptric camera (Hyperion sequence). . . . .	244



---

4.46	Overview of the Tractor image sequence used in recall versus 1-precision experiments. . . . .	246
4.47	Estimated bandwidth of the equiangular catadioptric camera on-board the mobile robot tractor. . . . .	247
4.48	Recall vs. 1-precision results for the fisheye sequence (frame-rate 1). .	250
4.49	Recall vs. 1-precision results for the fisheye sequence (frame-rate 2). .	251
4.50	Recall vs. 1-precision results for the Hyperion sequence (frame-rate 1). .	252
4.51	Recall vs. 1-precision results for the Hyperion sequence (frame-rate 2). .	253
4.52	Recall vs. 1-precision results for the Hyperion sequence (frame-rate 4). .	254
4.53	Recall vs. 1-precision results for the tractor sequence (frame-rate 1). .	255
4.54	Recall vs. 1-precision results for the tractor sequence (frame-rate 2). .	256
4.55	Example set of correspondences found using SIFT and sSIFT in two images separated by a large change in camera pose. . . . .	259
5.1	Fisheye image sequence used for visual odometry estimates and place recognition. . . . .	266
5.2	Number of frames between successive estimates of camera egomotion. . . . .	272
5.3	Alignment of visual odometry and ground truth segments used to find the error in the visual odometry estimates. . . . .	275
5.4	Position of keypoints on the stereographic image and the corresponding error ellipses on the ground plane. . . . .	282
5.5	The angular thresholds used to estimate the visual odometry for the Hyperion sequence with the standard DLT. . . . .	283
5.6	Visual odometry errors (Hyperion) using linear estimate and Euclidean ground plane constraint — standard DLT. . . . .	284
5.7	Visual odometry errors (Hyperion) using linear estimate and Euclidean ground plane constraint for the fixed frame-rate — spherical, weighted and iterative weighted DLT's. . . . .	289
5.8	Visual odometry errors (Hyperion) using linear estimate and Euclidean ground plane constraint for the variable frame-rate — spherical, weighted and iterative weighted DLT's. . . . .	290

5.9	Box plot of the visual odometry errors (Hyperion) using non-linear estimate and Euclidean ground plane constraint — transfer and geometric errors. . . . .	296
5.10	Visual odometry estimates versus GPS ground truth for the Euclidean ground plane constraint. . . . .	297
5.11	Box plots of the visual odometry errors using fixed frame-rate (Hyperion, Triggs ground plane constraint) — standard, spherical, weighted, and iterative weighted DLT's. . . . .	301
5.12	Box plots of the visual odometry errors using variable frame-rate (Hyperion, Triggs ground plane constraint) — standard, spherical, weighted, and iterative weighted DLT's. . . . .	302
5.13	Box plots of the visual odometry errors (Hyperion, Triggs ground plane constraint) — linear and non-linear estimates. . . . .	306
5.14	Visual odometry estimates versus GPS ground truth for the Trigg's ground plane constraint. . . . .	307
5.15	Box plots of the visual odometry errors (Hyperion, Euclidean ground plane constraint) — fixed and variable frame-rates. . . . .	312
5.16	Box plots of the visual odometry errors (Hyperion, Triggs ground plane constraint) — fixed and variable frame-rates. . . . .	313
5.17	The epipolar geometry between two cameras centred at points $\mathbf{C}$ and $\mathbf{C}'$ . . . . .	314
5.18	Visual odometry estimates versus GPS ground truth for fixed frame-rate — generalised visual odometry using linear estimate of the essential matrix. . . . .	318
5.19	Visual odometry estimates versus GPS ground truth for variable frame-rate — generalised visual odometry using linear estimate of the essential matrix. . . . .	319
5.20	Visual odometry results for Hyperion sequence without constraints on camera motion or scene points. . . . .	323
5.21	Visual odometry results for Fisheye sequence without constraints on camera motion or scene points. . . . .	325
5.22	Example visual words in the vocabulary. . . . .	330

---

5.23	Similarity matrices for the fisheye sequence using visual word vectors with and without the reliability weighting. . . . .	331
5.24	Thresholding of the cosine similarity score using the visual word vectors $V'$ for a value of 0.6. . . . .	331
5.25	The location of the potential visual loop closure events. . . . .	332
A.1	Calibration results for image 1. . . . .	344
A.2	Calibration results for image 2. . . . .	345
A.3	Calibration results for image 3. . . . .	346
A.4	Calibration results for image 4. . . . .	347
A.5	Calibration results for image 5. . . . .	348
A.6	Calibration results for image 6. . . . .	349
A.7	Calibration results for image 7. . . . .	350
A.8	Calibration results for image 8. . . . .	351
A.9	Calibration results for image 9. . . . .	352
A.10	Calibration results for image 10. . . . .	353



# List of Tables

2.1	Solutions for the entire class of central projection catadioptric cameras.	37
2.2	Relationship between the unified image model parameters $m$ and $l$ and the eccentricity $\epsilon$ of the reflective surface (swept conic section).	42
2.3	Summary of pinhole camera models.	46
2.4	Ray-based camera models.	50
2.5	Summary of the calibration results for the unified and unified (affine) camera models.	79
2.6	Summary of the calibration results for each ray-based fisheye camera model considered.	86
4.1	Distances and rotation angles $R = R_y(\beta)R_x(\alpha)$ used to generate the wide-angle images.	189
4.2	Median percentage correlation and number of correspondences for the parabolic catadioptric camera using SIFT and sSIFT (mean values shown in brackets).	194
4.3	Median percentage correlation and number of correspondences for the fisheye camera using SIFT and sSIFT (mean values shown in brackets).	195
4.4	Median percentage correlation and number of correspondences for the parabolic catadioptric camera using SIFT, sSIFT and pSIFT (mean values shown in brackets).	222
4.5	Median percentage correlation and number of correspondences for the fisheye camera using SIFT, sSIFT and pSIFT (mean values shown in brackets).	223
4.6	Mean number of correct keypoint correspondences between image pairs.	257

5.1	Visual odometry errors (Hyperion, Euclidean ground plane constraint) — standard DLT. . . . .	281
5.2	Visual odometry errors (Hyperion, Euclidean ground plane constraint) — standard, spherical, weighted and iterative weighted DLT's. . . . .	291
5.3	Visual odometry errors (Hyperion, Euclidean ground plane constraint) — standard, spherical, weighted and iterative weighted DLT's. . . . .	292
5.4	Visual odometry errors (Hyperion, Euclidean ground plane constraint) using non-linear estimate and fixed frame-rate — transfer and geometric error cost functions. . . . .	298
5.5	Visual odometry errors using fixed frame-rate (Hyperion, Triggs ground plane constraint) — standard, spherical, weighted, and iterative weighted DLT's. . . . .	303
5.6	Visual odometry errors using variable frame-rate (Hyperion, Triggs ground plane constraint) — standard, spherical, weighted, and iterative weighted DLT's. . . . .	304
5.7	Visual odometry errors (Hyperion, Triggs ground plane constraint) — linear and non-linear estimates. . . . .	308
5.8	Visual odometry errors (Hyperion, Euclidean and Triggs ground plane constraint) — fixed and variable frame-rates. . . . .	311

# Abbreviations

SIFT	Scale-Invariant Feature Transform
sSIFT	Spherical Scale-Invariant Feature Transform
pSIFT	Parabolic Scale-Invariant Feature Transform
MSER	Maximally Stable Extremal Region
SVD	Singular Value Decomposition
PCA	Principal Component Analysis
SFT	Spherical Fourier Transform
ISFT	Inverse Spherical Fourier Transform
DLT	Direct Linear Transform





# Publications


1. Peter Hansen, Peter Corke, Wageeh Boles and Kostas Daniilidis. Scale Invariant Feature Matching with Wide Angle Images. Proceedings *International Conference on Intelligent Robots and Systems (IROS)*, pages 1689-1694, San Diego, USA, 2007.
2. Peter Hansen, Peter Corke, Wageeh Boles and Kostas Daniilidis. Scale-Invariant Features on the Sphere. Proceedings *International Conference on Computer Visison (ICCV)*, pages 1-8, Rio de Janeiro, Brazil, 2007.
3. Peter Hansen, Peter Corke and Wageeh Boles. Outdoor Localization Using Wide-Angle Visual Feature Matching and Image Retrieval. In *13th International Symposium of Robotics Research (ISRR)*, pages 00-00, Hiroshima, Japan, 2007.
4. Peter Hansen, Wageeh Boles and Peter Corke. Spherical Diffusion for Scale-Invariant Keypoint Detection in Wide-Angle Images. *Digital Image Computing: Techniques and Applications (DICTA)*, Canberra, Australia, 2008.
5. Peter Hansen, Peter Corke and Wageeh Boles. Wide-Angle Visual Feature Matching for Outdoor Localization. *The International Journal of Robotics Reseach*, Vol. 29, No. 2-3, pages 267-297, 2010.



# Authorship

“The work contained in this thesis has not been submitted to meet requirements for an award at this or any other higher education institution. To the best of my knowledge and belief, the thesis contains no material previously published or written by another person except where due reference is made.”

*Signature:*

A handwritten signature in black ink, consisting of a series of loops and a long horizontal stroke ending in a small dot.

*Date:*

*31 May 2010*



# Acknowledgments

I would like to thank my supervisors, Professor Wageeh Boles and Dr Peter Corke, for their guidance and support during my candidature, and for the advice and suggestions they gave to me during research discussions. In particular I would like to thank Peter Corke for all the efforts he made reviewing extensive draft material, especially thesis chapters, and for the thorough and helpful feedback he provided. Peter Corke also deserves a special mention for the contributions he made in the writing of papers for publication.

I am grateful to QUT and CSIRO for providing me with research funding during my candidature, and to CSIRO for access to their laboratories and equipment. During my candidature I was located at the CSIRO ICT Centre Centre Autonomous System Laboratory at QCAT, and I would like to thank all the staff and students in the lab. I spent time with many of them discussing my research, and I am very appreciative for all the comments, suggestions and help they provided me. In particular I would like to thank Stephen Nuske for being my unofficial computer support ‘guru’ during my early days at the lab. I want to also thank Kane Usher, Jonathan Roberts, Peter Corke and Carnegie Mellon University for providing me with image data sets.

I was fortunate enough to spend four months of my candidature at the GRASP Laboratory at the University of Pennsylvania as a visiting research student. I am grateful to the GRASP Laboratory for giving me this opportunity, and to Professor Kostas Daniilidis for supervising me during my stay. Kostas Daniilidis spent a considerable amount of time with me discussing wide-angle image processing, in particular the concept and practical implementation of image processing on the sphere. The advice and suggestions he gave me were invaluable. I want to also thank the other staff and students who were at the GRASP Laboratory during my stay for their help and hospitality.

In particular I would like to thank Ameesh Makadia for sharing with me his expertise regarding spherical FFT's and a collection of his Matlab/mex files.

My friends and family have been extremely supportive throughout the years, and for this I thank them. Finally, I am grateful to my parents for their continuous patience and support. I dedicate my thesis to them.

To my Parents.

# Chapter 1

## Introduction

*Most mobile robots are required to be able to localise themselves with respect to their surrounding environment. Vision sensors can be used for this purpose, and two core components of many vision-based localisation systems are visual odometry and visual place recognition. Wide-angle cameras are frequently used since they have considerable advantages compared to narrow field of view planar perspective cameras for incremental motion estimation due to their ability to more accurately disambiguate rotational and translational motion. The large field of view of wide-angle cameras also has potential advantages for visual place recognition. A critical question which has received little attention is how image processing algorithms should be applied to the images acquired with wide-angle cameras which inherently have extreme radial distortion. A major contribution of this work is a suitable method of keypoint detection and description in wide-angle images and a robust wide-baseline keypoint matching algorithm suitable for use with wide-angle images. The algorithm is suited to, and evaluated for, visual odometry and visual place recognition.*

Mobile robots have numerous and growing applications in real world scenarios. This work is motivated by applications to environmental monitoring and exploration. Examples include robots for ecological monitoring of the Great Barrier Reef [65, 239], and the Mars exploration rovers [145], both of which operate in unstructured outdoor environments without navigational infrastructure such as GPS. Critical to the success of these, and many other outdoor mobile robots, is the ability to localise either with respect to a global coordinate frame, or in a local coordinate frame with respect to objects near the robot.

When considering outdoor above ground vehicle localisation GPS appears to be an attractive solution requiring only a GPS receiver to obtain a global coordinate for the vehicle. However, numerous factors can limit its use which may be considered as the failure modes for GPS, and include multi-path propagation in natural or urban canyons, degenerate satellite configurations, and occlusions due to buildings for example. GPS can at best only provide a global coordinate, but without a known map of the operating environment there is no way for the vehicle to estimate its surroundings without additional sensors. Using GPS alone is therefore not suited for dynamic or unknown environments. It is interesting to note that for the case where a map of the environment is known, other sensors such as scanning laser rangefinders [219] and computer vision [186] have been shown through extensive experiments to provide effective alternatives to GPS.

Both laser and vision sensors can also be used for localisation in unstructured environments. As they are able to perceive the surrounding environment, they have the additional capability of being able to incrementally build a map of the operating environment as they move, a process referred to as simultaneous localisation and mapping (SLAM). This map can then be used to aid in localisation and identify when the vehicle returns to a previously visited location. Laser scanners have been demonstrated as a suitable sensor for this purpose in large scale outdoors environments, in particular using SLAM by Bosse et al [23, 24]. Although visual SLAM is presently suited predominantly to small structured environments, accurate localisation using only vision has also been demonstrated in unstructured outdoors environments with notable works including Nistér et al [181, 182], Tardif et al [218], Scaramuzza [198] and Maimone et al [145]. However, there are a number of advantages in using vision over laser scanners. Vision sensors are relatively low cost, small and lightweight making them easily retrofitted to most robots. Vision is also more information rich, capable of obtaining information over an area of the environment (compared to a single scanning point) with additional cues such as colour and texture. The use of vision for localisation is considered in this thesis for these reasons. More specifically, *wide-angle* cameras are considered as they have potential advantages when compared to narrow field of view cameras, as will be discussed.

Image formation with narrow field of view cameras is typically a perspective projection of scene points to the image plane where straight lines in the scene appear straight in the image, as illustrated in figure 1.7a. The term *perspective camera* will be used to refer exclusively to narrow field of view perspective camera for the remainder of the thesis. Although image formation with some wide-angle cameras is described by perspective projection, the term *wide-angle camera* will be used to refer to those





(a) Narrow field of view perspective camera.



(b) Wide-angle camera (fisheye lens) with a field of view in excess of a full hemisphere.

Figure 1.1: Images of the same scene obtained with a narrow field of view perspective camera and a wide-angle (fisheye) camera. The dashed red lines on the wide-angle image show the superimposed field of view of the perspective camera in (a). The radial distortion in the wide-angle camera is most evident towards the lower half of the image where the straight lines of the tiles are curved in the image.

with a near hemispherical field of view whose output images are characterised as having extreme radial distortion where straight lines in space appear curved in the image, as shown in figure 1.1b.

## 1.1 Fundamentals and Challenges of Vision-Based Localisation

### 1.1.1 Visual Odometry

Camera egomotion is the change in rotation and translation between camera viewpoints. By integrating incremental estimates of camera egomotion, one can find the location of the camera with respect to some reference starting location. This process is typically referred to in the robotics literature as visual odometry [181, 182] and is the foundation for many vision-based localisation frameworks. Camera egomotion can be estimated by observing how the appearance of the surrounding environment changes from one image to the next. For example, figure 1.2a shows a sequence of images obtained from a camera moving through an indoor environment. From human perception it is clear that the camera is moving forwards through the scene, and this is based on observing how objects in the environment change position through the image sequence.

The change in appearance of the environment between two images needs to be measured quantitatively to estimate camera egomotion. Assume that each pixel in the first image is associated with a unique scene point. For each pixel in the first image, the position of the corresponding scene points in the second image could be found where the change in position gives the so called *dense* optical flow. The *sparse* optical flow is more typically used in many visual odometry applications which uses only a select number of salient scene points, although some methods have been proposed which attempt to obtain an egomotion estimate without the need to find any corresponding points [146]. Referring to figure 1.2, given a sequence of images (1.2a) the fundamental steps of visual odometry using sparse optical flow may be summarised as follows:

1. Locate and describe distinctive world points in the images (figure 1.2b): this is the method by which image processing algorithms identify visually salient (distinctive) scene points in the images. Many terminologies have been used to describe these sets, including but not limited to: corners, features, interest points and keypoints. The terminologies used are typically specific to a particular algorithm, however, to avoid confusion the term *keypoint* will be used in this chapter. Each keypoint is then assigned a unique *descriptor* which encodes the image information within a local region surrounding the keypoint. There are a large number of methods used to detect and describe keypoints, and a detailed taxonomy is presented in chapter 3.
2. Find corresponding keypoints between successive images (figure 1.2b): given a set of keypoints in two images, the keypoints are matched to find the keypoint *correspondences*. A keypoint in each image is typically identified as a corresponding pair based on the similarity (distance) between their descriptors which may be measured quantitatively using a number of methods — these will be discussed in chapter 3. Key to the success of finding correct keypoint correspondences is the ability to both detect the same keypoints in each image and to describe them correctly via their descriptors. Additionally, there must be sufficient overlap between the views or else it may not be possible to find keypoint correspondences.
3. Use keypoint correspondences to estimate camera egomotion: assuming that a set of correct keypoint correspondences between views has been found (sparse optical flow), this information is used to estimate the camera egomotion using fundamentals of two-view geometry [95] where a number of methods will be discussed in chapter 5. Two factors which have the potential to limit the accuracy

of the egomotion estimate, which will be discussed in this chapter, are the ability to resolve the magnitude of the translation (scale ambiguity), and to reliably decouple rotational and translational motion.

4. Integrate estimates of camera egomotion (figure 1.2c): by integrating the estimates of the camera egomotion, the pose (location and orientation) of the camera can be found with respect to some initial pose. It is important to find accurate estimates of camera egomotion since long-term integration of egomotion errors can result in large accumulated errors in the estimate of camera pose [127].

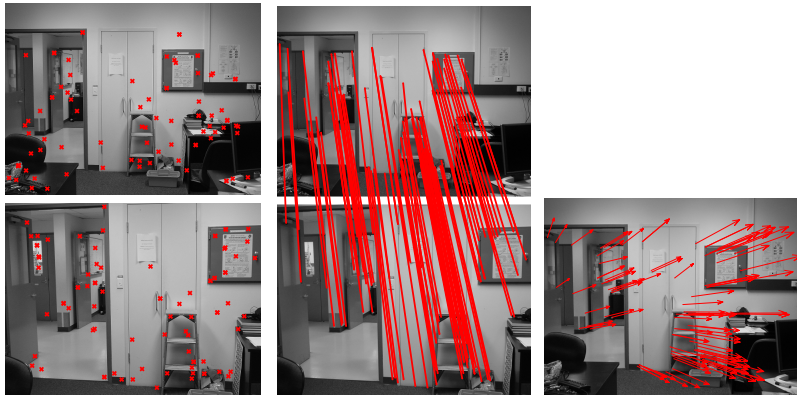
As mentioned in the third step, there is a scale ambiguity in the magnitude of the translational component of the camera egomotion found using monocular image sequences (applies to both perspective and wide-angle images). This ambiguity occurs since an image is simply a 2D representation of the 3D environment whereby only the direction and not the distance to a scene point can be found. For example, slow motion with near objects appears the same as fast motion with distant objects where both will produce similar sparse optical flow. One solution used to resolve the correct magnitude of translation is to use stereo vision where two cameras at a known position with respect to each other triangulate the Euclidean position of the scene points. This has been used successfully for this purpose for visual odometry [1, 181], but requires two cameras to be used.

Resolving the scale ambiguity using monocular camera requires more consideration. One method is to utilise knowledge of the camera configuration with respect to scene points. An example which has become commonplace for visual odometry estimation of ground vehicles is to use a subset of keypoint correspondences known to lie in the ground plane [48, 33, 198, 200]. The known height of the camera above the ground plane can then be used to resolve the magnitude of the camera translation. As is the case when using stereo vision, the correct scaled egomotion estimate can be obtained on a frame-to-frame basis, that is, without having to use any information from previous frames.

It is still possible without using constraints on the positions of the scene points to resolve the scale ambiguity in the egomotion estimates for monocular image sequences. However, this scale ambiguity can only be resolved with respect to previous egomotion estimates which leaves a single *global* scale ambiguity. For example, referring to figure 1.2c, although the relative pose between all views can be recovered there is no way to know if the camera has moved several metres or kilometres; this is the global scale ambiguity. One approach is to use *structure from motion* algorithms (as



(a) A typical image sequence acquired by a moving camera.



(b) Example set of keypoints detected in two images (left), the keypoint correspondences (middle) and the change in position of the keypoints superimposed on the first image (right) typically referred to as the sparse optical flow.



(c) Estimated relative camera pose of each image in the sequence obtained using a structure from motion algorithm.

Figure 1.2: Given some image sequence, the change in position of scene points in the image can be used to estimate the camera egomotion between views. The pose of the camera with respect to some starting pose can be found by integrating successive egomotion estimates, a process referred to as visual odometry. The visual odometry estimate in this examples was found using a structure from motion algorithm — although the relative pose between views has been found, there is no way to know if the camera has moved several metres or kilometres using only the monocular image sequence.

used in figure 1.2c), where for a given estimate of the camera egomotion, the positions of the scene points can be reconstructed from the positions of the keypoint correspondences in the images. Assuming that any of the keypoint correspondences in following frames are associated with these points, they can be used to both assist in the estimation of the camera egomotion and resolve the relative scale between successive egomotion estimates [181, 218]. Unlike methods which use stereo vision or constraints on the positions of scene points, structure from motion approaches require that information from previous frames in the sequence be retained.

More recently simultaneous localisation and mapping (SLAM) has been employed for vision-based localisation, including visual odometry, which incrementally builds a map of the operating environment as the camera moves. This map is then used for localisation, where the estimate of the camera location is found with respect to all known points in the map in a probabilistic manner [43]. Unlike structure from motion approaches, SLAM has the capability for loop closure. Loop closure is the ability to detect that the vehicle has returned to a previously mapped region of the environment and re-localise itself with respect to this map. However, similarly to structure from motion approaches, there is still a global scale ambiguity in localisation estimates as only the relative Euclidean coordinates of scene points can be found during the mapping phase. SLAM is an active area of research in computer vision [166, 121, 192, 193, 42, 43, 224, 150, 238] and Davison et al [58] have demonstrated real-time (30Hz) implementations [59]. Unfortunately, as noted by Dailey and Parnickun [53] most methods are suited to relatively small, structured environments and most successful when the camera remains primarily in the mapped environment [43].

Assuming for now that the scale ambiguity can be reliably resolved between egomotion estimates, there is still the problem of disambiguating the components of the rotation and translation in the egomotion estimate. This ambiguity is well documented in the literature for narrow field of view perspective cameras, and it can have a great effect on the accuracy of egomotion estimates [89]. Integrating inaccurate estimates of camera egomotion therefore limits the accuracy of localisation estimates [127], as previously discussed.

To illustrate this ambiguity, refer to figure 1.3a which shows a camera centred at the origin of the world coordinate frame of reference observing some scene.  $R_x, R_y$  and  $R_z$  denote rotations about the  $x, y$  and  $z$  axes respectively. Consider the two cases where the camera either rotates by some small angle about the  $y$  axis,  $R_y$ , or translates in the direction of the  $x$  axis,  $t_x$ . For a unit sphere centred at the origin of the camera

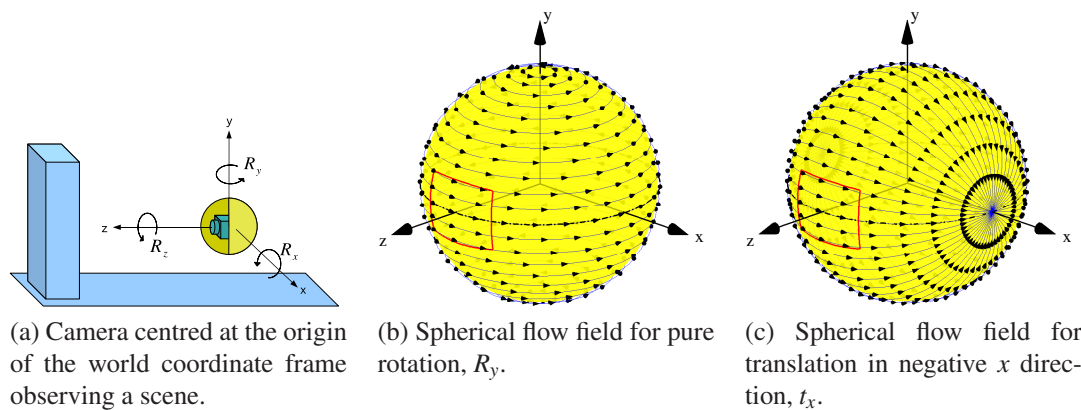


Figure 1.3: Camera centered at the origin of the world coordinate frame observing a scene. For the case of translation, the spherical flow fields intersect at antipodal points which are the centre of expansion and contraction. For this example, the camera's principal (optical) axis is the  $z$  axis.

coordinate frame, the optical flow of scene points between the views would follow the patterns shown on the view sphere in figures 1.3b and 1.3c for the rotation and translation respectively; these are often referred to as the spherical flow fields [89]. The red lines in each figure indicate the effective field of view of a typical perspective camera. It is evident from the figures that the spherical flow fields appear very similar within the field of view of the perspective camera. This similarity is illustrated using a real perspective camera in figure 1.3 which shows the sparse optical flow found for a small change in camera rotation  $R_y$  (top row) and a small change in camera translation  $t_x$  (bottom row). It can be seen from this figure that both produce extremely similar results. For the purposes of later discussions, observe that the spherical flow field for the translation  $t_x$  in figure 1.3c intersects at antipodal points on the sphere. These antipodal points are the centre of expansion and contraction.

To further illustrate this similarity, and to consider how it may be resolved, define  $\mathbf{X} = (x, y, z)^T \in \mathbb{R}^3$  as the Euclidean position of a scene point in the camera coordinate system. For a perspective projection, a scene point projects to a pixel coordinate  $\mathbf{u} = (u, v)^T$  in the image plane by

$$\begin{bmatrix} u \\ v \end{bmatrix} = \frac{f}{z} \begin{bmatrix} x \\ y \end{bmatrix} \quad (1.1)$$

where  $f$  is the focal length of the camera — a more detailed description of perspective projection will be presented in chapter 2. Given the estimate of the change in position  $\dot{u}, \dot{v}$  of the keypoints between two images, Hutchinson, Hager and Corke [105] derive the image Jacobian  $\mathbf{J}$  which describes the apparent motion of keypoints  $(\dot{u}, \dot{v})$  in the image plane for small changes in camera translation  $\mathbf{t} = (t_x, t_y, t_z)^T$  and small changes



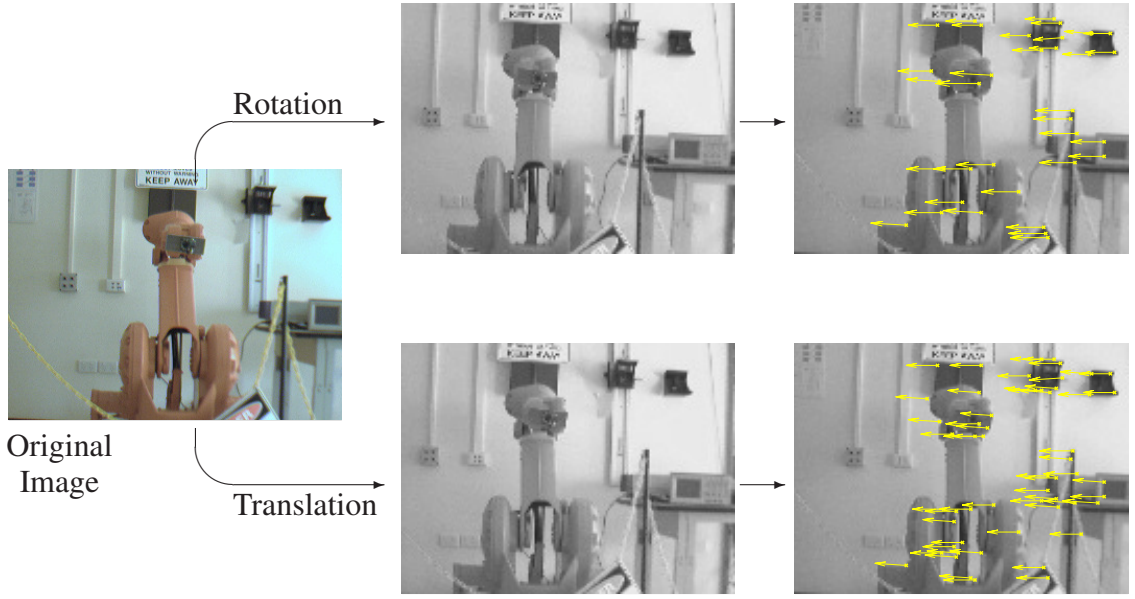


Figure 1.4: Similarities in optical flow produced for a change in camera rotation  $R_y$  (top row) and a change in camera translation  $t_x$  (bottom row). The rightmost column shows the change in position of the keypoints (sparse optical flow) superimposed on the original image.

in camera rotation  $R_x(\omega_x), R_y(\omega_y)$  and  $R_z(\omega_z)$  about the  $x, y$  and  $z$  axes respectively, where  $\omega_x, \omega_y$  and  $\omega_z$  are the changes in angles:

$$\begin{bmatrix} \dot{u} \\ \dot{v} \end{bmatrix} = \mathbf{J} \begin{bmatrix} t_x \\ t_y \\ t_z \\ \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} \quad (1.2)$$

where

$$\mathbf{J} = \begin{bmatrix} \frac{f}{z} & 0 & \frac{-u}{z} & \frac{-uv}{f} & \frac{f^2+u^2}{f} & -v \\ 0 & \frac{f}{z} & \frac{-v}{z} & \frac{-f^2-v^2}{f} & \frac{uv}{f} & u \end{bmatrix} \quad (1.3)$$

The ambiguity between rotation and translation for small motion is evident from equations 1.2 and 1.3. For a camera constrained only to translation  $t_x$  and rotation  $R_y(\omega_y)$  as in this example, the equation of motion reduces to

$$\begin{bmatrix} \dot{u} \\ \dot{v} \end{bmatrix} = \begin{bmatrix} \frac{f}{z} & \frac{f^2+u^2}{f} \\ 0 & \frac{uv}{f} \end{bmatrix} \begin{bmatrix} t_x \\ \omega_y \end{bmatrix}. \quad (1.4)$$

Observing from figure 1.4 that there is almost no optical flow vertically ( $\dot{v} \approx 0$ ), equation 1.4 shows that it is difficult to reliably decouple rotation and translation without knowing the depths  $z$  of scene points. Although an estimate for the change rotation and translation can be found, it is highly sensitive to noise in the estimate of the sparse optical flow [89] and becomes more difficult to obtain with both reductions in depth discontinuities in the scene (i.e. variations in the distance to scene points from the camera) and decreasing field of view (larger focal length  $f$  for a perspective camera) [54]. Even for the case where there exist depth discontinuities in the scene, for very small changes in rotation or translation there is minimal parallax which makes the estimate of six degree of freedom camera egomotion and the recovery of scene structure non-linear and ill-posed [179, 178]. As a consequence, many frames are required to obtain an accurate scene reconstruction using perspective cameras which can be used for egomotion estimation [100], for example using structure from motion algorithms or visual SLAM. Gluckman and Nayar [89] state that to obtain accurate egomotion estimates that are insensitive to camera orientation, either the focus of expansion or contraction should ideally be within the cameras field of view. However, they note that this is difficult to achieve using perspective cameras. With respect to figure 1.3a, the appearance of the focus of expansion or contraction would only be within the cameras field of view for predominantly translational motion  $t_z$  in the direction of the  $z$  axis.

It is necessary to consider then how the accuracy of visual odometry may be improved. Before discussing the potential advantages of using wide-angle cameras, it is of benefit to review a number of approaches adopted in the literature for perspective cameras which include:

- Resolve the position of world points (stereo vision and structure from motion)
- Obtain an estimate of motion from another sensor (sensor fusion)
- Make assumptions (or constraints) regarding world points and/or camera motion
- Increase change in pose between views (keypoint tracking and wide-baseline keypoint detection and matching)

### **Resolve the position of scene points (stereo vision and structure from motion)**

Stereo vision can be used to calculate the relative Euclidean position of scene points in the camera coordinate system as previously discussed. Not only can this be used to resolve the scale ambiguity, it has the potential to obtain more accurate egomotion estimates than those found using monocular images [182]. An early use of stereo for



egomotion estimation and scene reconstruction is given by Ho and Chung using images sequences consisting of a small number of frames [100]. They observed improved accuracy in the position of the reconstructed scene points and egomotion estimates using stereo vision when compared to the use of monocular images and a structure from motion algorithm which they found required considerably more images to achieve accurate reconstruction. This was due to the effective baseline (change in pose) between the images used for scene reconstruction. For stereo this was 50cm between the two cameras, and for the monocular scheme the change in pose between successive images in the sequence which was comparatively smaller.

Nistér, Naroditsky and Bergen [181] have also used stereo for egomotion estimation, in their case 28cm baseline separation between the cameras. In contrast, although stereo was capable of recovering the scale ambiguity in the egomotion estimate, they found that the use of the triangulated 3D points from stereo pairs were not suitable for the direct estimation of camera rotation. They attribute this to the errors in triangulation and the uncertainty in depth which has been found to grow quadratically with distance [110]. This uncertainty in the depth direction and source of error in egomotion estimates was also noted by Mallet and Lacroix [149] who were required to discard reconstructed points with high uncertainty in the depth direction to obtain more accurate 6 degree of freedom egomotion estimates for a blimp and the generation of elevation maps [110, 111]. This restricts the use of stereo vision in some respects to cases where scene points are close to the cameras, for example in indoor environments or outdoors using downward facing cameras near the ground. Similar problems in the use of stereo vision for reliable egomotion estimation, in particular rotation, were also observed in early works of Olson, Matthies, Schoppers and Maimone [190, 189] for obtaining visual odometry estimates of a planetary rover. They found that the localisation errors grew super linearly with distance travelled (using a forward facing perspective camera) caused primarily from the errors in the estimate of rotation. When they used an absolute orientation in simulations, the error growth was reduced considerably and grew more linearly. In more recent works, they have incorporated methods of multi-frame feature tracking, as will be discussed shortly, to improve accuracy of their visual odometry system with accurate results achieved over long transits [188, 145]. This is a similar strategy used by Nistér et al [181, 182] and Johnson et al [109] to achieve high accuracy visual odometry estimates over long image sequences using stereo vision.

Monocular structure from motion algorithms can also be used to recover the position of scene points and potentially improve the accuracy of camera egomotion estimates. These algorithms require both the use and calibration of a single camera. In contrast, stereo vision requires both the use and calibration of two cameras, and the

calibration of the relative pose between the two cameras. However, as previously discussed, structure from motion algorithms can only recover the position of scene points after an estimate of the camera egomotion has been found and can only resolve the scale ambiguity with respect to previous egomotion estimates. For this reason, only corresponding points that were found in previous frames can be used to assist in finding the current estimate of the camera egomotion. As noted by Nistér et al [181, 182], this has potential limitations if the position of the scene points found previously were for a small translational component in the camera egomotion; for small translation, there is great uncertainty in the depth direction when resolving the scene structure. As noted in the same work, stereo vision in contrast does not suffer from this problem and is more suitable for slow moving, or even stationary cameras (without the use of multi-frame tracking).

### **Obtain an estimate of motion from another sensor**

Fusion of vision and inertial data is a popular approach used to improve the accuracy of egomotion estimation [91, 47], where improved results can be obtained using inexpensive inertial measurement units. Another advantage of using inertial data is the ability to resolve the scale ambiguity for monocular vision-based localisation [192].

There are numerous examples of works combining vision and inertial data, including applications for unmanned aerial vehicles (UAVs) by Corke [46] and Kanade [116]. Examples can also be found for autonomous underwater vehicles (AUV's) by Dubabin, Corke and Buskey [65], and Chroust and Vincze [39] who state that fusion of vision and inertial data is suitable as vision is ideal for slow motion estimation and inertial more suited to fast motion where difficulties arise in obtaining reliable optical flow. These advantages were also noted in the works of Huster and Rock [103, 104], who fused inertial data and monocular ego-motion estimation for localisation of an AUV. Further evidence of localisation improvements using vision and inertial data is found in the works of Eustice [67] who fused inertial data with egomotion estimates obtained from a downward facing perspective camera. Corke, Lobo and Dias [47] and Lobo and Dias [141, 140] also describe advantages of using vision and inertial data, for example using inertial data to obtain a vertical reference cue from gravity when considering full 6 degree of freedom camera motion. A detailed description of methods is given by Strelow and Singh [214], who give quantitative evidence of localisation improvements through fusion of egomotion estimates from vision and inertial cues for structure from motion using both perspective and wide-angle (catadioptric) cameras. Although all the methods described have the potential to improve the overall accuracy of visual odometry, it is still of benefit to obtain the most accurate egomotion estimates

possible using vision.

### **Make assumptions (or constraints) regarding scene points and/or camera motion**

Another method which has been used in an attempt to improve the accuracy of vision-based localisation is to make assumptions regarding camera motion, for example being constrained to 3 degree of freedom planar motion, and/or the relative position of scene points.

A novel technique was proposed in a series of works by Campbell, Sukthankar, Nourbakhsh and Pahwa [34, 35] who use the sparse optical flow in different regions of the image to estimate 3 degree of freedom planar camera motion. Points above the horizon were first used to estimate the camera rotation as they are generally further away than those below and more sensitive to rotation with respect to their change in position on the image plane between views. The remaining points below the horizon were then used to find the camera translation. A similar scheme was used by Thanh et al [220], however, multiple omnidirectional cameras were used to resolve the depth of points which were then classified far or near. Again, the far points were used to first estimate rotation, then the near points used to find the translation using this estimate of rotation.

In some scenarios it is possible to assume that the scene points used to estimate the egomotion lie in the ground plane. This constraint can be used to improve the estimate of camera egomotion and has been utilised for visual odometry in numerous works [48, 33, 248, 198]. As discussed previously it has the additional advantage that the scale ambiguity in the egomotion estimate can be resolved given the known height of the camera from the ground plane. However, making assumptions regarding camera motion and/or scene points is not suitable for many practical applications, for example, a camera mounted to a robot traversing uneven terrain.

### **Increase change in pose between views**

It was mentioned previously that for monocular images sequences, structure from motion algorithms require a large change in camera pose to more accurately reconstruct the position of scene points which may be used to estimate camera egomotion and/or the relative scale between the current and previous egomotion estimates. By increasing the change in pose between two views, there is also a greater change in the appearance of the operating environment in the images and hence more distinctive sparse optical flow. Assuming that reliable keypoint correspondences can be found, then there is the potential to improve the accuracy of the egomotion estimate. For

monocular sequences, this is potentially the most practical method used to improve egomotion estimates without the use of additional sensors.

One method that can be used in an attempt to increase the change in pose between views used to compute camera egomotion is to *track* keypoints across multiple frames before computing the egomotion. In this context, tracking refers to the ability to find the same scene points in the operating environment in multiple consecutive images. Although a more formal discussion is reserved for chapter 3, tracking can be performed using classical algorithms such as Kanade-Lucas-Tomasi (KLT) [144, 209, 223], or by detecting and matching the same keypoints in multiple images [181].

Keypoint tracking using KLT was used successfully by Corke, Strelow and Singh [48] to improve the accuracy of egomotion estimation considerably compared to simple frame-to-frame estimates. The tracking approach using detection and matching of the same keypoints in multiple images was used by Matthies et al [188] for rover localisation. By tracking the 3 dimensional positions of scene points (found using stereo vision), they found a 27.7% reduction in localisation error over localisation estimates using simple simple frame-to-frame matching. A similar method of tracking was used by Nistér, Naroditsky and Bergen [181]. In their application, features were tracked over multiple frames using an additional algorithm they developed [179] to select the optimal frames to use for construction of the trifocal tensor used for Euclidean scene reconstruction and the estimation of camera egomotion. Using this approach, the three frames could be selected over several hundred frames allowing the change in pose between each to be extended. Although Neumann, Fermüller and Aloimonos [178] state that the decoupling of rotation and translation is a geometric problem which exists using both small and large baselines, the results of Nistér, Naroditsky and Bergen [181] showed very good accuracy. Agrawal and Konolige also found improvements in the accuracy of localisation using multi-frame tracking [2] when compared to their previous work using simple frame-to-frame egomotion estimates [1], where in both vision was fused with inertial and GPS data for localisation. Konolige, Agrawal and Solà have demonstrated excellent long-range localisation accuracy by fusing egomotion estimates using this approach to multi-frame tracking with inertial and GPS data [124].

Rather than track features over multiple frames, it is logical to consider why feature correspondences are not simply found over wide-baselines, where wide-baseline refers to a larger change in pose between images. Similarly to tracking, this has the potential to improve the accuracy of egomotion estimates. This may be understood by considering egomotion estimation as a signal to noise ratio problem. During keypoint detection

there is some error in the estimate of the position of the scene points in the image. By increasing the baseline separation, the change in position of the scene points between images will typically increase which improves the signal to noise ratio. However, the ability to reliably detect and match the same scene points in images separated by wide-baselines becomes difficult [222, 203]. This is due to the fact that the appearance of the same regions in the scene can appear significantly different as a result of changes in illumination and projective deformations. A number of works have addressed this problem of wide-baseline keypoint detection and matching, and a taxonomy of these methods is presented in chapter 3. In brief, they have been shown to be valuable for vision-based localisation applications. They are frequently used also for tracking purposes, where the same keypoints are detected and matching across multiple images — this is the same method of tracking described previously in [181].

To summarise, the use of keypoint tracking and/or wide-baseline keypoint detection and matching algorithms can be used to extend the change in pose between views used to compute camera egomotion. This has the potential to improve the accuracy of visual odometry. However, due to the small field of view of perspective cameras there is a limit to the change in pose between views which can be achieved before there is insufficient overlap to find enough keypoint correspondences to estimate the camera egomotion between views. For these reasons, the use of wide-angle cameras has become an increasingly popular choice for visual odometry.

### **Advantages of wide-angle camera**

Wide-angle cameras have an extended field of view, often in excess of a full hemisphere, and the images they produce exhibit extreme radial distortion, as shown in figure 1.1. Example wide-angle cameras include fisheye and catadioptric [174]. Since there are finite limits on the size of the images (number of pixels) they produce, wide-angle cameras trade spatial resolution for an increased field of view. The advantages of their use for visual odometry compared to perspective cameras are twofold. First, they are able to more accurately decouple rotational and translational motion [178, 89]. Secondly, they allow more overlap between views over large changes in camera pose which is suited for keypoint tracking or wide-baseline keypoint detection and matching.

To illustrate, consider the example in figure 1.3 where a camera changes pose by some rotation  $R_y$  or translation  $t_x$ . The appearance of the theoretical spherical flow fields for a rotation  $R_y$  and translation  $t_x$  in the image for a perspective and wide-angle camera are shown in figure 1.5. It is of interest to note that although appearance of

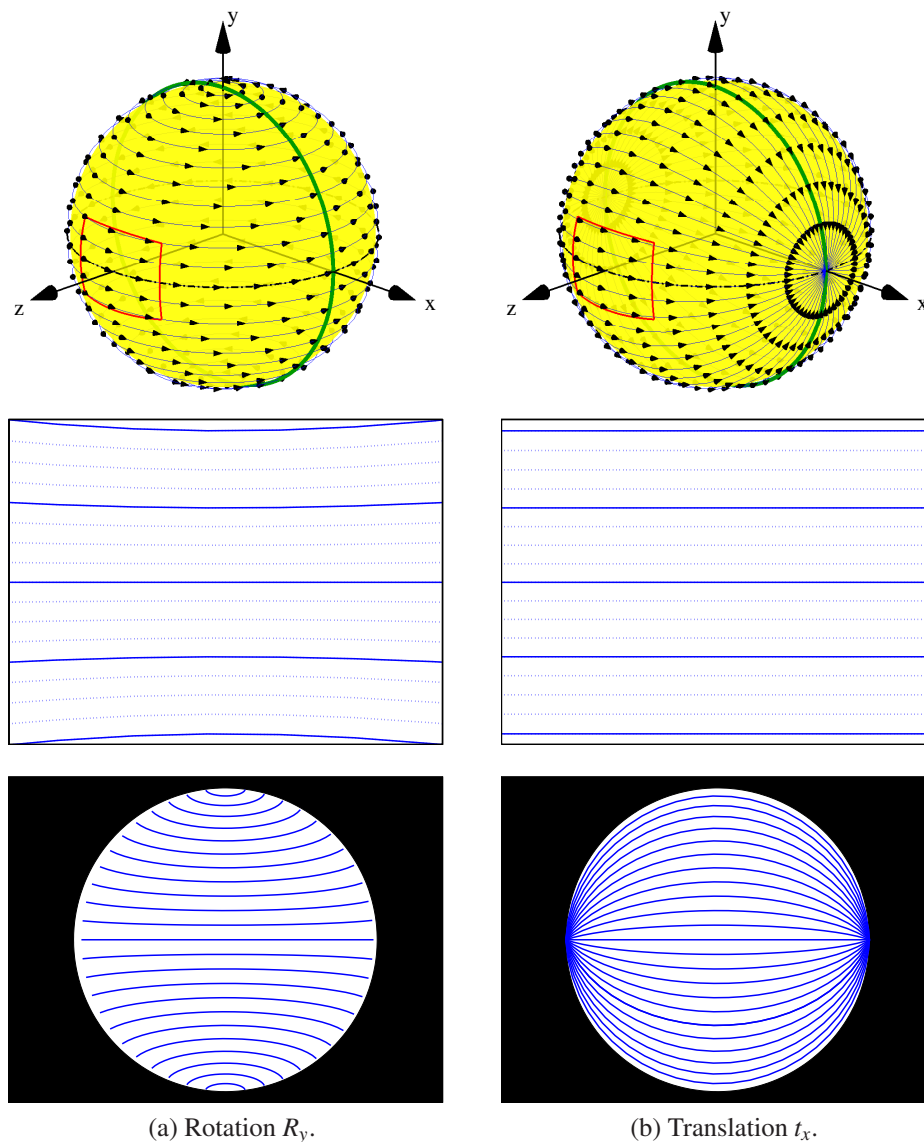


Figure 1.5: Theoretical spherical flow fields (top row) for pure rotation and translation, and their appearance in the perspective image (middle row) and wide-angle image (bottom row) for the cameras in figure 1.1. The red and green lines on the top row indicate the field of views of the perspective and wide-angle camera respectively.

these fields will vary depending on the wide-angle camera used, the sparse optical flow obtained from any wide-angle camera viewing the same scene would, in theory, project to identical spherical flow fields on the unit view sphere [89]. From simple inspection, the difference in the appearance of the spherical flow fields represented in the image planes for each motion is more apparent for the wide-angle camera.

Gluckman and Nayar state that wide-angle images are able to greatly simplify the decoupling of rotation and translation due to the visibility of either the focus of expansion or contraction in a hemispherical image, or both in a full spherical motion



field [89]. They also note that motion estimation using wide-angle images is less sensitive to noise (with respect to keypoint location) compared to perspective images since distinct motion patterns are evident, as demonstrated in figure 1.5. This observation has been validated empirically by Nelson and Aloimonos [177], and Strelow and Singh [214]. In addition, Neumann et al [178] have also considered the suitability of camera selection for structure from motion, considering both the field of view and linearity of structure from motion estimation. They describe a hierarchy of camera selection with wide-angle cameras shown to be superior to perspective cameras. The same authors also describes the advantages of spherical motion parameters obtained from omnidirectional (wide-angle) cameras compared to perspective, concluding that biological selection of spherical imaging systems for many animals is most likely the optimal choice [68].

A number of direct comparisons have been made for vision-based localisation using perspective and wide-angle cameras. Streckel et al [213] compared the accuracy of structure from motion estimates using images (of the same size) obtained from a perspective and a fisheye camera, and it was found that the results obtained using the fisheye camera were superior. Despite the fact that the fisheye cameras had a reduced angular resolution, they observed that the increased field of view permitted many more keypoints to be tracked across images, where the spatial distribution of the keypoints over the wide field of view gave improved localisation estimates. A similar comparison of a perspective and wide-angle camera was presented by Davison et al [58]. They found that more accurate localisation estimates using a visual SLAM system were obtained with a wide-angle fisheye camera compared to those with a narrow field of view perspective camera.

To recap, wide-angle cameras are potentially more suitable for visual odometry than perspective cameras. It is easier to more reliably decouple rotational and translational motion using images obtained from wide-angle cameras, and wide-angle cameras can maintain an increased overlap in the images they capture for large changes in camera pose. As is the case for any camera, the ability to accurately estimate camera egomotion from the images they produce, and hence find reliable visual odometry estimates, is dependent on the ability to detect and match the same keypoints in different images.

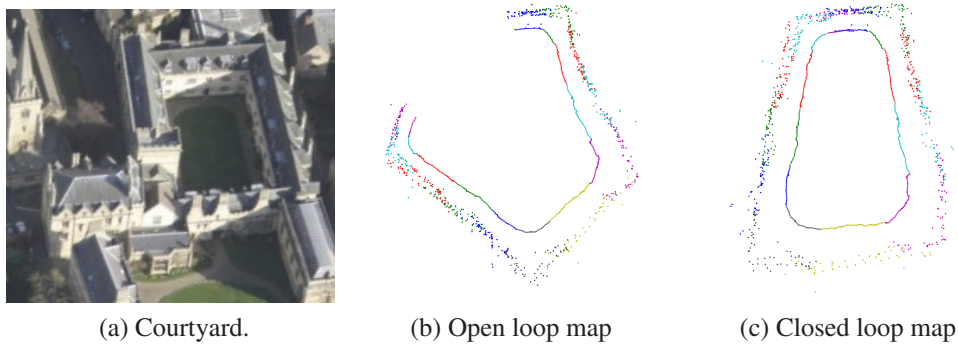


Figure 1.6: Loop closure for a mobile robot operating in an outdoor courtyard. The ‘open loop’ map (b) shows the estimate of the vehicle location and reconstructed position of scene points prior to loop closure. When the vehicle detects that it has revisited the same location it closes the loop (c). The location of the vehicle is updated and the updated ‘closed loop’ map found. These figures have been provided courtesy of Brian Williams [238], Department of Engineering Science at Oxford University.

### 1.1.2 Visual Place Recognition

A key part of SLAM is loop-closure, recognising correctly that the camera has returned to a previously visited location. This is of great benefit as the localisation estimates for a camera drift due to integrated errors in egomotion estimates. This means also that the same scene point can be mapped at two separate locations in the map. If the camera returns to the same location and the corresponding scene points in the current and previous map can be found, the camera can both re-localise itself with respect to the previous map and update the global map (i.e. ‘close the loop’), as illustrated in figure 1.6. There are a number of different methods used to detect loop-closure [238] which will be discussed in chapter 5. Among these, appearance-based methods have been shown to provide good results and robustness [51]. They detect potential loop closure in the space of appearance by comparing the similarity of the image content in the current image to all those images at previous locations in the transit. These methods can also be used for place recognition within simple topological maps, for example simply identifying that a camera has returned to some room visited previously. It is more appropriate to then refer to this process using the more general term *visual place recognition*.

As discussed, appearance based methods compare the similarity of image content. This image content can be based on a global representation of the image, for example using wavelet image decomposition [225] or colour histograms [194]. Alternatively, and more commonly, local image content is used where the image is described by information derived from the set of keypoints detected in the image [202, 79, 154, 51,



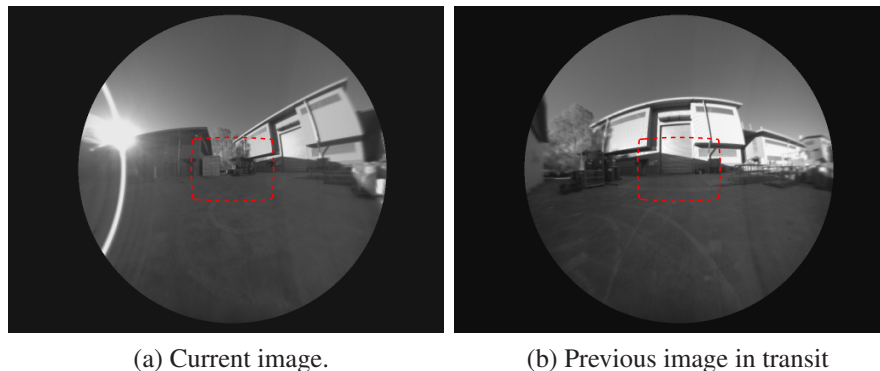


Figure 1.7: The increased field of view of wide-angle images provides advantages for visual place recognition. This is due to the ability to maintain sufficient overlap between images in the same region of the operating environment subject to large changes in camera pose. The red lines superimposed on each of the images is the approximate field of view of a typical perspective camera. Notice that for the perspective camera there is minimal overlap.

52, 38, 4, 69, 99, 107], although combinations of global and local methods have been considered [196]. For local image description, many methods utilise the concept of visual words introduced by Sivic and Zisserman in [210] for image retrieval. Methods using visual words allow the similarity of an image to be compared to all those in a large database of images (i.e. all previous images in the transit) efficiently and quickly, for example using the vocabulary tree algorithm of Nistér and Stewénus [183].

It is of interest to consider here the suitability of camera selection for visual place recognition. As was the case for visual odometry, there is an advantage in the use of wide-angle cameras as a result of their increased field of view which has seen them used for visual place recognition by Andreasson et al [3], Booij et al [22] and Ullah et al [231]. To highlight this advantage, figure 1.7 shows two images of the same scene taken by a wide-angle fisheye camera at different viewpoints — the camera has returned to a previously visited area of the operating environment. The red superimposed lines indicate the effective field of view of a typical perspective camera. It can be observed from the figure that even though there is a substantially large difference in the camera pose between views, in particular rotation, there is still a large overlap between the wide-angle images. If a perspective camera were used, from the effective field of view shown it is evident that there would be minimal overlap. The chances of reliably recognising that the camera has returned to a previously visited location for this example would therefore be greater for the wide-angle camera.

To conclude, wide-angle cameras have potential advantages when compared to perspective cameras for vision based localisation, including visual odometry and visual

place recognition. For both, the ability to detect and match the same keypoints in different images subject to large changes in camera pose is a necessary criteria. This is an image processing problem which relates specifically to wide-baseline keypoint detection and matching. In general, wide-baseline keypoint detection and matching algorithms have been designed for use with perspective cameras. Their suitability for use with wide-angle images or new algorithms designed specifically for wide-angle images are areas of research which have attracted little attention.

## 1.2 Wide-Angle Image Processing

In general, most implementations of wide-baseline keypoint and matching are applied directly to wide-angle images without considering or accounting for the extreme radial distortion in the images [200], or simply to the rectified panoramic representation of the original image [22, 233, 172]. Although these approaches have worked with some success, there is little evidence in the literature of more suitable methods designed for, and evaluated with, wide-angle images. One notable exception is the work presented by Briggs et al [26] who considered wide-baseline matching with wide-angle images. However, their method was suited only to 1 dimensional rectified panoramic images which is not suited to generalised 6 degree of freedom camera motion. Although not developed for wide-baseline keypoint detection and matching, Daniilidis et al [56] formulated image processing as an operation on the sphere. The importance of this approach, which will be discussed in detail in chapter 3, is that it accounts for the radial distortion in the image.

As the existing methods of wide-baseline keypoint detection, description and matching have worked with success with perspective images, it seems logical to consider how they can be adapted to suit wide-angle images, that is, to account for the radial distortion in the image. One way to do this would be to use the approach of Daniilidis et al [56] and reformulate these existing algorithms as operations on the sphere.

## 1.3 Research Questions and Methodology

It is proposed that the use of wide-angle images is suitable for use in applications relating to vision-based localisation of mobile robots, including visual odometry and visual place recognition. This thesis addresses the following research question:

*‘Can a method of wide-baseline keypoint detection and matching be found suited for use with any wide-angle camera for vision based localisation, including visual odometry and visual place recognition?’*

This gives rise to a number of additional questions that are addressed in this thesis:

1. What types of wide-angle cameras exist, what are the methods of modelling the distortion in the image, and how can they be calibrated to resolve the camera’s intrinsic parameters?
2. What existing methods of wide-baseline keypoint detection are suitable for vision-based localisation?
3. What are the limitations of applying these existing methods to wide-angle images and how, if possible, can they be adapted to suit wide-angle images?
4. Can an alternative to existing wide-baseline keypoint detection and matching algorithms be developed which is more suitable for use with wide-angle images?
5. Assuming a suitable alternative can be found, can it be used to obtain accurate visual odometry estimates for a mobile robot, and if so, what are the effects of increasing the change in pose between views with respect to accuracy?
6. Assuming again that a suitable alternative can be found, can it be used to obtain robust visual place recognition using wide-angle cameras?

To answer these questions, an extensive review of literature is first conducted which identifies various classes of wide-angle cameras and methods used to both model and calibrate for the radial distortion in the images. A novel and generic calibration algorithm is developed which is used to select, from a range of empirical choices proposed in the literature, a suitable model for a fisheye camera used in this work. This model assumes that the fisheye camera has a single effective viewpoint (central projection) where the model describes a mapping from rays in space to the image plane. As with all central projection cameras, this model can be used to map the wide-angle image to the unit view sphere. This is followed with an extensive review of wide-baseline keypoint detection and matching algorithms. This includes the class of algorithms based on the scale-space framework and a number of alternate approaches. The suitability of these methods is considered with respect to vision based localisation where the Scale-Invariant Feature Transform (SIFT) proposed by Lowe [142], which is used extensively throughout the literature for vision based localisation, is identified as a prime candidate.

As with many image processing algorithms, most wide-baseline methods including SIFT have been designed for use with perspective images. The limitation of applying these methods directly to wide-angle images is identified as their inability to account for the radial distortion in the image. It is proposed that SIFT, and any other similar algorithms based on the scale-space framework, can be reformulated as an image processing operation on the unit sphere; since any central projection wide-angle image can be mapped to the sphere, this approach is suited for all central projection wide-angle cameras. This general approach was inspired by the work of Daniilidis et al [56] who proposed and demonstrated image processing as an operation on the sphere for wide-angle images, and Bülow's work relating to scale-space theory for functions on the sphere [31, 30].

A number of variants of the SIFT algorithm, which are the primary contributions of the thesis, are developed for use with the images obtained with calibrated central projection wide-angle cameras. A wide-angle image is considered as a function on the sphere, and the underlying scale-space for a wide-angle image is therefore defined as the solution of the spherical heat diffusion equation which was solved by Bülow [31]. These variants are based on alternate ways of implementing spherical diffusion: in the spherical Fourier domain (termed spherical SIFT) and a more efficient approximate diffusion on the stereographic image plane (termed parabolic SIFT). Suitable means for obtaining keypoint descriptors are formulated which considers keypoint support regions defined on the sphere. For spherical SIFT, factors relating to bandwidth selection are considered where a suitable anti-aliasing filter is developed and validated through experiments. The new methods are compared to a direct application of SIFT (SIFT is applied directly to the wide-angle without making any account for the radial distortion) through extensive experiments using synthetic and real wide-angle images (fisheye and catadioptric). A quantitative comparison of the relative performance is made using the percentage correlation of keypoint correspondences between images and recall versus 1-precision statistics. The parabolic SIFT (pSIFT) algorithm is identified as a better alternative to SIFT for use with wide-angle images and has similar computational expense.

The pSIFT algorithm is used to obtain visual odometry estimates for both a fish-eye and catadioptric image sequence. The keypoint detection algorithm is validated through extensive experiments where it is shown that accurate visual odometry estimates can be obtained. This includes a visual odometry algorithm using a ground plane constraint, and a generalised structure from motion algorithm. A variable frame-rate algorithm is used similar to that proposed by Mouragnon et al [170] to track keypoints through multiple frames in order to increase the change in pose between views used to

compute the camera egomotion. The visual odometry estimates using this method are compared to those using every frame through extensive experiments. Finally, experiments using a real fisheye image sequence taken along a 4.4 kilometre path are used to validate the new keypoint detector for visual place recognition.

## 1.4 Contributions

The contributions of the thesis include:

1. Review of image formation with wide-angle cameras and the development of a novel and generic calibration algorithm which effectively calibrates for the camera intrinsic parameters on the unit sphere using multiple images of a planar checkerboard pattern.
2. Review of wide-baseline keypoint detection and matching algorithms and their suitability for both vision based localisation and use with wide-angle images.
3. Development of two variants of SIFT, termed spherical and parabolic SIFT, using the underlying scale-space as the solution of the heat diffusion equation on the sphere which are both suitable for use with any central projection wide-angle camera. This includes the formulation of an approximate spherical diffusion operation for parabolic SIFT which operates on the stereographic image plane.
4. Analysis of bandwidth selection when obtaining scale-space images via convolution in the spherical Fourier domain (spherical SIFT) and the development of a suitable anti-aliasing filter which may be used to counterfeit aliasing.
5. Systematic experiments comparing the performance of spherical SIFT and parabolic SIFT to SIFT (operating directly on the wide-angle images without accounting for the image distortion) using extensive synthetic and real wide-angle images.
6. A coordinate weighting scheme suitable for egomotion estimating using the Direct Linear Transform (DLT) and a ground plane constraint with parabolic SIFT keypoints. The weighted spherical coordinates of keypoints are used to estimate camera egomotion using the DLT. This weighting estimates the uncertainty of a keypoints position on the sphere relative to its uncertainty in position found in a wide-angle image during detection. Using a real wide-angle image sequence, the accuracy of the visual odometry estimates found using this weighting scheme

are shown to be more robust than those found using simply the normalised homogeneous coordinates of the calibrated keypoints.

7. A comparison of visual odometry estimates found using a fixed frame-rate and variable frame-rate tracking algorithm.
8. Development of a word *reliability* metric which is incorporated into the “Video Google” algorithm [210] for visual place recognition. Using this metric, robust place recognition using a wide-angle (fisheye) camera and the new parabolic SIFT keypoint detector is demonstrated through experiments.

## 1.5 Structure of Thesis

**Chapter 2** presents the foundations for image formation with wide-angle cameras and methods of calibration. This includes the development of the novel calibration algorithm and its use in selecting a suitable camera model for a wide-angle camera.

**Chapter 3** reviews methods for detecting, describing and matching keypoints between views. This includes some ‘classical’ methods suited for small changes in camera pose between views, and wide-baseline methods suited for large changes in camera pose between views. The wide-baseline algorithms include those based on the scale-space framework, and a range of alternate methods. From this review, the suitability of these wide-baseline methods is considered with respect to wide-angle images and a method is selected which may be potentially modified for used with wide-angle images.

**Chapter 4** develops the spherical SIFT and parabolic SIFT algorithms that are suited for keypoint detection and description with central projection wide-angle cameras. The algorithms are variants of the Scale-Invariant Feature Transform and effectively implement image processing as a function on the sphere. One of the key components of the algorithms is the definition of scale-space for wide-angle images as the solution of the spherical heat diffusion equation on the sphere. The methods are compared to the standard SIFT algorithm using artificial and real wide-angle images in multiple experiments.

**Chapter 5** applies the new parabolic SIFT keypoint detection algorithm to visual odometry using real wide-angle image sequences obtained with a fisheye and

catadioptric camera. The visual odometry estimates are found using various constraints on camera motion and the relative position of scene points. The accuracy of visual odometry estimates are compared using both a fixed frame-rate (every consecutive image in the sequence) and a fixed frame-rate which increases the change in pose between views used to compute the camera egomotion. As part of this, a coordinate weighting scheme for parabolic SIFT keypoints is developed for egomotion estimation using the Direct Linear Transform (DLT) and a ground plane constraint. This follows with the application of the parabolic SIFT keypoint detector for visual place recognition using a real wide-angle (fisheye) image sequence. This includes the detailed implementation of the word reliability metric which is validated through experiments.

**Chapter 6** presents the conclusions for the work addressing each of the research questions in this chapter and proposes a number of directions for future work.





## Chapter 2

# Wide-Angle Image Formation and Calibration

*For computer vision applications, image formation is typically considered as a geometric mapping of scene points to the image plane. This mapping is defined by a parametric camera model which varies depending on the wide-angle camera used. For applications to vision-based localisation, for example structure from motion, it is necessary to find a suitable camera model and calibrate to find the model parameters. This chapter presents a review of image formation for a number of classes of wide-angle cameras, including catadioptric and dioptric, and a range of calibration techniques. A novel and generalised calibration algorithm is then developed suited for use with any central projection camera using multiple images of a planar checkerboard pattern. This includes a robust means for finding the checkerboard intersections (grid points) used for, and operating in parallel with, calibration. A fisheye camera used extensively in experiments throughout the thesis is calibrated using this algorithm, and a suitable model is selected empirically from a range proposed in the literature. An important observations made in this chapter is that the image obtained with any central projection camera can be mapped to the unit view sphere. The significance of this becomes apparent in later chapters when deriving a generalised means for wide-baseline keypoint detection and matching with wide-angle images.*

## 2.1 Introduction

A digital camera used in computer vision provides as output a discrete two-dimensional image representative of the surrounding environment. Typically the values at each element of the image (pixels) are light intensity measurements, either colour or greyscale. For practical purposes, image formation can be considered as a geometric mapping of scene points  $\mathbf{X} = (X, Y, Z)^T \in \mathbb{R}^3$  defined with respect to the world coordinate frame of reference to pixel locations  $\mathbf{u} = (u, v)^T \in \mathbb{R}^2$  in the image plane, where there exists an inverse mapping from points in the image plane to rays in space. This mapping is frequently described in the literature by a parametric camera *model*; a function which defines for each pixel in the image plane either the corresponding ray in space (ray-based models), or the corresponding point in the perspective image plane (pinhole based models). It is critical to correctly model the geometric process of image formation for many vision based localisation tasks. The camera must also be calibrated whereby both the pose of the camera with respect to the world coordinate frame and parameters of the camera model are found. Structure from motion algorithms, for example, require the camera model and calibration parameters to be found for egomotion estimation and the inverse mapping of points in the image plane to rays in space for scene reconstruction.

Recall from chapter 1 that image formation with narrow field of view cameras is typically a perspective projection of scene points to the image — they are referred to as perspective cameras. Perspective cameras project straight lines in space to straight lines in the image [95], and the image obtained with a perspective camera is considered to be *undistorted* due to this property. Wide-angle cameras in contrast have an increased field of view, and although image formation with some wide-angle cameras is still described by perspective projection, for those with a near hemispherical field of view this is not the case. The wide-angle camera which captured the image in figure 1.1b, for example, has in excess of a hemispherical field of view — a camera is omnidirectional if the field of view is in excess of a full hemisphere [162]. The image is considered to be distorted as straight lines in space appear curved in the image<sup>1</sup>. This distortion can include components in both the radial and tangential directions in the image, both of which will be discussed in section 2.2.2.

Image formation with perspective and wide-angle cameras is reviewed in section 2.2. Although most perspective cameras are generally not considered wide-angle, many pinhole based models for fisheye cameras describe image formation as a map-

---

<sup>1</sup>The exception is for straight lines which pass directly through the centre of distortion

ping from the fisheye to perspective image plane — pinhole models are discussed in section 2.2.2.3. This review of image formation is restricted to parametric camera models which describe the process of image formation. Although non-parametric models have been developed [98], parametric models are used far more extensively in the literature, and they have been used successfully to accurately model an extensive range of cameras. An important observation made from this review is that the image obtained with any central projection camera (perspective and wide-angle) can be mapped to the unit view sphere centred at the camera's single effective viewpoint (centre of projection). This has significance in later chapters where suitable means for wide-angle image processing are considered.

A review of camera calibration algorithms follows in section 2.3. Many calibration algorithms are specific to a single camera model. A novel algorithm is developed in section 2.4 that can be used to calibrate any central projection wide-angle camera and takes as input multiple images of a planar checkerboard pattern. As part of this calibration algorithm, a robust means for finding the checkerboard intersections (grid points) required for calibration is developed. The fisheye camera used extensively in experiments throughout this work is then calibrated using this algorithm using a number of the models proposed in the literature — the image in figure 1.1b was obtained with this camera. These results are used to select empirically the most suitable model for this camera.

## 2.2 Image Formation

### 2.2.1 Perspective Cameras

The projection of a scene (world) point  $\mathbf{X}$  to a point at a pixel position  $\mathbf{u}$  on the perspective image plane is defined by the ideal pinhole model. The position  $\mathbf{u}$  is defined as the intersection of the image plane with the ray from point  $\mathbf{X}$  which passes through the viewpoint (pinhole)  $\mathbf{0}$ , as illustrated in figure 2.1. The camera's principal axis is normal to the image plane, passes through the pinhole  $\mathbf{0}$ , and intersects the image plane at the principal point  $\mathbf{u}_0$ . For the camera centred at the origin of the world coordinate frame of reference and whose principal axis is aligned with the  $z$  axis, similar triangles

define the mapping  $\mathbf{X} \mapsto \mathbf{u}$  as

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} \mapsto \begin{pmatrix} fX/Z \\ fY/Z \end{pmatrix}, \quad (2.1)$$

where  $f$  is the camera focal length with units of pixels to be dimensionally correct.

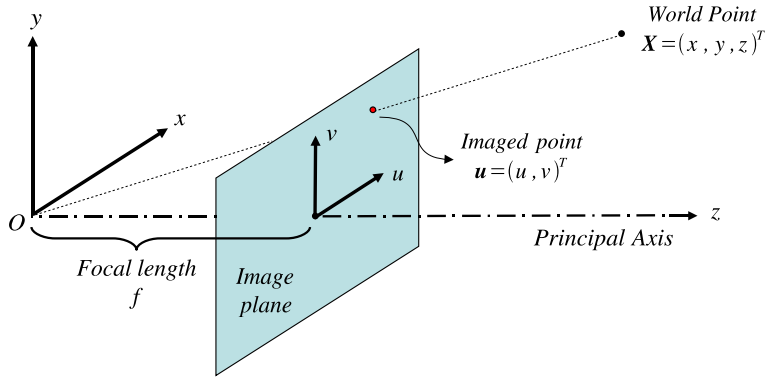


Figure 2.1: Image formation with a perspective camera defined by the ideal pinhole model. The camera centre  $\mathbf{C} = (C_x, C_y, C_z)^T$  is located at the origin  $\mathbf{0} = (0, 0, 0)^T$  of the world coordinate frame, and the camera's principal axis is aligned with the  $z$  axis of the world coordinate frame. The camera's principal axis intersects the image plane at the principal point.

In practice, a more generalised pinhole model is used which defines the transformation of scene points  $\mathbf{X}$  to their associated points  $\mathbf{u}$  in the image plane by the camera matrix  $P$ :

$$\mathbf{u} = P\mathbf{X}, \quad (2.2)$$

where both  $\mathbf{X}$  and  $\mathbf{u}$  are represented by their homogeneous 4 and 3 vector coordinates respectively. The process of image formation defined by the camera matrix  $P$  is dependent on the camera intrinsic values (the parametric camera model) and the camera extrinsic parameters  $R \in SO(3)$  and  $\mathbf{C} = (C_x, C_y, C_z)^T \in \mathbb{R}^3$ . Here,  $R$  is the orientation of the camera with respect to the world coordinate frame of reference, and  $\mathbf{C}$  is the position of the camera centre (pinhole) in the world coordinate frame of reference. The camera matrix  $P$  can then be decomposed as

$$P = K R [I_{3 \times 3} | -\mathbf{C}], \quad (2.3)$$

where  $I_{3 \times 3}$  is the  $3 \times 3$  identity matrix. The camera intrinsic variables are defined by

the camera calibration matrix  $K$  [96]:

$$K = \begin{bmatrix} A & \mathbf{u}_0 \\ \mathbf{0}^T & 1 \end{bmatrix} = \begin{bmatrix} f_u & s & u_0 \\ 0 & f_v & v_0 \\ 0 & 0 & 1 \end{bmatrix}, \quad (2.4)$$

where  $A$  is an affine transform. This affine transform models the imperfections of many CCD sensors, including separate focal lengths  $f_u$  and  $f_v$  to account for uneven scaling in the  $u, v$  image coordinates respectively, and the parameter  $s$  to model shearing. The parameter  $\mathbf{u}_0 = (u_0, v_0)^T$  is the pixel position in the image of the principal point (point of intersection of the camera's principal axis and image plane).

Perspective cameras have less than a hemispherical field of view, being able to only image scene points  $\mathbf{X}$  in front of the camera. Assume that the camera is centred at the origin of the world coordinate frame of reference where  $\mathbf{C} = (0, 0, 0)^T$  and  $R = I_{3 \times 3}$  is the  $3 \times 3$  identity matrix. A point  $\mathbf{X} = (X, Y, 0)^T$  for example projects to a point with a homogeneous coordinate  $\mathbf{u} = (u, v, 0)^T$ , which is a point at infinity in the image plane. The homogeneous coordinates  $\mathbf{u} = P\mathbf{X}$  and  $\mathbf{u}' = P\mathbf{X}'$  are also projectively equivalent for the scene point  $\mathbf{X}$  in front of the camera and the scene point  $\mathbf{X}' = -\mathbf{X}$  behind the camera. Perspective projection is therefore unable to define the projection of a scene point behind the camera to a unique position in the image.

### 2.2.2 Wide-Angle Cameras

Wide-angle cameras can be classified as catadioptric or dioptric. The term catadioptric is used to denote an imaging system using a reflective surface and either a perspective or orthographic camera [175]. The name originates from dioptrics, which is the science of refracting elements (lenses), and catoptrics, the science of reflective surfaces (mirrors). Although a number of works have considered catadioptric cameras with multiple reflective surfaces [176, 173], those with a single reflective surface are most commonly used for vision-based localisation. Wide-angle dioptric cameras attain an increased field of view using a specially shaped lens; these are frequently referred to as fisheye lenses, where a camera fitted with such a lens is referred to as a fisheye camera.

As for perspective cameras, the process of image formation is dependent on the camera's intrinsic and extrinsic parameters. In the following discussion regarding image formation with wide-angle cameras, the term *camera model* is defined to mean only the camera intrinsic parameters — the position of all scene points are assumed

to be in the camera's frame of reference. The wide-angle camera models discussed in this section (excluding the pinhole-based fisheye models in section 2.2.2.3) describe the mapping of scene points to the camera's sensor. Each element on the sensor has a coordinate  $\mathbf{x}(x, y) = (x, y)^T$  with respect to the principal point which is parameterised by polar coordinates  $\mathbf{x}(r, \zeta) = (r \sin \zeta, r \cos \zeta)^T$ . A point on the sensor with a coordinate  $\mathbf{x}$  maps to a point on the image with a coordinate  $\mathbf{u}(u, v) = (u, v)^T$  by the transform

$$\mathbf{u} = A\mathbf{x} + \mathbf{u}_0. \quad (2.5)$$

Here,  $A$  is an affine transform which models uneven scaling in the  $x, y$  sensor coordinates and shearing, and  $\mathbf{u}_0 = (u_0, v_0)^T$  is the position of the camera centre. There are many definitions for the camera centre, and a detailed taxonomy is given by Willson [240] and Willson and Shafer [241]. Willson and Shafer state that the camera centre is often defined as the intersection of the optical (principal) axis of camera's lens with the sensor. However, most cameras have multiple lenses, and in the case of catadioptric cameras, combinations of lenses and mirrors. As each lens and mirror has its own optical axis, it is difficult to define any one single optical axis for the camera. Misalignment of the optical axis results in decentering (non radially symmetric) distortions in wide-angle images. In the following discussions it is assumed that wide-angle cameras have a single well defined optical axis. This axis is the camera's principal axis, which is orthogonal to the camera's sensor, intersects the camera's sensor at the origin  $\mathbf{x} = (0, 0)^T$ , and intersects the image plane at the point  $\mathbf{u}_0$ . The principal point  $\mathbf{u}_0$  is therefore considered as the centre of distortion in a wide-angle image.

### 2.2.2.1 Central vs Non-central Cameras

Before discussing catadioptric and fisheye camera models, it is necessary to define the terms central projection and non-central projection. A central projection camera has a single effective viewpoint. With respect to the pinhole model in figure 2.1, the camera is central projection as the rays from all scene points intersect at the camera's single viewpoint located at  $\mathbf{0}^T$ . A scene point  $\mathbf{X}$  therefore projects to the same coordinate  $\mathbf{u}$  in the image independent of its position on the ray (i.e.. independent of the homogeneous scale factor of  $\mathbf{X}$ ).

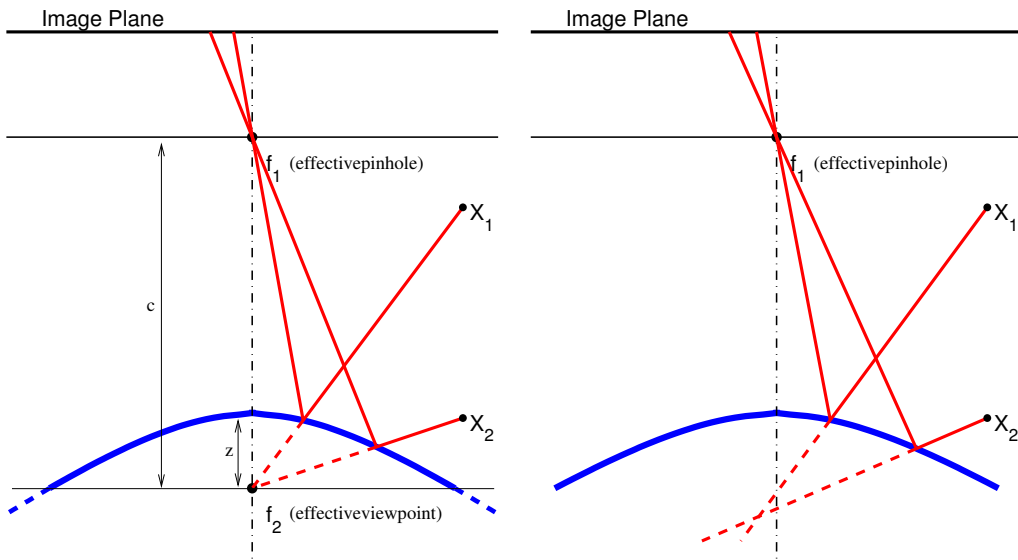
An example central and non-central projection catadioptric camera is illustrated in figures 2.2a and 2.2b respectively. For the central projection camera there is a single viewpoint at  $f_2$ . For the non-central camera there is a locus of viewpoints termed a caustic [216, 215]. As discussed by Nayar and Baker [175], central cameras are ideal

as they permit geometrically correct perspective images to be produced from regions of the image which may be used for (undistorted) human viewing and image processing. This is possible as the image obtained with any central projection camera can be mapped to the unit view sphere centred at the single viewpoint. This is illustrated in figure 2.3 for a central projection catadioptric camera and (assumed) central projection fisheye camera. Recalling that for a central projection perspective camera that the position in the image  $\mathbf{u}$  of a point  $\mathbf{X}$  is independent of its homogeneous scale factor, then the points  $X$  and  $\frac{X}{\|X\|}$  will both project to the same point  $\mathbf{u}$  in the image. As  $\frac{X}{\|X\|}$  is a point on the unit view sphere, a geometrically correct perspective image can be obtained given any wide-angle image mapped to the view sphere. However, not all regions in the wide-angle image can be mapped via the sphere to the perspective image as it has less than a hemispherical field of view. For the purposes of later discussions, define  $\eta$  as a point on the unit view sphere parameterised as

$$\eta(\theta, \phi) = \begin{bmatrix} \sin(\theta) \cos(\phi) \\ \sin(\theta) \sin(\phi) \\ \cos(\theta) \end{bmatrix}, \quad (2.6)$$

where  $\theta \in [0, \pi)$  is an angle of colatitude and  $\phi \in [0, 2\pi)$  an angle of longitude, as shown in figure 2.4.

For non-central cameras, without accounting for the locus (caustic) of viewpoints and depth of points, the reconstructed perspective image would contain some degree of parallax error [86]. The inability to produce geometrically correct perspective images from the images obtained with non-central cameras is illustrated more clearly with reference to figure 2.5. Consider two world points  $\mathbf{X}_1$  and  $\mathbf{X}_2$  and assume a unit sphere is centred at the intersection of the ray from point  $\mathbf{X}_1$  and the camera's principal axis — refer to this point as the reference viewpoint. Points  $\mathbf{X}_1$  and  $\mathbf{X}_2$  intersect this unit view sphere at positions  $\eta_1$  and  $\eta_2$  respectively which in turn map to points  $\mathbf{u}_1$  and  $\mathbf{u}_2$  in the image plane respectively. Notice that the ray  $\tilde{\mathbf{X}}_2$  from point  $\mathbf{X}_2$  does not intersect the camera's principal axis at the reference viewpoint. If a geometrically correct perspective image were to be produced, the point defined as the intersection of the sphere and the ray from  $\mathbf{X}_2$  passing through the reference viewpoint needs to be found — this can be considered as the correct coordinate for the point  $\mathbf{X}_2$  on the view sphere with respect to the reference viewpoint. However, even if the location of the ray  $\tilde{\mathbf{X}}_2$  were known with respect to the reference viewpoint, unless the position of the point on this ray is known then the correct position on the unit view sphere with respect to the reference viewpoint cannot be found. For example, the point at two different positions  $\mathbf{X}'_2$  and  $\mathbf{X}''_2$  will map to different positions  $\eta'_2$  and  $\eta''_2$  on the view sphere with respect



(a) Central projection catadioptric camera. (b) Non central projection catadioptric camera.

Figure 2.2: Central versus non-central catadioptric projection. For central projection all rays intersect at a single effective viewpoint  $f_2$ . For non-central, there is a locus of viewpoints (caustic). The point  $f_1$  for both cases can be considered as the effective pinhole. For the central camera, the distances  $c$  and  $z$  refer to the those in the derivation of central catadioptric cameras by Nayar and Baker [175] in equations 2.7 and 2.8.

to the reference viewpoint.

### 2.2.2.2 Catadioptric

#### Central Catadioptric

Nayar and Baker [175] were the first to derived the entire class of central projection catadioptric cameras with a single reflective surface, more specifically the shapes of the reflective surfaces (mirrors) and their required axial separation from the cameras. This derivation was based on the assumption of perspective image formation for the camera and specular reflection whereby the angle of incidence of a ray to the mirror is equal to the angle of reflection. They show that the surface slope of the mirrors is obtained from a quadratic first order differential equation whose solutions are

$$\left(z - \frac{c}{2}\right)^2 - r^2 \left(\frac{k}{2} - 1\right) = \frac{c^2}{4} \left(\frac{k-2}{k}\right) \text{ for } (k \geq 2) \quad (2.7)$$

$$\left(z - \frac{c}{2}\right)^2 + r^2 \left(1 + \frac{c^2}{2k}\right) = \left(\frac{2k+c^2}{4}\right) \text{ for } (k > 0) \quad (2.8)$$



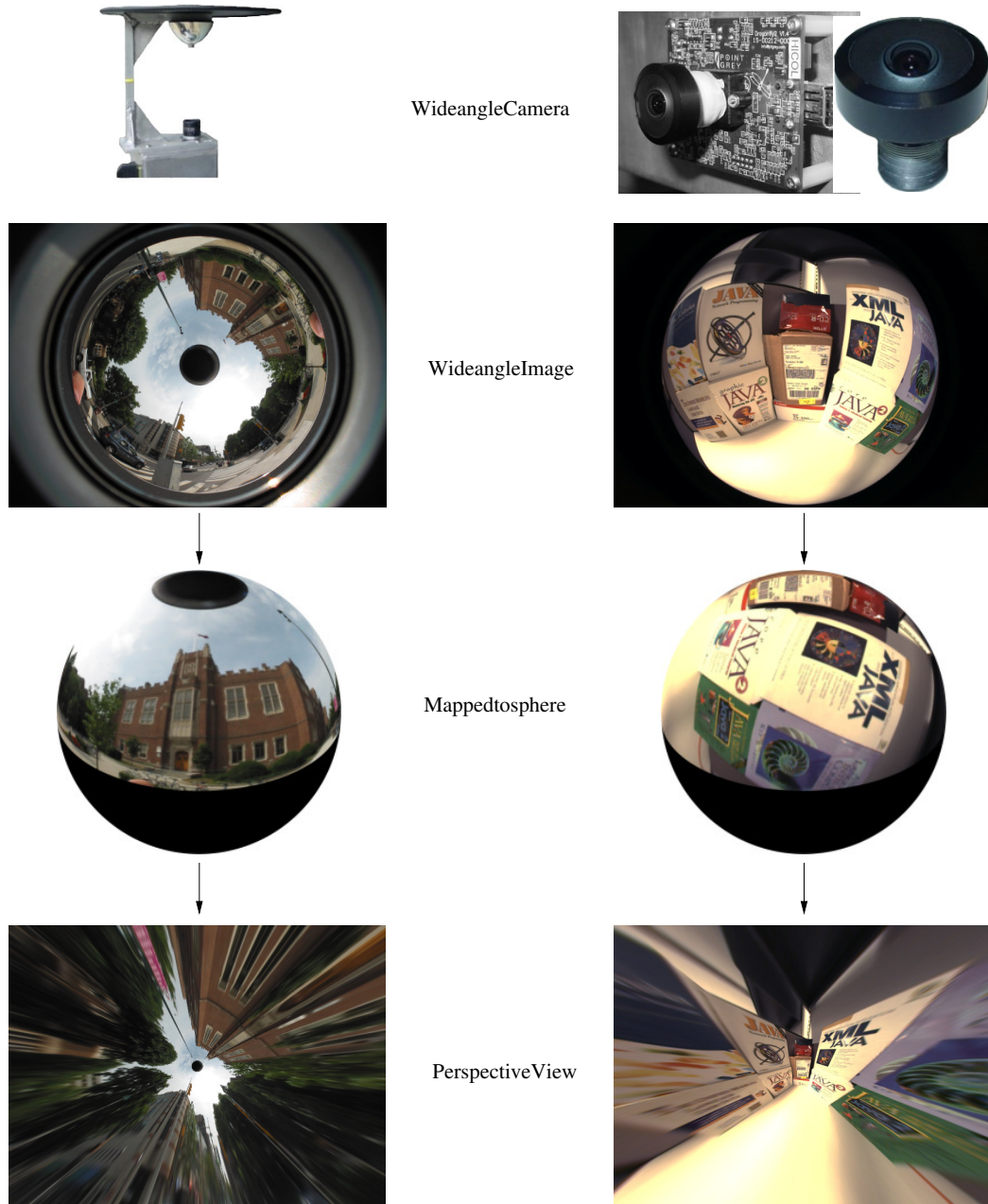


Figure 2.3: Catadioptric (left column) and fisheye (right column) cameras. The top row shows the typical camera sensors. If each camera is central projection, the image can be mapped back to the unit view sphere from which an undistorted perspective view can be obtained.

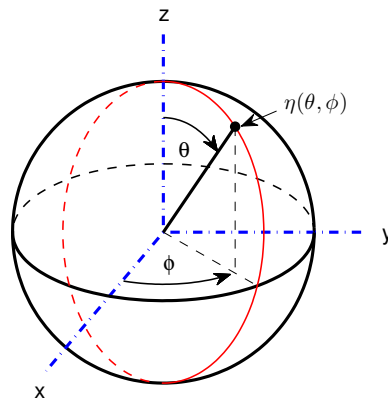


Figure 2.4: Spherical Coordinates. A ray in space originating from the centre of the sphere can be parameterised by an angle of colatitude  $\theta \in [0, \pi)$  and longitude  $\phi \in [0, 2\pi)$ . This ray intersects the unit sphere at the point  $\eta(\theta, \phi)$ .

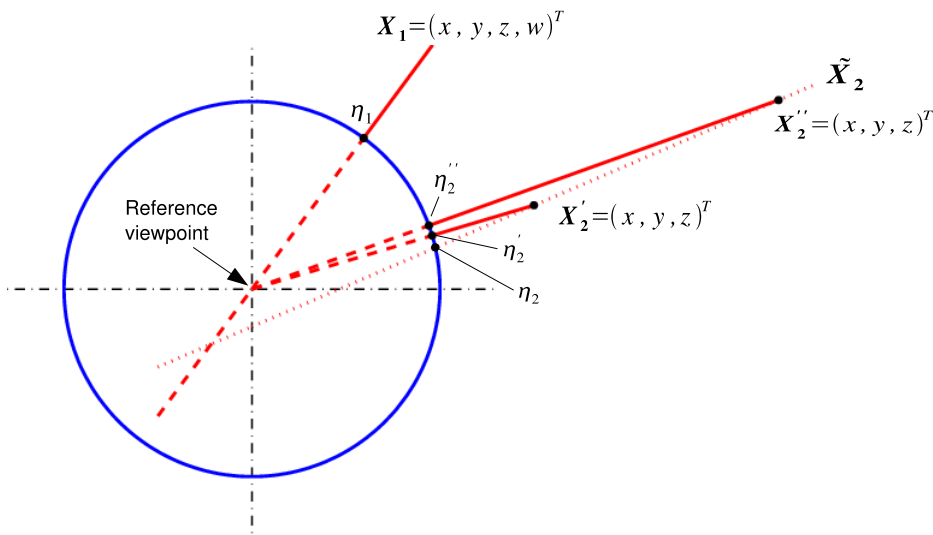


Figure 2.5: Without knowing the spherical coordinates of all world point on the the view sphere with respect to a single effective viewpoint, a geometrically correct perspective image cannot be produced. Even if the location of the ray  $\tilde{\mathbf{X}}_2$  were known with respect to the reference viewpoint, unless the position of the point on this ray is known then the correct position on the unit view sphere with respect to the reference viewpoint cannot be found. For example, the point at two different positions  $\mathbf{X}'_2$  and  $\mathbf{X}''_2$  will project to different positions  $\eta'_2$  and  $\eta''_2$  on the view sphere with respect to the reference viewpoint.

where, referring to figure 2.2a,  $c$  is the separation between the effective pinhole and viewpoint,  $k > 0$  is a constant, and  $z$  is the height of the mirror from the effective viewpoint (focus). They note from inspection of equations 2.7 and 2.8 that the solution for the entire class of mirrors are swept conic sections. A summary of the solutions determined by the values  $k$  and  $c$  is given in table 2.1<sup>2</sup>

<sup>2</sup>For a parabolic catadioptric camera, there is an orthographic projection of the rays from the mirror to the image and not perspective. Although this violates the initial assumptions made in the derivation

Condition	Equation	Mirror Type
$k = 2$ and $c > 0$	$z = \frac{c}{2}$	Planar
$c = 0$ and $k \geq 2$	$z = \sqrt{\frac{k-2}{2}} r^2$	Conical
$c = 0$ and $k > 2$	$z^2 + r^2 = \frac{k}{2}$	Spherical
$k > 0$ and $c > 0$	$\frac{1}{a_e^2} \left(z - \frac{c}{2}\right)^2 + \frac{1}{b_e^2} r^2 = 1$ where: $a_e = \sqrt{\frac{2k+c^2}{4}}$ and $b_e = \sqrt{\frac{k}{2}}$	Ellipse *
$k > 2$ and $c > 0$	$\frac{1}{a_h^2} \left(z - \frac{c}{2}\right)^2 - \frac{1}{b_h^2} r^2 = 1$ where: $a_h = \frac{c}{2} \sqrt{\frac{k-2}{k}}$ and $b_h = \frac{c}{2} \sqrt{\frac{2}{k}}$	Hyperbola *
$k \rightarrow \infty$ and $c \rightarrow \infty$	$z = \frac{h^2 - r^2}{2h}$	Parabola *

Table 2.1: Solutions for the entire class of central projection (single viewpoint) catadioptric cameras with a single reflective surface derived by Nayar and Baker [175]. The mirrors are all swept conic sections. The parameter  $c$  is the separation between the focal (image) plane and the single viewpoint,  $k > 0$  some constant, and  $z$  is the height of the mirror from the viewpoint (focus) — see figure 2.2a. \* denotes the practical solutions identified by Nayar and Baker [175, 7].

For each solution, the mirror must be located at a specific position with respect to the camera. A camera would need to be located in the centre of the spherical mirror for example which is not a practical solution as the mirror would obstruct the camera. The three practical catadioptric cameras identified by Nayar and Baker [175, 7] are those which use an elliptical, hyperbolic or parabolic mirror. Figure 2.6 shows the configuration of a catadioptric camera using each of these mirrors. For the parabolic catadioptric camera, there is an orthographic projection of the rays from the mirror to the image plane (all rays remain parallel to the cameras principal axis). This requires the use of a camera with an orthographic lens system, one example being a telecentric lens [174].

Nayar and Baker note that a major limitation of the elliptical catadioptric camera is the convex mirror which restricts the maximum field of view to a single hemisphere. They also state that although the design of a parabolic catadioptric camera is more difficult than a hyperbolic catadioptric camera, both the construction and calibration of the parabolic catadioptric camera is easier. This is due to the fact that there is an orthographic projection of the rays from the mirror to the image plane. The projection

---

of the mirror shapes, Nayar and Baker [7] show that a solution is possible from 2.7 as  $c \rightarrow \infty$ ,  $k \rightarrow \infty$  and  $\frac{c}{k} = h$  is a constant.

of scene points to the image is therefore invariant to the distance between the reflective surface and the mirror (assuming it is above the mirror). This property makes calibration simpler than with the other classes, particularly when there is a parallel misalignment between the camera's optical axis and the axis of the mirror. Image formation with parabolic catadioptric cameras has also a number of important properties which have permitted simple calibration algorithms to be developed [83], the details of which will be discussed in following sections. A number of different practical designs were developed by Nayar [174].

### The unified image model for central catadioptric cameras

The *unified* image model for central catadioptric cameras derived by Geyer and Daniilidis [85, 55, 84, 86] proves an equivalence between central catadioptric image formation and a two step mapping via the sphere. Importantly, this single model can be used to describe image formation with central projection elliptical, hyperbolic, and parabolic catadioptric cameras. This work originated when proposing a method of calibration for parabolic catadioptric cameras [83], where it was observed that lines in space project to conic sections in the image for central catadioptric cameras. From these observations, the equivalence was proved based on the generalised mapping of points from quadratic surfaces.

The unified model states that for central catadioptric cameras (ellipse, hyperbola, parabola), the process of image formation is equivalent to the projection of a scene point to a sphere centred at the single viewpoint, then from the sphere to the image sensor. A geometric representation of the equivalence is shown in figures 2.7 and 2.8. for a central projection parabolic catadioptric camera. For the parabolic camera, the first stage of the mapping is from a scene point  $\mathbf{X}$  to the parabolic surface whose single effective viewpoint is  $f$ . The second stage of the mapping is orthographic projection from the parabolic surface to a point  $\mathbf{x}$  on the camera sensor  $\ell$ . For a sphere whose centre is at the focus  $f$  of the parabola, and whose radius is equal to the distance of the focus  $f$  to the nearest point on the directrix of the parabola, the first stage of the mapping is from the point  $\mathbf{X}$  to the surface of the sphere followed by stereographic projection to the point  $\mathbf{x}$  on the camera sensor  $\ell$ .

Referring to figure 2.9, image formation using the unified image model is defined by the two parameters  $l$  and  $m$ .  $l$  is the point of projection on the axis orthogonal to the camera sensor  $\ell$  which passes through the focal point (centre of sphere), and  $m$  is the distance of the focal point from the sensor. For a unit sphere, the parameters  $m$  and  $l$  are dependent on the eccentricity  $\varepsilon$  of the mirror (conic section), as summarised in

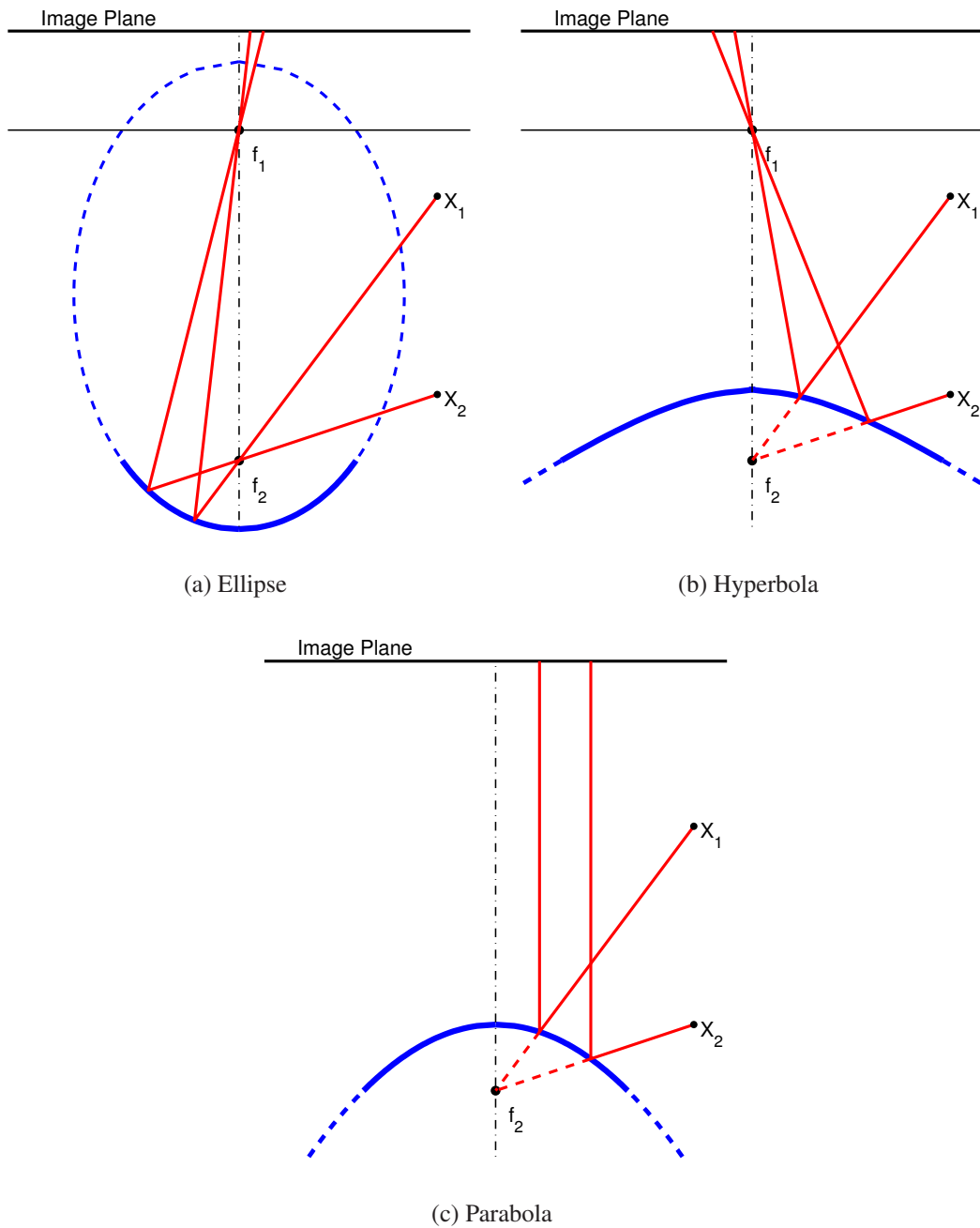


Figure 2.6: The three practical central catadioptric cameras identified by Nayar and Baker [175, 7] capable of obtaining an increase the field of view image use an elliptical, hyperbolic or parabolic reflective surface. All are central projection where incoming rays intersect at a single effective viewpoint  $f_2$ . For the parabolic catadioptric camera, the second stage of the mapping is orthographic projection from the reflective surface to the image plane.

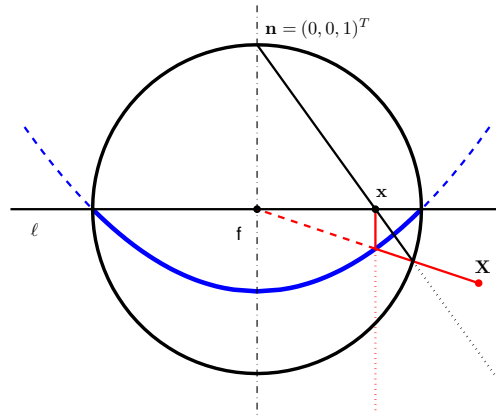


Figure 2.7: Geometric equivalence of image formation using a parabolic catadioptric camera and the unified image model. For the parabolic camera, the first stage of the mapping is from a world point  $\mathbf{X}$  to the parabolic surface whose single effective viewpoint is  $f$ . The second stage of the mapping is orthographic projection from the parabolic surface to a point  $\mathbf{u}$  on the camera sensor  $\ell$ . For the unified image model, the equivalent mapping is from the point  $\mathbf{X}$  to the surface of the sphere centred at  $f$  followed by stereographic projection (from the north pole  $\mathbf{n} = (0, 0, 1)^T$ ) to the point  $\mathbf{u}$  on the camera sensor  $\ell$ .

table 2.2. As illustrated in figure 2.10, for a parabolic camera the point of projection is  $l = 1$ . For both the hyperbola and ellipse, the point of projection is in the range  $0 < l < 1$ . Notice also that the unified image model can model perspective projection for  $l = 0$ .

The unified image model describes the mapping of a point on the sphere  $\eta(\theta, \phi)$  to a point  $\mathbf{x}(r, \zeta)$  defined by polar coordinates on the camera sensor by the equation

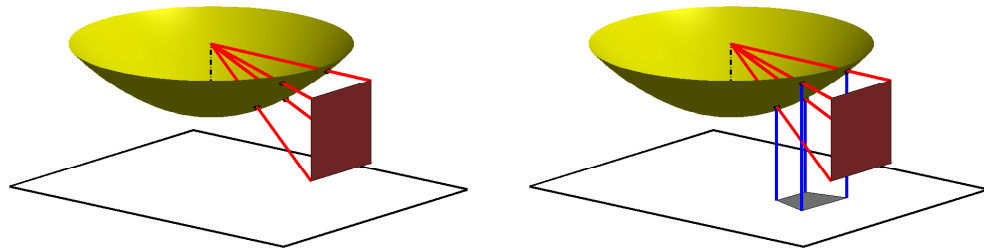
$$r = \frac{(l+m) \sin(\theta)}{l + \cos(\theta)}, \quad \zeta = \phi, \quad (2.9)$$

where the inverse is

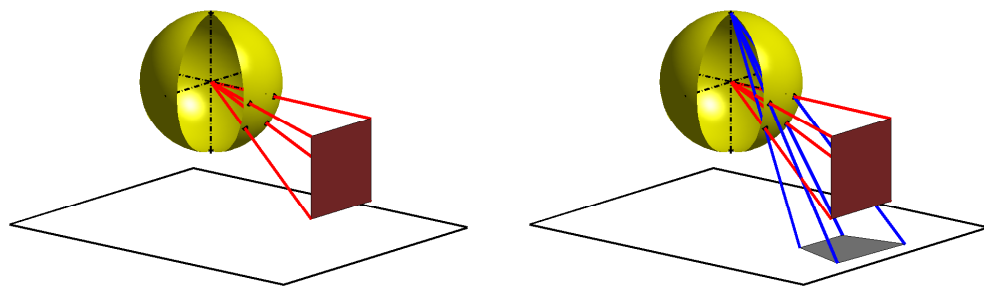
$$\theta = \arccos \left( \frac{(l+m) \sqrt{r^2(1-l^2) + (l+m)^2 - lr^2}}{r^2 + (l+m)^2} \right), \quad \phi = \zeta. \quad (2.10)$$

The transformation from the sensor to image plane coordinates and vice-versa is described by equation 2.5.

As a final note, for the case where the camera model is unknown, Scaramuzza [199, 198] considers a generalised central projection catadioptric camera model (and method for calibration) where the radial distortion is described by a Taylor polynomial.



(a) Image formation with a parabolic catadioptric camera. Two stage mapping is perspective projection to the parabolic mirror followed by orthographic projection to the camera sensor



(b) Image formation under the unified image model. Two stage mapping is perspective projection to the sphere followed by stereographic projection to the camera sensor

Figure 2.8: Equivalent image formation using a parabolic catadioptric camera and the unified image model.

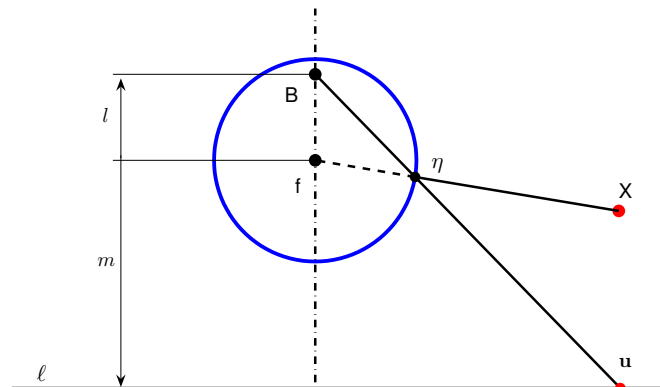


Figure 2.9: Nomenclature for the unified image model. The model is dependent on two parameters; the point of projection  $l$ , and the distance  $m$  from the centre of the sphere (single effective viewpoint) to the camera sensor  $\ell$ .

### Non-central Catadioptric

Although central catadioptric cameras are typically preferred over non-central, as discussed previously, a number of novel non-central designs have been developed by Gaspar et al [81] which may be tailored to specific needs. These include designs capa-

Eccentricity $\varepsilon$	Conic	Variables
$0 < \varepsilon < 1$	ellipse	$l = \frac{2\varepsilon}{1+\varepsilon^2}$ $m = \frac{2\varepsilon(2p-1)}{1+\varepsilon^2}$
$\varepsilon = 1$	parabola	$l = 1$ $m = 2p - 1$
$\varepsilon > 1$	hyperbola	$l = \frac{2\varepsilon}{1+\varepsilon^2}$ $m = \frac{2\varepsilon(2p-1)}{1+\varepsilon^2}$

Table 2.2: Relationship between the unified image model parameters  $m$  and  $l$  and the eccentricity  $\varepsilon$  of the reflective surface (swept conic section).

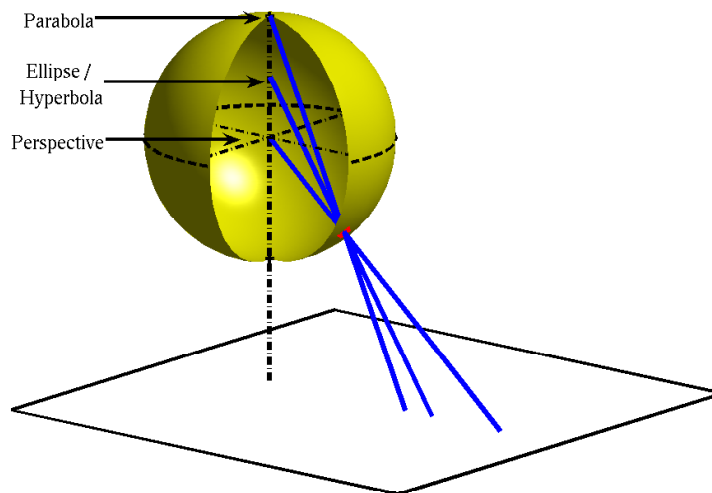


Figure 2.10: Points of projection for central catadioptric cameras under the unified image model.

ble of producing images with constant vertical, horizontal or angular resolution which may be useful for some image processing applications. Chahl and Srinivassan [37] have also proposed a number of catadioptric cameras capable of obtaining omnidirectional images. They derived a number of different mirrors, which coupled with a perspective camera, produce images where the radius  $r$  of a point  $\mathbf{x}(r, \zeta)$  on sensor plane is proportional to the angle of colatitude  $\theta$  of the corresponding ray in space — often referred to as an equiangular camera.

Even when using theoretically central catadioptric cameras (elliptical, hyperbolic and parabolic), slight imperfections in manufacturing and assembly of the camera can



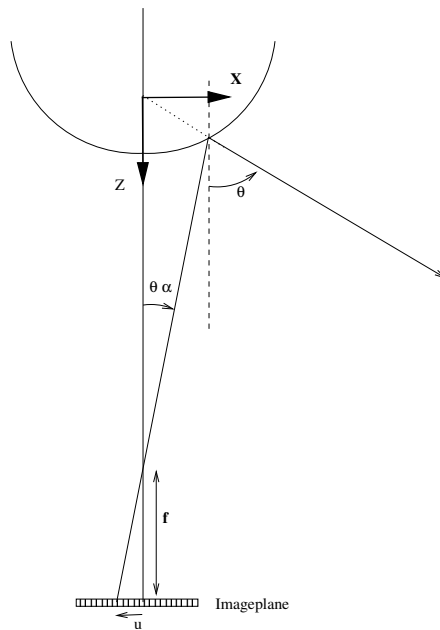


Figure 2.11: Equiangular catadioptric camera model. Although the camera is slightly non-central, accurate structure from motion using this model has been obtained by Corke et al [48] under the assumption of central projection.

result in a non-central camera. A model designed specifically for non-central cameras can therefore be used to obtain more accurate modelling of image formation. Examples include the non-central models of Grossberg and Nayar [93], Swaminathan et al [216], and Mičušík and Pajdla [163] — a comprehensive review of these models is not presented here. Alternately, for practical purposes slightly non-central cameras are often assumed to be central. An example is the use of a non-central equiangular catadioptric camera used for structure from motion of a planetary rover by Corke, Strelow and Singh [48]. Images obtained by this camera are used in later experiments where it is assumed to also be a central projection camera. Referring to figure 2.11, the assumed central projection model is

$$r = f \tan\left(\frac{\theta}{\alpha}\right), \quad \zeta = \phi. \quad (2.11)$$

where  $\mathbf{x}(r, \zeta)$  is the coordinate of a point on the camera sensor.

To recap, for both the central and non-central catadioptric cameras the models can be considered as describing the projection of scene points  $\mathbf{X}$  to points  $\mathbf{x}(r, \zeta)$  on the camera sensor. Equation 2.5 can then be used to find the image plane coordinates  $\mathbf{u}$ .

### 2.2.2.3 Wide-Angle Dioptric (fisheye)

Wide-angle dioptric cameras are typically equipped with a fisheye lens and referred to simply as fisheye cameras. In a sense, a fisheye camera is similar to the human visual system where a lens is used to achieve high spatial resolution at the fovea (centre of image), and low spatial resolution towards the periphery [14], which typically decreases non-linearly [251].

Fleck [72] discusses that the design of fisheye lenses is a compromise between a number of factors including size, cost, geometry and focus of the image, and illumination considerations such as vignetting (intensity drop off towards the periphery). This has resulted in numerous fisheye lens designs by different manufacturers so it can not be assumed that a single standard model is suited for all. Due to the complex design of fisheye lenses it is not possible to derive an exact camera model — most models proposed in the literature are empirical. Therefore, the most suitable model for a particular camera would ideally be selected through experimental comparison. This would require first that for each camera model, the intrinsic parameters can be obtained via calibration. Secondly, some error metric would need to be specified so that the accuracy of the camera models could be compared quantitatively. This metric would be specific to the calibration algorithm used.

Before discussing in detail camera calibration methods, the remainder of this section describes a number of central projection fisheye camera models proposed in the literature. As mentioned earlier, fisheye cameras are frequently assumed to be central projection. With reference to figure 2.12, fisheye camera models can be classified into two classes. The first class are the *pinhole* models which define a direct transform from the fisheye image to an undistorted perspective image. The second class are the *ray-based* models which define the mapping from scene points to the fisheye image. For the ray-based models, each pixel in the image is associated with a ray in space originating from the viewpoint — the scene point  $\mathbf{X}$  imaged at a pixel location  $\mathbf{u}$  is known to lie somewhere on this ray. As each ray intersects the view sphere at some point  $\eta$ , the central projection ray-based models can be used to map a wide-angle fisheye image to the unit view sphere.

#### Pinhole Models

Wide-angle images are sometimes converted to undistorted perspective images for image processing<sup>3</sup>. Pinhole-based models describe a direct transform from distorted

<sup>3</sup>As a perspective image has less than a hemispherical field of view, not all pixels in an omnidirectional wide-angle image can be mapped to a perspective image.

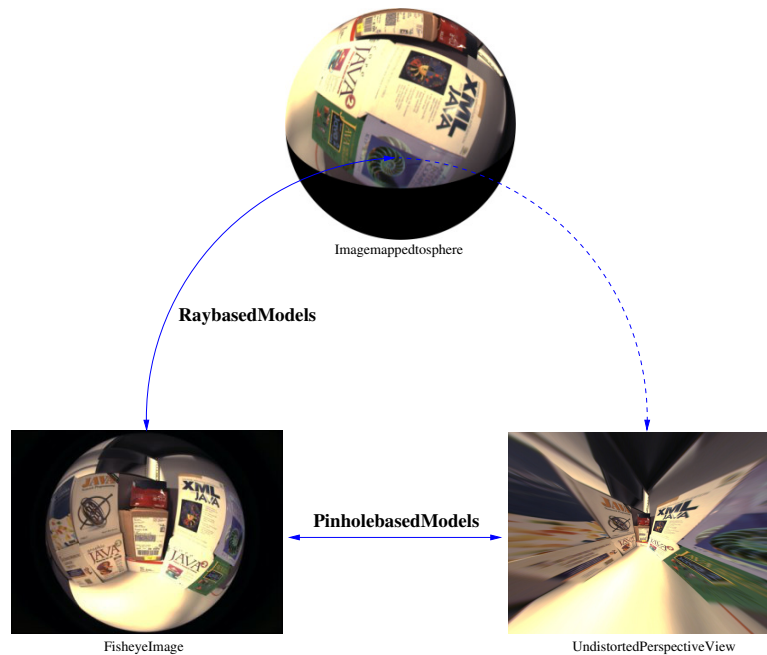


Figure 2.12: Pinhole-based and ray-based camera models. Pinhole models define a direct mapping from a fisheye image to an undistorted perspective image. Ray-based models define a mapping from world points to the fisheye image. This can be simplified as a mapping from the view sphere to the fisheye image and vice-versa. The dashed line indicates that for any ray-based model, the fisheye image can be mapped to a perspective view via the sphere.

wide-angle image coordinates to undistorted perspective image coordinates without requiring any knowledge of the 3D coordinates of the scene points  $\mathbf{X}$ . For this discussion,  $\mathbf{x}(r, \zeta) = (r \cos \zeta, r \sin \zeta)^T = (x, y)^T$  is the coordinate of a point in a fisheye image relative to the principal point  $\mathbf{u}_0$ , and  $\mathbf{x}_p(r_p, \zeta_p) = (r_p \cos \zeta_p, r_p \sin \zeta_p)^T = (x_p, y_p)^T$  is the coordinate of a point in a perspective image relative to the principal point  $\mathbf{u}_{0_p}$ . A summary of the models discussed in this section are given in table 2.3.

Early modelling of variable resolution fisheye images is presented by Schwartz [66] using a complex logarithmic model based on biological vision systems, more specifically the cortical magnification which exists in humans and other primates. However, the fisheye image described by the model is not ideal as it is disjoint along the vertical centre axis, as shown by Basu and Licardie [14]. To overcome this problem, Basu and Licardie [14] proposed two fisheye camera models which they refer to as fisheye transforms (FET's). Both models are representative of real fisheye cameras as they achieve continuity in both axes. The first was the logarithmic model

$$r = s \log(1 + \lambda r_p), \quad \zeta = \zeta_p \quad (2.12)$$

Model	Mapping Function
Log Polar - (Schwartz [66])	$x = \log r_p$ $y = \zeta_p$
Logarithmic - (Basu and Licardie [14])	$r = s \log(1 + \lambda r_p), \quad \zeta = \zeta_p$
Polynomial - (Basu and Licardie [14])	$r = \sum_{i=0}^4 k_i r_p^i, \quad \zeta = \zeta_p$
Field of View (FOV) - (Devernay and Faugeras [61])	$r_p = \frac{\tan(r\omega)}{2 \tan \frac{\omega}{2}}, \quad \zeta_p = \zeta$
Division - (Fitzgibbon [71], Bräuer-Burchardt and Voss [25])	$r_p = \frac{r}{1+kr^2}, \quad \zeta_p = \zeta$
Division - (Thirthala and Pollefeys [221])	$r_p = \frac{r}{(1+k_1r^2+k_2r^4+k_3r^6+\dots)}, \quad \zeta_p = \zeta$
Polynomial - (Shah and Agrawal [206, 207])	$r_p = k_1r + k_2r^2 + k_3r^3 + k_4r^4 + k_5r^5$ $\zeta_p = a_1r + a_2\zeta^2 + a_3\zeta^3 + a_4\zeta^4 + a_5\zeta^5$
Polynomial - (Swaminathan and Nayar [217])	$x_p = x + \cos(\phi)\Delta r(\mathbf{x}) + \Delta T_x(\mathbf{x})$ $y_p = y + \sin(\phi)\Delta r(\mathbf{x}) + \Delta T_y(\mathbf{x})$ $\Delta r(\mathbf{x}) = C_3r^3 + C_5r^5$ $\Delta T_x(\mathbf{x}) = [P_1r^2(1 + 2\cos^2(\zeta)) + 2P_2r^2\sin(\zeta)\cos(\zeta)]$ $\Delta T_y(\mathbf{x}) = [P_2r^2(1 + 2\sin^2(\zeta)) + 2P_1r^2\sin(\zeta)\cos(\zeta)]$

Table 2.3: Summary of pinhole camera models.  $\mathbf{x}(r, \zeta)$  denotes the coordinate of a point on the fisheye image from the principal point, and  $\mathbf{x}_p(r_p, \zeta_p)$  denotes the coordinate of a point on the perspective image relative to the principal point.

which is based on that of Schwartz [66], and the second the quadratic polynomial model

$$r = k_0 + k_1r_p + k_2r_p^2 + k_3r_p^3 + k_4r_p^4, \quad \zeta = \zeta_p. \quad (2.13)$$

Although they observed that the polynomial model gave improved accuracy in their experiments with a real fisheye camera, they suggest that the logarithmic mapping is more ideal. The logarithmic mapping has fewer variables in general  $(s, \lambda)$  which vary the magnitude of distortion and the scale respectively. In contrast, the distortion and scale can not be easily adjusted for the polynomial FET as they are complex functions of the parameters  $k$  which presents challenges using non-linear methods of calibration [14] — without an accurate initial estimate for the coefficients of the polynomial, non-linear methods can potentially converge on solutions for which the calibration ob-

jective function is a local minima. The inverse mapping from a fisheye to a perspective image using the logarithmic FET is also far easier than with high-order polynomials which often require iterative techniques to solve the roots of the polynomial.

A novel model was proposed by Devernay and Faugeras [61] termed the field of view model which describes image formation by the function

$$r_p = \frac{\tan(r\omega)}{2 \tan \frac{\omega}{2}} \quad \zeta_p = \zeta. \quad (2.14)$$

They state that the model is based on the *ideal* fisheye model where the radius  $r$  on the fisheye image plane is roughly proportional to the angle of colatitude  $\theta$  of the corresponding ray in space from the principal axis ( $r \propto \theta$ ). The model is dependent on a single parameter  $\omega$ , which is the field of view of the ideal fisheye lens, and can be varied to model deviations from the ideal model. Although the model is simplistic with only a single variable, they consider that if it is not sufficient to model the distortion then the additional polynomial

$$x' = x(1 + k_2 r^4 + \dots) \quad (2.15)$$

$$y' = y(1 + k_2 r^4 + \dots) \quad (2.16)$$

$$r = \sqrt{x'^2 + y'^2}, \quad \zeta = \arctan(y'/x'), \quad (2.17)$$

can be used before applying equation 2.14.

Another single parameter model, referred to as the division model, was proposed separately by both Fitzgibbon [71] and Bräuer-Burnhardt and Voss<sup>4</sup> [25] of the form

$$r_p = \frac{r}{1 + k r^2}, \quad \zeta_p = \zeta \quad (2.18)$$

where the radial lens distortion is described by a single parameter  $k$ . However, Fitzgibbon notes that the primary advantage of the simplistic single parameter model is the fact that it permits linear calibration using keypoint correspondences between two image. Fitzgibbon states that while this is useful for obtaining a rough calibration, more sophisticated models may be required to more accurately describe the process of image formation with many cameras. The inability for this division model to accurately calibrate many fisheye cameras is due to the fact that it assumes image formation is described by stereographic projection [244]. A point at radius  $r$  on the wide-angle image maps via inverse stereographic projection to a point  $\eta$  on the view sphere with angle

---

<sup>4</sup>The division model presented in the work of Bräuer-Burnhardt and Voss defines the radial distortion in the form  $r_p = r/(1 - k r^2)$ , which is different from equation 2.18 only in the sign of the constant  $k$ .

of colatitude

$$\theta = \arctan\left(\frac{r}{m}\right), \quad (2.19)$$

where  $m$  is the distance of the image plane to the view sphere. The same point with angle of colatitude  $\theta$  on the view sphere would then project to a point on the perspective plane at a distance  $m_p$  from the view sphere at a radius

$$r_p = m_p \tan(\theta) \quad (2.20)$$

$$= \frac{2m_p}{m+1} \left( \frac{r}{1 - \frac{1}{(m+1)^2} r^2} \right) \quad (2.21)$$

from the principal point. This equation is of the same form as equation 2.18, and as a result the division model has since been included in a unified framework for central catadioptric cameras by Barreto [11] and Barreto and Daniilidis [13].

In an attempt to more accurately calibrate fisheye cameras, an extension of the division model was considered by Thirthala and Pollefeys [221] who introduced in the denominator a higher order polynomial in  $r$

$$r_p = \frac{r}{(1 + k_1 r^2 + k_2 r^4 + k_3 r^6 + \dots)}, \quad \zeta_p = \zeta. \quad (2.22)$$

A polynomial model was also first proposed by Shah and Agrawal in [206] and discussed in further detail in [207]. Unlike the model of Basu and Licardie [14] in 2.13, the perspective radius  $r_p$  is a polynomial in  $r$ . Their model includes both radial and decentering distortions and is given as

$$r_p = k_1 r + k_2 r^2 + k_3 r^3 + k_4 r^4 + k_5 r^5 \quad (2.23)$$

$$\zeta_p = a_1 r + a_2 \zeta^2 + a_3 \zeta^3 + a_4 \zeta^4 + a_5 \zeta^5, \quad (2.24)$$

which they used to calibrate a stereo fisheye camera pair used for vision-based localisation of a mobile robot [208]. Swaminathan and Nayar [217] also proposed the following fisheye camera model which accounts for both radial and decentering distortions:

$$x_p = x + \cos(\phi) \Delta r(\mathbf{x}) + \Delta T_x(\mathbf{x}) \quad (2.25)$$

$$y_p = y + \sin(\phi) \Delta r(\mathbf{x}) + \Delta T_y(\mathbf{x}), \quad (2.26)$$

where

$$\Delta r(\mathbf{x}) = C_3 r^3 + C_5 r^5 \quad (2.27)$$

$$\Delta T_x(\mathbf{x}) = [P_1 r^2 (1 + 2 \cos^2(\zeta)) + 2P_2 r^2 \sin(\zeta) \cos(\zeta)] \quad (2.28)$$

$$\Delta T_y(\mathbf{x}) = [P_2 r^2 (1 + 2 \sin^2(\zeta)) + 2P_1 r^2 \sin(\zeta) \cos(\zeta)], \quad (2.29)$$

and  $P$  and  $C$  are the camera model parameters.

To recap, pinhole models define a mapping from a wide-angle image to a perspective image. As a perspective image is restricted to less than a hemispherical field of view, pinhole models are not ideal for use with omnidirectional wide-angle cameras with greater than a hemispherical field of view image. One could always discard the regions on the wide-angle image which do not map to the perspective image, however, as discussed in chapter 1 the extended field of view of wide-angle images is their primary advantage. A more suitable alternative is the class of ray-based models as many of them can be used to model omnidirectional cameras. As a final note, any pinhole model can in effect be converted to a ray based model by substituting  $\phi = \zeta_p$  and  $\theta = \arctan(r_p/m)$ , where  $m$  is a scale factor that would need to be resolved during calibration.

### Ray-based Models

Numerous ray-based models have been proposed and a summary of those discussed in this section is given in table 2.4. Ray-based models are more flexible than pinhole models as many are able to model cameras with an arbitrary field of view. The models in table 2.4 are for central projection cameras where the mapping is defined with respect to the spherical polar coordinates  $\theta, \phi$  of scene points  $\mathbf{X}$ . These can be obtained by first projecting a point  $\mathbf{X}$  to the unit view sphere, centred at the single effective viewpoint, to find the point  $\eta(\theta, \phi) = X/||X||$  and then solving for  $\theta, \phi$  from equation 2.6. The ray-based models typically define for the spherical polar coordinates of a point  $\mathbf{X}$  the corresponding polar coordinates of the point  $\mathbf{x}(r, \zeta)$  on the camera's sensor.

Fleck discusses a number of ray-based models suited to fisheye cameras [72], including stereographic, equidistant, sine law, and equisolid angle models defined in table 2.4. These ray-based models discussed by Fleck are often considered as being *ideal*, where the radial distortion of many fisheye cameras is designed to follow one of these, in particular the equiangular model. Interestingly, Fleck [72] argues that stereographic projection is the most preferred model due to a number of factors. One factor is the ability of stereographic projection to model cameras with field of view in excess

Model	Mapping Function
Stereographic - (Fleck [72])	$r = k \tan \frac{\theta}{2}, \quad \zeta = \phi$
Equiangular (equidistant) - (Fleck [72])	$r = k\theta, \quad \zeta = \phi$
Sine Law - (Fleck [72])	$r = k \sin \theta, \quad \zeta = \phi$
Equisolid angle - (Fleck [72])	$r = k \sin \frac{\theta}{2}, \quad \zeta = \phi$
Combination - (Bakstein and Pajdla [9])	$r = a \tan \left( \frac{\theta}{b} \right) + c \sin \left( \frac{\theta}{d} \right), \quad \zeta = \phi$
Mičušík 1 - (Mičušík and Pajdla [164, 162])	$r = \frac{a - \sqrt{a^2 - 4b\theta^2}}{2b\theta}, \quad \zeta = \phi$
Mičušík 2 - (Mičušík [162])	$r = \frac{a}{b} \sin(b\theta), \quad \zeta = \phi$
Polynomial - (Xiong and Turkowski [243])	$r = k_1\theta + k_2\theta^2 + k_3\theta^3, \quad \zeta = \phi$
Polynomial - (Ho [101])	$r = k_1\theta + k_2\theta^3 + k_3\theta^5, \quad \zeta = \phi$
Unified - (Geyer and Daniilidis)	$r = \frac{(l+m)\sin(\theta)}{l+\cos(\theta)}, \quad \zeta = \phi$
Rational - (Claus and Fitzgibbon [45])	$\mathbf{d} = A\chi(u', v'), \quad \mathbf{u}'(u', v') = \mathbf{u} - \mathbf{u}_0$ $A_{3 \times 6}, \quad \chi(u', v') = (u'^2, u'v', v'^2, u', v', 1)^T$ $\theta = \arccos(\mathbf{d}_z / \ \mathbf{d}\ ), \quad \phi = \arctan(\mathbf{d}_y, \mathbf{d}_x)$
Kannala and Brandt [118, 119]	$\mathbf{x}(x, y) = (r + \Delta_r(\theta, \phi))\mathbf{u}_r(\phi) + \Delta_t(\theta, \phi)\mathbf{u}_\phi(\phi),$ where $r = k_1\theta + k_2\theta^3 + k_3\theta^5 + k_4\theta^7 + k_5\theta^9$ $\Delta_r(\theta, \phi) = (l_1\theta + l_2\theta^3 + l_3\theta^5)$ $(i_1 \cos \phi + i_2 \sin \phi + i_3 \cos 2\phi + i_4 \sin 2\phi)$ $\Delta_t(\theta, \phi) = (m_1\theta + m_2\theta^3 + m_3\theta^5)$ $(j_1 \cos \phi + j_2 \sin \phi + j_3 \cos 2\phi + j_4 \sin 2\phi)$

Table 2.4: Summary of ray-based camera models. A point on the camera's sensor has a coordinate  $\mathbf{x}(r, \zeta) = (r \cos \zeta, r \sin \zeta)^T = (x, y)^T$ . A ray in space from a scene point  $\mathbf{X}$  is parameterised by spherical coordinates  $\theta, \phi$ , and intersects the unit sphere, centred at the viewpoint, at the point  $\eta(\theta, \phi) = \mathbf{X} / \|\mathbf{X}\|$ .

of a hemisphere (omnidirectional), which applies also to the equiangular and equisolid angle models. Another factor is that stereographic projection is a conformal mapping — a more detailed discussion relating to the conformal nature of stereographic projection will be presented in chapter 4. This means that straight lines in space project to circles on the image plane, a property which has permitted simple calibration algo-



rithms to be developed for these cameras [83]. It also means that the image produced by a camera whose process of image formation is described by stereographic projection is locally perspective. This property is of significance when considering image processing and is the inspiration for one of the methods of keypoint detection with wide-angle images proposed in chapter 4. In reality many of the ideal models are not capable of describing image formation with real fisheye cameras with a high degree of accuracy. Many alternates have been proposed as a result.

As was the case for some of the pinhole models, polynomials have also been used for ray-based models. An example is the cubic polynomial model of Xiong and Turkowski [243]:

$$r = k_1\theta + k_2\theta^2 + k_3\theta^3, \quad \zeta = \phi. \quad (2.30)$$

They state that the equidistant model  $r = k\theta$  is a good approximation for many cameras, however, the higher order polynomial terms are able to more accurately model radial distortion towards the periphery of a wide-angle camera's field of view. In the same work, they validated their model with a real fisheye camera where they were able to seamlessly register multiple images obtained by a rotating camera.

A novel hybrid model was presented by Bakstein and Pajdla [9] which is a combination of stereographic and equisolid angle projections and has the form

$$r = a \tan \frac{\theta}{b} + c \sin \frac{\theta}{d}, \quad \zeta = \phi. \quad (2.31)$$

They initially considered the the model  $r = a \tan \frac{\theta}{b}$ , but found improved accuracy extending it to that given in equation 2.31. A limitation of the model is that it is not algebraically invertible (as is the case with some higher order polynomials).

In the work of Mičušík and Pajdla [164] and Mičušík [162], two models for fisheye cameras were presented which are given in equations 2.32 and 2.33 respectively:

$$r = \frac{a - \sqrt{a^2 - 4b\theta^2}}{2b\theta}, \quad \zeta = \phi, \quad (2.32)$$

$$r = \frac{a}{b} \sin(b\theta), \quad \zeta = \phi. \quad (2.33)$$

In both cases, the variable  $b$  models distortions from the ideal equiangular projection (as  $b$  approaches zero, each model converges to ideal equiangular projection). They also model and calibrate for the affine transform in the sensor to image coordinate conversion given in equation 2.5 which accounts for uneven scaling in  $x, y$  sensor coordinates and shear.

Ying and Hu [244] suggested that the unified image model for central catadioptric cameras derived by Geyer and Daniilidis [86] (equation 2.9) could be extended to include many fisheye cameras. Referring to figure 2.9, they proposed that many fisheye cameras could be modelled for a point of projection  $l \geq 1$ . They showed using this model that line images, which are the projection of straight lines in space onto the camera sensor, are constrained to be either circles for  $l = 1$  (stereographic projection) or ellipses for  $l > 1$ . As noted in their work, the use of a single model for both central catadioptric and fisheye cameras has advantages during calibration where techniques developed for central catadioptric cameras can be applied [245].

Ho [101] argues that there is no evidence to support the claim that image formation with fisheye cameras is described by the extension of the unified model proposed by Ying and Hu [244] as the complex design of fisheye lenses makes it difficult to derive an exact camera model. They suggest therefore that modelling fisheye cameras using the extended unified model can be of limited accuracy. For this reason, similarly to Xiong and Turkowski [243] they proposed a generalised polynomial of the form

$$r = k_1\theta + k_2\theta^3 + k_3\theta^5, \zeta = \phi. \quad (2.34)$$

They compared this model to two other polynomials (including the polynomial of Xiong and Turkowski [243] in 2.36)

$$r = k_1 + k_2\theta + k_3\theta^2 + k_4\theta^3 \quad (2.35)$$

and

$$r = k_1\theta + k_2\theta^2 + k_3\theta^3 \quad (2.36)$$

for calibration of eight different fisheye cameras. They found that their polynomial in 2.34 was able to achieve the smallest residual reprojection errors of grid points for each camera. It is of interest to note here that the modelling of image formation with a fisheye camera is an inherently empirical process, and there is no evidence that polynomial functions are ideal. For many calibration algorithms it could be possible to simply select a high order polynomial which could in theory obtain ‘perfect’ calibration for the cost function used.

A novel model was proposed by Claus and Fitzgibbon [45] which they refer to as the rational model. They define for each pixel  $\mathbf{u}'(u', v') = \mathbf{u} - \mathbf{u}_0$  in the fisheye image the corresponding ray in space denoted as  $\mathbf{d}(u', v')$  by

$$\mathbf{d}(u', v') = A_{3 \times 6} \chi(u', v'). \quad (2.37)$$

$\chi(u', v')$  is a monomial in  $u'$  and  $v'$ ,

$$\chi(u', v') = \left[ u'^2 \quad u'v' \quad v'^2 \quad u' \quad v' \quad 1 \right]^T, \quad (2.38)$$

which they refer to as the *lifted* image points, and  $A_{3 \times 6}$  contains the 18 camera model parameters. The angle of colatitude  $\theta$  and longitude  $\phi$  of a ray are related to  $\mathbf{d}$  by

$$\theta = \arccos \left( \frac{\mathbf{d}_z}{\|\mathbf{d}\|} \right), \quad (2.39)$$

$$\phi = \arctan \left( \frac{\mathbf{d}_y}{\mathbf{d}_x} \right). \quad (2.40)$$

A number of ray-based models include decentering distortions. An example is the model proposed by Kanalla and Brandt [118]. The radial distortion is modelled using the polynomial

$$r = k_1\theta + k_2\theta^3, \quad (2.41)$$

where higher order powers are not included as the polynomial would not be analytically invertible. However, in later work [119] they extend the polynomial

$$r(\theta) = k_1\theta + k_2\theta^3 + k_3\theta^5 + k_4\theta^7 + k_5\theta^9, \quad (2.42)$$

which they claim is able to model stereographic, equidistant, equisolid angle, and orthogonal projection with a ‘moderate’ level of accuracy. Unlike radial distortion, decentering distortion is not symmetric and is a function of both  $\theta$  and  $\phi$ . They model decentering distortion with two additional distortion terms; one acting radially  $\Delta_r(\theta, \phi)$  and the other tangentially  $\Delta_t(\theta, \phi)$  which are defined as

$$\Delta_r(\theta, \phi) = (l_1\theta + l_2\theta^3 + l_3\theta^5)(i_1 \cos \phi + i_2 \sin \phi + i_3 \cos 2\phi + i_4 \sin 2\phi) \quad (2.43)$$

and

$$\Delta_t(\theta, \phi) = (m_1\theta + m_2\theta^3 + m_3\theta^5)(j_1 \cos \phi + j_2 \sin \phi + j_3 \cos 2\phi + j_4 \sin 2\phi), \quad (2.44)$$

where  $l, i, m$  and  $j$  are the camera model parameters. The relationship between a ray in space and the corresponding point  $\mathbf{x}$  on the camera’s sensor is

$$\mathbf{x} = (r(\theta) + \Delta_r(\theta, \phi))\mathbf{u}_r(\phi) + \Delta_t(\theta, \phi)\mathbf{u}_\phi(\phi), \quad (2.45)$$

where  $\mathbf{u}_r(\phi)$  and  $\mathbf{u}_\phi(\phi)$  are unit vectors in the radial and tangential directions respec-

tively. Referring to equation 2.5, they account for uneven scaling in the sensor to image coordinate transform using a matrix  $A$  of the form

$$A = \begin{bmatrix} m_u & 0 \\ 0 & m_v \end{bmatrix}. \quad (2.46)$$

To summarise, the pinhole and ray-based camera models discussed are for central projection cameras. In reality most fisheye cameras are not central projection as a result of the complex nature of the lens design. A number of non-central camera models have been proposed as a result. Examples include the model of Grossman and Nayar [93], and the CAHVORE model of Gennery [82] which has been used to model image formation for a range of cameras (narrow and wide-angle field of view) on the Mars exploration rovers. However, in most circumstances fisheye cameras are assumed to be central projection for practical purposes, and this assumption is made for the the fisheye camera used throughout the thesis. Although only some of the fisheye camera models discussed account for uneven scaling and shear by the affine matrix  $A$  in equation 2.5, it can be included with all the models discussed to further improve accuracy if required.

## 2.3 Review of Camera Calibration

Camera calibration is the process of estimating the parameters of the camera model which describe image formation. This includes the intrinsic camera model parameters, and for some calibration algorithms, the extrinsic parameters which define the pose of the camera with respect to a world coordinate frame of reference (the pose with respect to a calibration target for example). There is extensive literature relating specifically to camera calibration, and a comprehensive review of notable historical developments is presented by Clarke and Fryer in [44]. This section provides a review of some fundamental approaches to calibration which focuses primarily on methods used with wide-angle cameras.

Methods of camera calibration can be broadly categorised into the following groups:

1. Full-range: resolve camera intrinsic and extrinsic parameters using known relative Euclidean coordinates of calibration control points. These control points are typically points on a calibration target such as a checkerboard pattern.
2. Auto-calibration: uses keypoint correspondences between multiple views to cal-

ibrate camera parameters.

3. Plumb-line: calibrate camera using the fundamental projective invariant that straight lines in space project to straight lines in the perspective plane. This method can be applied to any central projection pinhole or ray-based camera model.

### 2.3.1 Full-range

Full-range calibration algorithms use the precise relative 3D Euclidean coordinates of calibration control points, for example the corners a checkerboard pattern. For an estimate of both the camera intrinsic and extrinsic parameters, these control points are projected to pixel locations in the image. The reprojection errors are the Euclidean distances, measured in the image with units of pixels, between these points and the locations of the control points detected in the image. The sum of the reprojection errors can then be used as a quantitative objective function to be minimised during calibration. Importantly, full-range calibration algorithms can operate using only a single input image.

A full-range calibration algorithm was introduced in the seminal work of Tsai [227]. Two versions of the algorithm exist for the case where either coplanar or non-planar control points are used whose relative Euclidean coordinates are known precisely. The algorithm operates in two steps. The first step obtains an initial linear estimate of the camera extrinsic parameters, and the second step a non-linear estimate of both the extrinsic and intrinsic parameters (including a radial distortion coefficient) — the initial estimates of the extrinsic parameters are used in this stage. The second stage minimises the reprojection errors of the control points measured in the image.

Bakstein and Pajdla [8] also introduced a full-range calibration algorithm which they used for calibration of a fisheye camera using their model in table 2.4. Unlike Tsai, their camera had a near hemispherical field of view. They used for control points a calibration target inside a circular tube, where the reprojection error of these points in the image plane was used to calibrate the camera parameters using a non-linear iterative refinement. An interesting part of their algorithm was the technique used to estimate the affine matrix  $A$  and position of the principal point  $\mathbf{u}_0$  which defined the sensor to image plane transform in equation 2.5. Rather than estimate these parameters during iterative refinement, they were found by fitting an ellipse to the boundary of the camera's field of view in the image. This boundary encloses the area in an image in which regions of the scene can be imaged (see figure 2.13, pg. 63). However, as noted

by Li and Hartley [128], this approach is most suited when the full boundary of the camera's field of view is visible in the image. Kang [117] also notes that the methods used to fit contours to these boundaries, for example using binary thresholding, are not always robust as a single threshold cannot account for the large illumination variations across the wide field of view of the camera.

Mei and Rives [153] also considered a full-range calibration algorithm which they implemented with a variety of central projection catadioptric and (assumed) central projection fisheye cameras. The camera models used included both radial and tangential distortions. They used as control points the corners of a planar checkerboard pattern and calibrate the camera's intrinsic and extrinsic parameters by minimising the reprojection errors using iterative refinement. The boundary of the camera's field of view was used to obtain an initial estimate of the position of the principal point  $\mathbf{u}_0$  whose accuracy was improved during the iterative refinement. Full-range calibration has also been performed by Kannala and Brandt [118] using their model in table 2.4 for calibration of a fisheye camera with an approximately hemispherical field of view. They use circular control points on a planar calibration target and calibrate for the camera parameters by minimising the sum of squared distances of reprojection errors in the image plane. To prevent convergence on local minima, they fit an ellipse to boundary of the camera's field of view and use manufacturer data to obtain an initial estimates of the camera's intrinsic parameters.

Although most full-range calibration algorithms are applied to central projection cameras, full-range calibration has also been used successfully for non-central camera models with the CAHVORE model by Gennery [82]. As is the case with most full-range methods, the camera parameters are estimated using non-linear iterative refinement.

### 2.3.2 Auto-calibration

Auto-calibration is also referred to as self-calibration and is typically characterised by the ability to calibrate for the camera intrinsic parameters using keypoint correspondences between two or more views. Numerous algorithms have been developed using different constraints on relative position of the scene points associated with the correspondences (arbitrary position or coplanar) and/or the change in pose between viewpoints. In all cases, the scene points are assumed to remain rigid between views. The advantage of some auto-calibration algorithms is their ability to obtain a linear solution for the camera model parameters.

A number of algorithms have been developed that can calibrate a camera using images separated by a change in camera rotation. One example is the algorithm of Hartley [97] that can obtain a linear estimate of a perspective camera's intrinsic parameters using corresponding keypoints in a minimum of three views, where the scene points associated with these corresponding keypoints can be at any arbitrary position in space. Zhang [250] has also proposed a method of calibration using keypoint correspondences between images of the same scene separated by a change in camera rotation that is suited for cameras with radial distortion. Although the method requires a minimum of only two views (tested with up to 16 in experiments), unlike the algorithm of Hartley [97], the scene points associated with the keypoint correspondences must be coplanar, and a solution for the camera model parameters is obtained using a non-linear iterative refinement. Xiong and Turkowski [243] also proposed a novel auto-calibration algorithm which they used to calibrate a fisheye camera with an approximately hemispherical field of view modelled with the cubic polynomial in table 2.4. Given four images taken by a camera at different orientations (single-axis rotation), they use iterative techniques to estimate the camera model parameters by minimising registration errors (difference in pixel intensity values) in overlapping regions of the images. Although the position of the camera's principal point is found during calibration, they use a similar approach to Mei and Rives [153] and fit a circle to the boundary of the camera's field of view to obtain an initial estimate for the position of the principal point before calibration.

A number of potentially more versatile methods have been developed using correspondences between views which differ in pose by a rotation and translation. These again include cases using coplanar scene points (planar scenes), and for scene points at arbitrary location. An example of the former is the algorithm of Triggs [226] which requires a minimum of 5 views and solves for the camera intrinsic parameters and the extrinsic parameters which relates any two views by a planar homography<sup>5</sup>. For the latter, epipolar constraints have been used frequently for auto-calibration from two views<sup>6</sup>(although Thirtahlla and Pollefeys have developed a method for three views using the trifocal tensor [221]). An early example is given by Zhang [249] for calibration of a narrow field of view camera with both radial and tangential distortions modelled using polynomial functions. Using epipolar constraints, a keypoint in one view maps to a curve in the second view, where the error in this mapping (epipolar error) measured in the image plane was used as the objective function for non-linear calibration. For cameras with radial distortion, this epipolar error is a point to curve distance. Kang

---

<sup>5</sup>The reader is referred to chapter 5 for a detailed discussion about planar homographies.

<sup>6</sup>The reader is referred to chapter 5 for a detailed discussion on epipolar geometry.



[117] also proposed a method using epipolar constraints suited specifically to wide-angle catadioptric cameras. Again, the epipolar error measured in the image plane was used for calibration which was implemented using non-linear iterative techniques.

A notable work relating to auto-calibration using epipolar constraints is that of Fitzgibbon [71]. Although the algorithm developed by Fitzgibbon was specific to the single parameter division model proposed by the same author (refer to table 2.4), the camera's radial distortion coefficient was essentially built into the equations (fundamental matrix) defining the epipolar constraints. The equations were reformulated as a quadratic eigenvalue problem from which a linear solution could be obtained. Mičušík and Pajdla [164] generalised the approach of Fitzgibbon to obtain a linear solution for the two parameter model proposed by the same authors (refer again to table 2.4) which they used to calibrate a fisheye camera with excess of a hemispherical field of view. As only the radial distortion was incorporated into the equations which define the epipolar constraints, they fitted an ellipse to the boundary of the camera's view field to estimate the affine transform and position of the principal point in equation 2.5 which defines the sensor to image coordinate transform — this is used to correct the position of the correspondences before auto-calibration. However, as noted by Li and Hartley [128], Mičušík and Pajdla [164] use knowledge of the cameras known field of view, which combined with the fitted ellipse, reduces the radial distortion model to a single parameter model. Li and Hartley also present an auto-calibration algorithm using epipolar constraints in [128] using methods developed in earlier works by Hartley and Kang [98]. They use two-view invariants derived from the epipolar geometry between views, and in contrast to the methods of both Fitzgibbon and Mičušík and Pajdla, does not require them to solve explicitly for the fundamental matrix between views. An advantage of this is the the removal of potentially difficult interactions that can occur when simultaneously estimating both camera intrinsic and extrinsic parameters which leads to inaccurate estimates [236]. Furthermore, Li and Hartley's algorithm [128] can obtain a linear estimate of a camera's intrinsic parameters for models with more than one parameter which they validated experimentally with synthetic and real wide-angle cameras. Further improvements of these methods is presented again by Li and Hartley in [130], where methods specific for correspondences associated with planar and non-planar scene points are given.

### 2.3.3 Plumb-line

Plumb-line calibration algorithms exploit the projective invariant that straight lines in space project to straight lines in the perspective plane. The term 'plumb-line' originates



from early works relating to camera calibration of Brown [27], where plumb blobs attached to strings were used as reference straight lines to calibrate a narrow field of view camera modelled as having both radial and tangential distortion.

Recall that the pinhole based models in table 2.3 describe a mapping a distorted wide-angle image to the perspective image plane. Plumb-line calibration algorithms can be used to find the camera model parameters for which the images of straight lines in space in the distorted image (which appear curved) map as closely as possible to straight lines in a perspective image. A general method used for this purpose by both Devernay and Faugeras [61] and Swaminathan and Nayar [217] is to first detect edges in the image associated with straight lines in space. These edges are then mapped to a reconstructed perspective image for a given estimate of the camera intrinsic parameters. The error between these edges in the perspective image and fitted straight lines is used to define the error to be minimised during calibration using iterative techniques. For some camera models, the appearance of straight lines in space in the distorted image can be derived from the camera model. Using the division model of Bräuer-Burchardt and Voss [25] for example, straight lines in space are known to project to circles in the wide-angle image [11, 13] (the division model assumes image formation is described by stereographic projection). The calibration algorithm of Bräuer-Burchardt and Voss [25] therefore calibrates for the camera model parameters using iterative techniques by fitting directly circles to the detected lines in the images associated with straight lines in space.

The principles of plumb-line methods have also been used for central projection catadioptric camera calibration. They are formulated on the constraint that a straight line in space projects to a great circle on the unit view sphere centred at the viewpoint, which in turn projects to conic section on camera's sensor [86, 12] — this conic section is a circle for a parabolic catadioptric camera. If there is equal scaling and zero shear in the sensor to image coordinate transform, a straight line in space will project to a conic section in the image. Geyer and Daniilidis [83] showed that the intrinsic parameters of a parabolic catadioptric camera could be calibrated using only two pairs of parallel lines. In their experiments circles were fitted to points in the image known to be collinear in space. For each pair of parallel lines, the fitted circles intersect at two points on the image which project by inverse stereographic projection to antipodal points on the view sphere. These antipodal points are the vanishing points of the parallel lines. The vanishing points found for the two pairs of parallel lines was sufficient to estimate the camera intrinsic parameters. Interestingly, this method shares similarities with the perspective camera calibration algorithm described by Hartley and Zisserman [96] in which the vanishing points of the edges of a cube detected in a perspective

image are sufficient to calibrate the camera intrinsic parameters. Geyer and Daniilidis improved on their method and showed that calibration of a parabolic catadioptric camera could be achieved using the image of as few as three lines [87] assuming that the affine matrix  $A$  in the image to sensor coordinate transform is known (e.g.  $A = I_{2 \times 2}$ , see equation 2.5). Barreto and Araujo [12] later proved that hyperbolic and elliptical catadioptric cameras could be calibrated using the image of only two lines which project to conics on the camera's assuming again that the affine matrix  $A$  in the image to sensor coordinate transform is known.

The calibration algorithm of Geyer and Daniilidis [83] uses the intersection of parallel lines (vanishing points) for calibration. Becker and Bove [20] have also formulated a calibration algorithm, suited for use with central projection cameras, which uses the intersection of parallel lines. They detect in an image three mutually orthogonal sets of parallel lines in space. These are then projected to the unit sphere for a given estimate of the camera intrinsic parameters at which point great circles are fitted to them — a straight line in space projects to a great circle on the unit view sphere centred at the viewpoint. The calibration algorithm operates on the constraint that a set of parallel lines in space project to a set of great circles on the sphere that all intersect at two antipodal points. For an incorrect estimate of the camera intrinsic parameters, the set of fitted great circles corresponding to a set of parallel lines in space detected in the image will not intersect exactly at two antipodal points. Each pairwise combination of great circles in the set will intersect at two unique antipodal points resulting in a 'dispersion' of antipodal points (vanishing points) for the set. The calibration algorithm uses a non-linear iteration to find the camera intrinsic parameters which minimises the dispersion of the vanishing points for all sets of parallel lines.

### 2.3.4 Discussion

As just discussed there are a range of different calibration algorithms suitable for use with central projection wide-angle cameras. If one wanted to compare empirically the accuracy of a number of different camera models used to describe image formation for a given camera, the same calibration algorithm would need to be used to find the camera intrinsic parameters for each model. The objective function minimised during calibration can then be used to make this quantitative comparison. A number of different ray-based camera models will be compared for the fisheye camera used throughout the thesis. This is necessary as, unlike central catadioptric cameras, the model describing image formation cannot be easily derived geometrically. The following discussion considers which calibration algorithms would be most suited for comparing different

camera models.

The auto-calibration algorithms of Fitzgibbon [71], Mičušík and Pajdla [164], and Li and Hartley [128] have been used to calibrate fisheye cameras. They are able to obtain a linear estimate of the camera intrinsic parameters which avoids many limitations of iterative techniques that include [128]: lack of convergence, convergence on local minima, requirement for selecting an accurate initial estimate, requirement for selecting a stop criteria and computational expense. Kang [117] also suggests that auto-calibration algorithms using epipolar constraints are well suited for calibration of wide-angle cameras with a large field of view as they are able to find more accurately estimate camera egomotion than narrow field of view cameras. However, the algorithms of Fitzgibbon [71] and Mičušík and Pajdla [164] are suited only for specific camera models. The limitations of auto-calibration algorithms using epipolar constraints are also noted by Fitzgibbon [71]. Fitzgibbon states that although the methods are suitable for some computer vision applications, full-range calibration algorithms using known control points and bundle adjustment (iterative refinement) are preferred for accurate calibration. As previously mentioned, auto-calibration algorithms using epipolar constraints are also subject to the potentially harmful interactions which exists in the simultaneous estimation of both intrinsic and extrinsic camera parameters [236]. The algorithm of Li and Hartley [128] avoids this problem, and although it can include more radial distortion parameters, it does not easily generalise to any arbitrary model.

Full-range calibration algorithms have been used extensively for wide-angle camera calibration and require that the precise relative 3D Euclidean coordinates of the control points be found. Most use non-linear iterative techniques which require that good initial estimates of the camera extrinsic and extrinsic parameters are found prior to calibration to avoid convergence on local minima of the calibration objective function. With the exclusion of the algorithms of Geyer and Daniilidis [83, 87] and Barreto and Araujo [12], many plumb-line methods also use non-linear iterative techniques and can be used with any camera model. Unlike the full-range algorithms, plumb-line algorithms do not require the precise relative Euclidean coordinates of control points to be found. The calibration algorithm developed in the next section is based on the plumb-line calibration algorithms discussed and is used to select the most suitable ray-based camera model for the fisheye camera used throughout the thesis.

## 2.4 Camera Calibration Algorithm and Results

A novel calibration algorithm is developed in this section. It is based on the plumb-line calibration algorithms discussed previously, and it is suited for central projection catadioptric and central projection fisheye (with ray-based models) camera calibration. The algorithm uses similar constraints on intersections of parallel lines in space used by Geyer and Daniilidis [83] and Becker and Bove [20], and fundamentals of projective geometry on the sphere discussed in [86, 12]. The algorithm uses the constraint that a set of parallel lines in space project to great circles on the sphere which all intersect at two antipodal points. The algorithm is novel in the sense that this constraint is enforced strictly during calibration. Becker and Bove [20] used a similar constraint for calibration, but as discussed in the previous section, they minimise the dispersion of the antipodal points during calibration. Furthermore, the algorithm enforces the constraint that if two sets of coplanar parallel lines in space are orthogonal to each other, and each of these sets project to great circles on the sphere which intersect at two antipodal points, then the four antipodal points of intersection lie on another great circle which is the fronto-parallel horizon of the plane in space containing the lines [83]. These constraints will be illustrated more clearly in the following sections.

The calibration algorithm is used to calibrate the fisheye camera used throughout this work using a selection of the ray-based camera models in table 2.4, where the accuracy of each model is defined by a quantitative calibration objective function to be minimised. These results are used to select the model (and camera intrinsic parameters) most suited for the fisheye camera which is used for the remainder of the thesis. The calibration algorithm proposed operates offline using multiple images of a planar checkerboard calibration target. A sample image obtained by the fisheye camera used throughout the thesis is shown in figure 2.13 ( $1024 \times 768$  pixels).

### 2.4.1 Preliminaries

Referring to figure 2.14, define the set of *grid points* as the corners of the checkerboard squares. For a rectangular checkerboard target with  $n_i$  rows and  $n_j$  columns of grid points, let  $\mathbf{X}_{i,j} = (X, Y, Z)^T$  be the Euclidean world coordinate of a grid point. Define a line  $\mathcal{L}_i$  as the subset of all points  $\mathbf{X}_{i,j \in \{1,2,\dots,n_j\}}$ , where all lines  $\mathcal{L}_{i \in \{1,2,\dots,n_i\}}$  form the set of parallel lines  $\mathcal{L}$ . Define a line  $\mathcal{L}'_j$  as the subset of all points  $\mathbf{X}_{i \in \{1,2,\dots,n_i\},j}$ , where all lines  $\mathcal{L}'_{j \in \{1,2,\dots,n_j\}}$  form the set of parallel lines  $\mathcal{L}'$ . If a unit sphere is centred at the single effective viewpoint of a camera, and this viewpoint is at the origin of the world

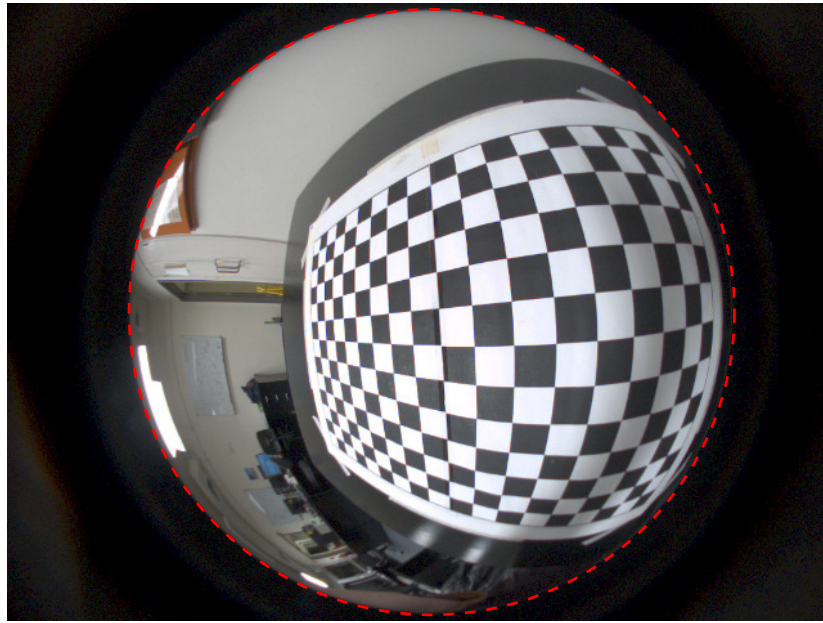


Figure 2.13: Example image ( $1024 \times 768$  pixels) of the planar checkerboard calibration target obtained with the fisheye camera used throughout this work. The camera is a Point Grey Research Dragonfly firewire camera fitted with an OmniTech Robotics fisheye lens with an approximate field of view of  $\theta = \pm 190^\circ/2$ . The red dashed line indicates the boundary of the camera's field of view.

coordinate frame of reference containing the grid points  $\mathbf{X}$ , then a ray from each point  $\mathbf{X}_{i,j}$  passing through the viewpoint intersects the sphere at a point  $\eta_{i,j} = \mathbf{X}_{i,j}/\|\mathbf{X}_{i,j}\|$ . With reference to figure 2.15, the following constraints can be made regarding the set of grid points  $\eta$  on the sphere which are used for both calibration and grid point detection:

1. Each line  $\mathcal{L}_i$  projects to a great circle  $G_i$  on the unit sphere. As points  $\mathbf{X}_{i,j \in \{1,2,\dots,n_j\}}$  are constrained to lie on the line  $\mathcal{L}_i$ , points  $\eta_{i,j \in \{1,2,\dots,n_j\}}$  are constrained to lie on the great circle  $G_i$ . Similarly, each line  $\mathcal{L}'_j$  projects to a great circle  $G'_j$  on the unit sphere. As points  $\mathbf{X}_{i \in \{1,2,\dots,n_i\},j}$  are constrained to lie on the line  $\mathcal{L}'_j$ , points  $\eta_{i \in \{1,2,\dots,n_i\},j}$  are constrained to lie on the great circle  $G'_j$ .
2. Since the set of lines  $\mathcal{L}$  are parallel, the set of great circles  $G$  will intersect at antipodal points  $\pm\eta_G$  on the unit sphere. Since the set of lines  $\mathcal{L}'$  are parallel, the set of great circles  $G'$  will intersect at antipodal points  $\pm\eta_{G'}$  on the unit sphere.
3. As all points  $\mathbf{X}$  are coplanar in space, the sets of parallel lines  $\mathcal{L}$  and  $\mathcal{L}'$  are coplanar. The antipodal points of intersection  $\pm\eta_G$  and  $\pm\eta_{G'}$  are therefore constrained

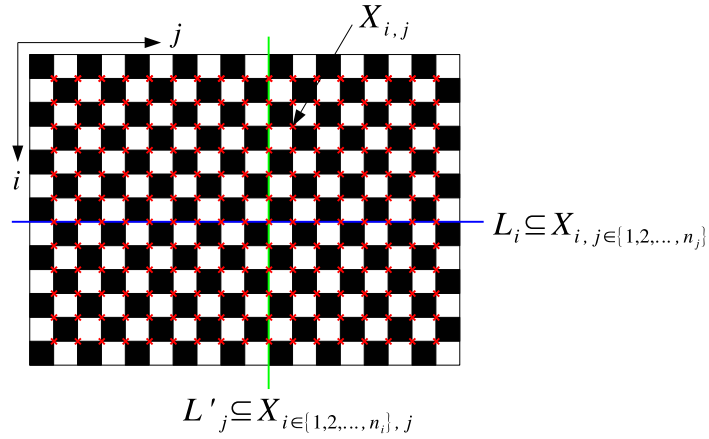


Figure 2.14: The grid points of the calibration target are defined as the corner points of the checkerboard squares with Euclidean coordinate  $\mathbf{X}_{i,j} \in \mathbb{R}^3$ . There are a total of  $n_i$  rows and  $n_j$  columns of grid points. A line  $\mathcal{L}_i$  is defined as the set of all points  $\mathbf{X}_{i,j \in \{1,2,\dots,n_j\}}$ , where  $\mathcal{L}$  is the set of parallel lines  $\mathcal{L}_{i \in \{1,2,\dots,n_i\}}$ . A line  $\mathcal{L}'_j$  is the set of all points  $\mathbf{X}_{i \in \{1,2,\dots,n_i\}, j}$ , where the set of parallel lines  $\mathcal{L}'$  includes all lines  $\mathcal{L}'_{j \in \{1,2,\dots,n_j\}}$ .

to lie on a single great circle  $G_{\eta_G, \eta_{G'}}$ ; this great circle is the fronto-parallel horizon of the plane in space in which the calibration target lies.

4. As the sets of parallel lines  $\mathcal{L}$  and  $\mathcal{L}'$  are orthogonal in space, the points  $\eta_G$  and  $\eta_{G'}$  satisfy the constraint  $\eta_G^T \eta_{G'} = 0$ .

The calibration algorithm uses only the position of the grid points  $\mathbf{u} = (u, v)^T$  detected in the image and not their exact relative Euclidean coordinates  $\mathbf{X}$  in space. However, each detected point  $\mathbf{u}_{i,j}$  needs to be indexed correctly whereby  $\mathbf{X}_{i,j} \mapsto \mathbf{u}_{i,j}$ . This ensures that the detected points in the image corresponding to any line  $\mathcal{L}_i$  or  $\mathcal{L}'_j$  are known. Then for a given estimate of the camera intrinsic parameters (including the image to sensor coordinate transform), each point  $\mathbf{u}_{i,j}$  can be mapped to a point  $\eta_{i,j}$  on the view sphere. If the camera model and intrinsic parameters were known precisely, constraints 1 through 4 would be satisfied. Using a simple plumb-line calibration algorithm, only constraint 1 would be used for calibration — for each set of points  $\mathbf{u}$  known to be collinear in space, a great circle would be fitted to the points detected in the image mapped to the sphere. However, the calibration algorithm described here is novel as it fits great circles to these points and enforces strictly constraints 2 through 4.

To enforce constraints 2 through 4, the position of the great circle  $G_{\eta_G, \eta_{G'}}$  and one of the antipodal pairs  $\pm \eta_G$  or  $\pm \eta_{G'}$  needs to be known. Only one of these antipodal pairs needs to be known as the other lies on the great circle  $G_{\eta_G, \eta_{G'}}$  and satisfies the constraint  $\eta_G^T \eta_{G'} = 0$ . This information is parameterised by the camera's extrinsic rotation  $R_f \in SO(3)$  which describes the relative orientation of the planar checkerboard calibration



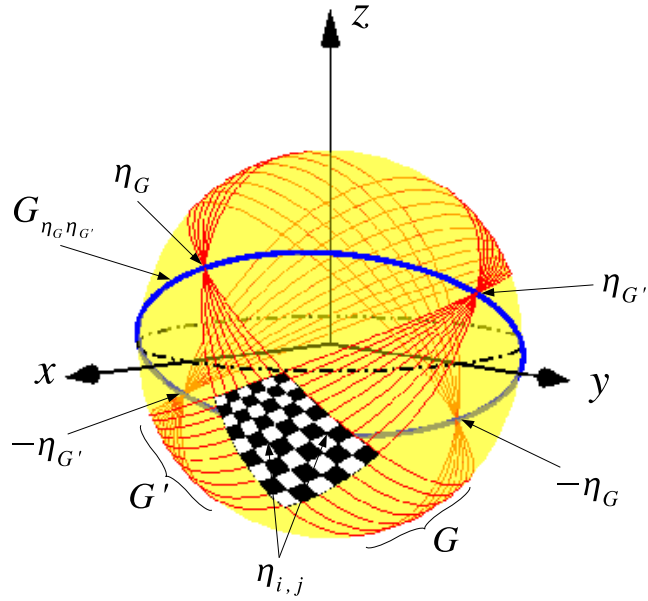


Figure 2.15: Constraints on grid points  $\mathbf{X}$  projected to the unit sphere. The sphere is centred at the origin of the calibration target coordinate frame of reference and the single effective viewpoint of the camera. All great circles  $G_i$  form the set of great circles  $G$  which intersect at antipodal points  $\pm\eta_G$ . All great circles  $G'_j$  form the set of great circles  $G'$  which intersect at antipodal points  $\pm\eta_{G'}$ . The antipodal points  $\pm\eta_G$  and  $\pm\eta_{G'}$  are constrained to lie on a single great circle  $G_{\eta_G, \eta_{G'}}$  where  $\eta_G^T \eta_{G'} = 0$ . The great circle  $G_{\eta_G, \eta_{G'}}$  is the fronto-parallel horizon of the plane in space in which the calibration target lies.

target with respect to the camera. The parametrisation of  $R_f$  will be discussed in detail in section 2.4.3. Note that position (translation) of the planar checkerboard pattern with respect to the camera does not need to be known.

For an input image of the checkerboard pattern, the calibration algorithms estimates the extrinsic rotation  $R_f$  and the camera intrinsic parameters. The camera intrinsic parameters include both the elements of affine matrix  $A$  and position of the principal point  $\mathbf{u}_0 = (u_0, v_0)^T$  in equation 2.5 (sensor to image coordinate transform), and the parameters specific to the camera model. The calibration algorithm can operate using multiple input images of the same checkerboard pattern. When multiple images are used, each image has its own unique extrinsic rotation  $R_f$ .

## 2.4.2 Objective Function

A suitable objective function is presented here which measures quantitatively how accurately the calibration constraints are satisfied for a given estimate of the camera

intrinsic parameters and rotation  $R_f$ .

As the unit vectors defined by the points  $\eta_G$  and  $\eta_{G'}$  are orthogonal,  $\eta_G^T \eta_{G'} = 0$ . There exists then a rotation matrix  $R_f \in SO(3)$  for which

$$\eta_G = R_f (1, 0, 0)^T = R_f \eta_{\tilde{G}}, \quad (2.47)$$

$$\eta_{G'} = R_f (0, 1, 0)^T = R_f \eta_{\tilde{G}'}, \quad (2.48)$$

since  $(1, 0, 0)(0, 1, 0)^T = 0$ . This rotation matrix  $R_f$  would rotate any point on the equator of the sphere to a point on the great circle  $G_{\eta_G, \eta_{G'}}$ . For an estimate of the rotation matrix  $R_f$ , let  $\tilde{\eta}$  be the set of points  $\eta$  rotated to a new position on the sphere by

$$\tilde{\eta}_{i,j} = R_f^T \eta_{i,j}, \quad (2.49)$$

where each point  $\eta_{i,j}$  is found by projecting the detected grid point  $\mathbf{u}_{i,j}$  in the wide-angle image to the sphere for the estimate of the camera's intrinsic parameters. Denote the sets of great circles associated with the points  $\tilde{\eta}$  as  $\tilde{G}$  and  $\tilde{G}'$ .

From equations 2.47 and 2.48, the sets of great circles  $\tilde{G}$  and  $\tilde{G}'$  corresponding to the points  $\tilde{\eta}$  will intersect at antipodal points  $\pm \eta_{\tilde{G}} = (\pm 1, 0, 0)^T$  and  $\pm \eta_{\tilde{G}'} = (0, \pm 1, 0)^T$  respectively. Each great circle  $\tilde{G}_i$  can therefore be rotated to lie on the equator by  $R_y(\zeta_i)^T$ , and each great circle  $\tilde{G}'_j$  rotated to lie on the equator by  $R_x(\xi_j)^T$ . The rotation matrices  $R_y(\zeta)$  and  $R_x(\xi)$  are rotations about the  $y$  and  $x$  axes respectively:

$$R_y(\zeta) = \begin{bmatrix} \cos \zeta & 0 & \sin \zeta \\ 0 & 1 & 0 \\ -\sin \zeta & 0 & \cos \zeta \end{bmatrix}, \quad (2.50)$$

$$R_x(\xi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \xi & -\sin \xi \\ 0 & \sin \xi & \cos \xi \end{bmatrix}. \quad (2.51)$$

Great circles need to be fitted to the sets of points  $\tilde{\eta}$ . Perfect fits will not occur due to: errors in the estimate of the grid point positions detected in the image plane, inability of the camera model to describe perfectly the process of image formation, and incorrect estimates for the camera intrinsic variables and extrinsic rotation  $R_f$ . The set of points  $\tilde{\eta}_{i,j \in \{1,2,\dots,n_j\}}$  are constrained to lie on the great circle  $\tilde{G}_i$ . It is proposed that the great circle  $\tilde{G}_i$  which best fits the set of points  $\tilde{\eta}_{i,j \in \{1,2,\dots,n_j\}}$  is the one which minimises the closest angular distance of the fitted great circle to these points. As  $\tilde{G}_i$  can be rotated to lie on the equator by  $R_y(\zeta_i)^T$ , then the sum of squared distances of



the points  $\tilde{\eta}_{i,j \in \{1,2,\dots,n_j\}}$  from  $\tilde{G}_i$  is the angular distance from the equator of the points  $\tilde{\eta}_{i,j \in \{1,2,\dots,n_j\}}$  rotated by  $R_y(\zeta_i)^T$ . The sum of squared distances is

$$\rho_i = \sum_{j=1}^{n_j} [\arcsin((R_y(\zeta_i)^T \tilde{\eta}_{i,j})_z)]^2 \quad (2.52)$$

$$= \sum_{j=1}^{n_j} [\arcsin((\sin(\zeta_i), 0, \cos(\zeta_i)) \tilde{\eta}_{i,j})]^2. \quad (2.53)$$

A similar analogy can be used to define the sum of squared distances of points  $\tilde{\eta}_{i \in \{1,2,\dots,n_i,j\}}$  from the great circle  $\tilde{G}'_j$ , and it is

$$\rho'_j = \sum_{i=1}^{n_i} [\arcsin((R_x(\xi_j)^T \tilde{\eta}_{i,j})_z)]^2 \quad (2.54)$$

$$= \sum_{i=1}^{n_i} [\arcsin((0, -\sin(\xi_j), \cos(\xi_j)) \tilde{\eta}_{i,j})]^2. \quad (2.55)$$

For small angles, the errors defined in equations 2.52 and 2.54 are approximated respectively as

$$\rho_i = \sum_{j=1}^{n_j} [(\sin(\zeta_i), 0, \cos(\zeta_i)) \tilde{\eta}_{i,j}]^2 \quad (2.56)$$

and

$$\rho'_j = \sum_{i=1}^{n_i} [(0, -\sin(\xi_j), \cos(\xi_j)) \tilde{\eta}_{i,j}]^2. \quad (2.57)$$

For any point  $\tilde{\eta}_{i,j}$  on the great circle  $\tilde{G}_i$ , the angle  $\zeta_i$  defined in 2.50 which rotates the point exactly to the equator is found from

$$\tilde{\eta}_{i,j}(x) \sin(\zeta_i) + \tilde{\eta}_{i,j}(z) \cos(\zeta_i) = 0, \quad (2.58)$$

and for any point  $\tilde{\eta}_{i,j}$  on the great circle  $\tilde{G}'_j$ , the angle  $\xi_j$  defined in 2.51 which rotates the point exactly to the equator is found from

$$-\tilde{\eta}_{i,j}(y) \sin(\xi_j) + \tilde{\eta}_{i,j}(z) \cos(\xi_j) = 0. \quad (2.59)$$

For all points  $\tilde{\eta}_{i,j \in \{1,2,\dots,n_j\}}$  constrained to lie on the great circle  $\tilde{G}_i$ , a solution for

$\sin(\zeta_i)$  and  $\cos(\zeta_i)$  is obtained from the set of simultaneous linear equations

$$\begin{bmatrix} \tilde{\eta}_{i,1}(x) & \tilde{\eta}_{i,1}(z) \\ \tilde{\eta}_{i,2}(x) & \tilde{\eta}_{i,2}(z) \\ \vdots & \vdots \\ \tilde{\eta}_{i,n_i}(x) & \tilde{\eta}_{i,n_i}(z) \end{bmatrix} \begin{bmatrix} \sin(\zeta_i) \\ \cos(\zeta_i) \end{bmatrix} = \mathbf{0} \quad (2.60)$$

from which  $\zeta_i = \arctan(\sin(\zeta_i) / \cos(\zeta_i))$ . For all points  $\tilde{\eta}_{i \in \{1,2,\dots,n_i\},j}$  constrained to lie on the great circle  $\tilde{G}'_j$ , a solution for  $\sin(\xi_j)$  and  $\cos(\xi_j)$  is obtained from the set of simultaneous linear equations

$$\begin{bmatrix} -\tilde{\eta}_{1,j}(y) & \tilde{\eta}_{1,j}(z) \\ -\tilde{\eta}_{2,j}(y) & \tilde{\eta}_{2,j}(z) \\ \vdots & \vdots \\ -\tilde{\eta}_{n_i,j}(y) & \tilde{\eta}_{n_i,j}(z) \end{bmatrix} \begin{bmatrix} \sin(\xi_j) \\ \cos(\xi_j) \end{bmatrix} = \mathbf{0} \quad (2.61)$$

from which  $\xi_j = \arctan(\sin(\xi_j) / \cos(\xi_j))$ . Note that both equations 2.60 and 2.61 are of the form  $A\mathbf{x} = 0$ . Letting  $USV = \text{svd}(A)$  be the singular value decomposition of the matrix  $A$ , the solution for  $\mathbf{x}$  which minimises the sum of squared errors  $A\mathbf{x}$  is the column vector of  $V$  corresponding to the smallest non-zero singular value  $\text{diag}(S)$  subject to the condition  $\|\mathbf{x}\| = 1$ . As  $\|\mathbf{x}\| = 1$ , the values for  $\sin(\zeta_i)$  and  $\cos(\zeta_i)$  correspond directly to the values of  $\mathbf{x}$  as  $\sin^2(\zeta_i) + \cos^2(\zeta_i) = 1$ . The same is true for the values  $\sin(\xi_j)$  and  $\cos(\xi_j)$ . In both cases, the solutions for the angles  $\zeta_i$  and  $\xi_j$  are found which minimise the sum of squared errors  $\rho_i$  and  $\rho'_j$  in equations 2.56 and 2.57 respectively. It is proposed that for the camera model selected, the most accurate estimate for the camera intrinsic values and extrinsic rotation  $R_f$  which satisfies constraints 1 through 4 for a given image are those which minimise the quantitative objective function error

$$\varepsilon = \sum_{i=1}^{n_i} \rho_i + \sum_{j=1}^{n_j} \rho'_j. \quad (2.62)$$

In the following experiments the camera is calibrated using multiple images of the same checkerboard pattern. Although the same camera intrinsic values apply to all images, there will be a unique extrinsic rotation  $R_f$  and objective function error  $\varepsilon_k$  defined in equation 2.62 for each. For  $N$  input images, the overall objective function error  $\hat{\varepsilon}$  is

$$\hat{\varepsilon} = \sum_{k=1}^N \varepsilon_k. \quad (2.63)$$

### 2.4.3 Parametrisation and initialisation of camera extrinsic rotation $R_f$

The rotation matrix  $R_f \in SO(3)$  is parameterised by Euler angles  $\alpha, \beta, \gamma$ , where

$$R_f = R_z(\gamma)R_y(\beta)R_z(\alpha), \quad (2.64)$$

and  $R_y$  and  $R_z$  are rotations about the  $y$  and  $z$  axes respectively. If desired,  $R_f$  could be parameterised using quaternions. Given an initial estimate of the camera intrinsic parameters, the position of the detected grid points  $\mathbf{u}$  in the image are mapped to the points  $\eta$  on the unit sphere and an estimate for the Euler angles  $\alpha, \beta, \gamma$  found. This estimate is used for the first iteration of calibration. The position of the antipodal points of intersection  $\eta_G$  and  $\eta_{G'}$  are estimated using the four outermost grid points  $\eta$  as

$$\eta_G = (\eta_{1,1} \times \eta_{1,n_j}) \times (\eta_{n_i,1} \times \eta_{n_i,n_j}), \quad \|\eta_G\| = 1 \quad (2.65)$$

and

$$\eta_{G'} = (\eta_{1,1} \times \eta_{n_i,1}) \times (\eta_{1,n_j} \times \eta_{n_i,n_j}), \quad \|\eta_{G'}\| = 1, \quad (2.66)$$

where  $\times$  denotes the vector cross product.

As both  $\eta_G$  and  $\eta_{G'}$  are constrained to lie on the great circle  $G_{\eta_G, \eta_{G'}}$ , which is the intersection of the fronto-parallel plane in space containing the calibration target and the sphere, the unit vector  $\mathbf{N} = (N_x, N_y, N_z)^T$  normal to this plane is

$$\mathbf{N} = \eta_G \times \eta_{G'}, \quad \|\mathbf{N}\| = 1. \quad (2.67)$$

An estimate of the Euler angles  $\beta$  and  $\gamma$  are obtained as

$$\beta = -\arcsin(N_z) + \frac{\pi}{2} \quad (2.68)$$

$$\gamma = \arctan\left(\frac{N_y}{N_x}\right). \quad (2.69)$$

Define then the point  $\eta'_G$  as

$$\eta'_G = (R_z(\gamma)R_y(\beta))^T \eta_G \quad (2.70)$$

which lies at some point on the equator. The estimate of the angle  $\alpha$  obtained is

$$\alpha = \arctan \left( \frac{\eta'_G(y)}{\eta'_G(x)} \right). \quad (2.71)$$

#### 2.4.4 Implementation

The calibration algorithm is implemented as follows, where the methods used for optimisation and criteria for terminating optimisation will be discussed in detail later:

1. For the current estimate of the camera intrinsic parameters, map the grid points  $\mathbf{X}$  detected at pixel locations  $\mathbf{u}$  in each image to points  $\eta$  on the unit sphere.
2. If it is the first iteration, initialise for each image the estimate of the camera extrinsic parameters (Euler angles)  $\alpha, \beta, \gamma$  which defined the rotation matrix  $R_f = R_z(\gamma)R_y(\beta)R_z(\alpha)$ , otherwise use the estimate from the previous iteration.
3. For each image, find the rotation matrix  $R_f$  which minimises the objective function error  $\varepsilon$  given in equation 2.62. This requires for each estimate of the rotation matrix  $R_f$  that each point  $\eta_{i,j}$  is rotated to a new position  $\tilde{\eta}_{i,j}$  — see equation 2.49.
4. Set the error  $\hat{\varepsilon}$  for the current estimate of the camera intrinsic values as the sum of all errors  $\varepsilon_k$  for each image — see equation 2.63.
5. Repeat from step 1 for a maximum of  $n$  iterations or until convergence of the objective function error  $\hat{\varepsilon}$ .

The Euler angles  $\alpha, \beta, \gamma$  for each image and the camera intrinsic parameters are found using a non-linear optimisation. For each estimate of the camera intrinsic parameters, the Euler angles are found using a non-linear optimisation with Matlab's 'lsqnonlin' function (Levenberg-Marquardt). The default options are selected, and iteration continues until convergence or a maximum of 2000 iterations is exceeded. The camera intrinsic parameters are found using a non-linear optimisation with Matlab's 'fminsearch' function (Nelder-Mead Simplex). Again, the default options are used, and the optimisation terminates when the error function  $\hat{\varepsilon}$  converges or a maximum of 5000 iterations is exceeded. These intrinsic parameters include the parameters specific to the model being used, and the parameters of the affine matrix and principal point which defines the sensor to image coordinate transform in equation 2.5.

### 2.4.5 Grid Point Detection

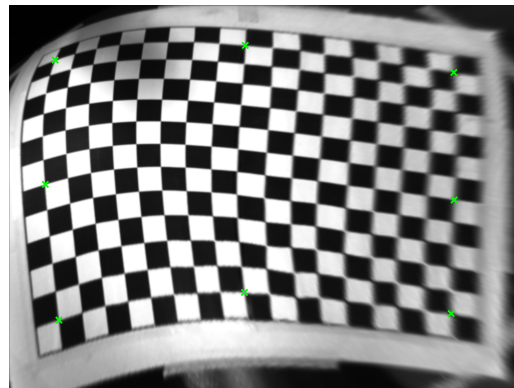
The pixel position  $\mathbf{u}_{i,j}$  of each grid point needs to be found in an image for calibration. Unfortunately, keypoint detectors such as the Harris corner detector [94] and SIFT [142] (both discussed in chapter 3) were unable to reliably detect the grid points in the image. The reason was the number of false positive produced, where without an initial estimate of the grid points, they cannot be easily removed. If a perspective camera were used, manually selecting the 4 outermost corners of the grid would allow the position of the remaining grid points to be estimated easily. However, this is not possible in the fisheye image due to the extreme radial distortion in the image.

A robust grid point detection algorithm is proposed here which is semi-supervised and operates in parallel with camera calibration. The general procedure is illustrated in figure 2.16 using a sample image of the checkerboard target obtained with the fisheye camera. Given the initial image of the calibration target the estimated position of the four outermost grid points are selected manually, as shown in figure 2.16a. These four points are sufficient to define two orthogonal sets of parallel lines in space, each containing two lines. A preliminary calibration step is implemented with these four corner points using the calibration algorithm outlined previously. The unified image model of Geyer and Daniilidis [86] is used as default for grid point detection, where the image is assumed to have equal scaling in the  $u, v$  directions and zero shear (with respect to equation 2.5,  $A = I_{2 \times 2}$  is the  $2 \times 2$  identity matrix). This model is selected as it contains only two parameters  $l$  and  $m$  in addition to the position of the principal point  $\mathbf{u}_0$ , and as will be shown in later experiments, is able to model the radial distortion with excellent accuracy. Stereographic projection is selected as the initial estimate ( $l = 1$ ), and the distance from the sphere to the image plane  $m$  is selected given the approximate field of view of the camera (obtained from manufacturer data as  $190^\circ$ ). The position of the principal point is set simply as  $\mathbf{u}_0 = (nc/2, nr/2)^T$ , where  $nc$  and  $nr$  are the number of image pixels in the  $u$  and  $v$  directions respectively. If desired a more accurate estimate could be obtained by fitting a circle to the boundary of the camera's field of view.

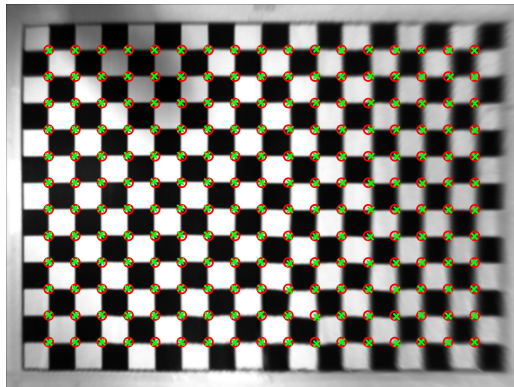
The initial calibration step finds an estimate of both the camera intrinsic parameters and the extrinsic rotation matrix  $R_f$  for the image. An orthonormal perspective image is then produced using these estimates, where linear interpolation is used to sample from the original image — the normal to the calibration target plane points directly out of the page. This orthonormal perspective image is twice the size of the original image, and a suitable uniform rescaling and translation is used to ensure the calibration target extends to the boundaries in this image, as shown in figure 2.16b.



(a) Initial image with 4 corner points selected manually.



(b) Reconstructed orthonormal view obtained from the first calibration step. The position of the 8 points used for the next calibration step selected from this image are shown.



(c) Reconstructed orthonormal view after second calibration step. The initial estimate of the grid points (circles) and the final position of the grid points found (crosses) are shown.



(d) Position of grid points mapped back to the original image. This is the input data used for full calibration.

Figure 2.16: General procedure of the semi-supervision grid point detection operating in parallel with camera calibration.

The four outermost grid points at each corner of the calibration target are reselected on this orthonormal perspective view. A grid point along each of the outermost edges of the calibration target is also selected to give eight points in total. These eight points are selected on two sets of orthogonal parallel lines, each set containing two lines. The automatic corner detector described shortly is used for selecting these points, the user needs to simply select manually a small local region surrounding each. The calibration algorithm is run again and a new estimate for the camera intrinsic values and extrinsic rotation  $R_f$  found. A new orthonormal perspective view is then produced which is again twice the size of the original image where a suitable rescaling and translation is applied to ensure the target extends near the boundaries of this image, as shown in figure 2.16c. It can be observed from the figure that a reasonably accurate orthonormal

perspective image is produced after calibration using as few as eight points (two sets of orthogonal parallel lines).

This final orthonormal perspective view is used to detect the grid point positions. As the position of the four outermost corners of the grid are already known, then given the number of grid points  $n_i$  and  $n_j$  in each direction a priori, the estimated position of all remaining grid points can be easily found. For the example shown, the estimated grid point positions are shown as red circles in figure 2.16c. The ability to estimate these grid point positions with relatively high accuracy is the primary advantage of the grid point detection algorithm as it permits a constrained local search space when finding each individual grid point. Starting at the uppermost corner, the automatic grid point detection algorithm is used to find the accurate position of the grid points.

The automatic grid point detection algorithm takes as input a small local patch centred around the estimated position of a grid point. The size of this patch is taken to be the estimated size of a single checkerboard square (measured in pixels) in the orthonormal perspective image. A binary threshold is applied to the patch, where the threshold is set as the mean intensity value to account for illumination variations (assumes that there should be an equal distributions of light and dark regions). Dilation is then applied and the resulting regions labelled using the Matlab image processing toolbox of Corke [191]. A grid point is assumed found when a minimum of three or a maximum of four unique regions are found. In the case of four, the grid point is taken as the intersection of all regions. For the most common case where three regions are found, the grid point is taken as the midpoint between the connecting line(s) of minimum distance between the separated regions. If there are multiple lines, the mean of the midpoints is taken as the grid point position. The automatic grid point detector is demonstrated in figure 2.17 for the four corner points of the calibration target in figure 2.16a. Notice that in each case the resolution of the local patch used for grid point detection is high (approximately  $100 \times 100$  pixels in most cases). This high resolution ensures that although the image processing steps operate on a pixel-wise basis, the position of the grid point found is of sub-pixel accuracy with respect to the original fisheye image. The position of the grid points found using the automatic method is shown by the green crosses in figure 2.16c. The position of each grid point is then mapped back to a point  $\mathbf{u}$  in the original fisheye image, as shown in figure 2.16d.

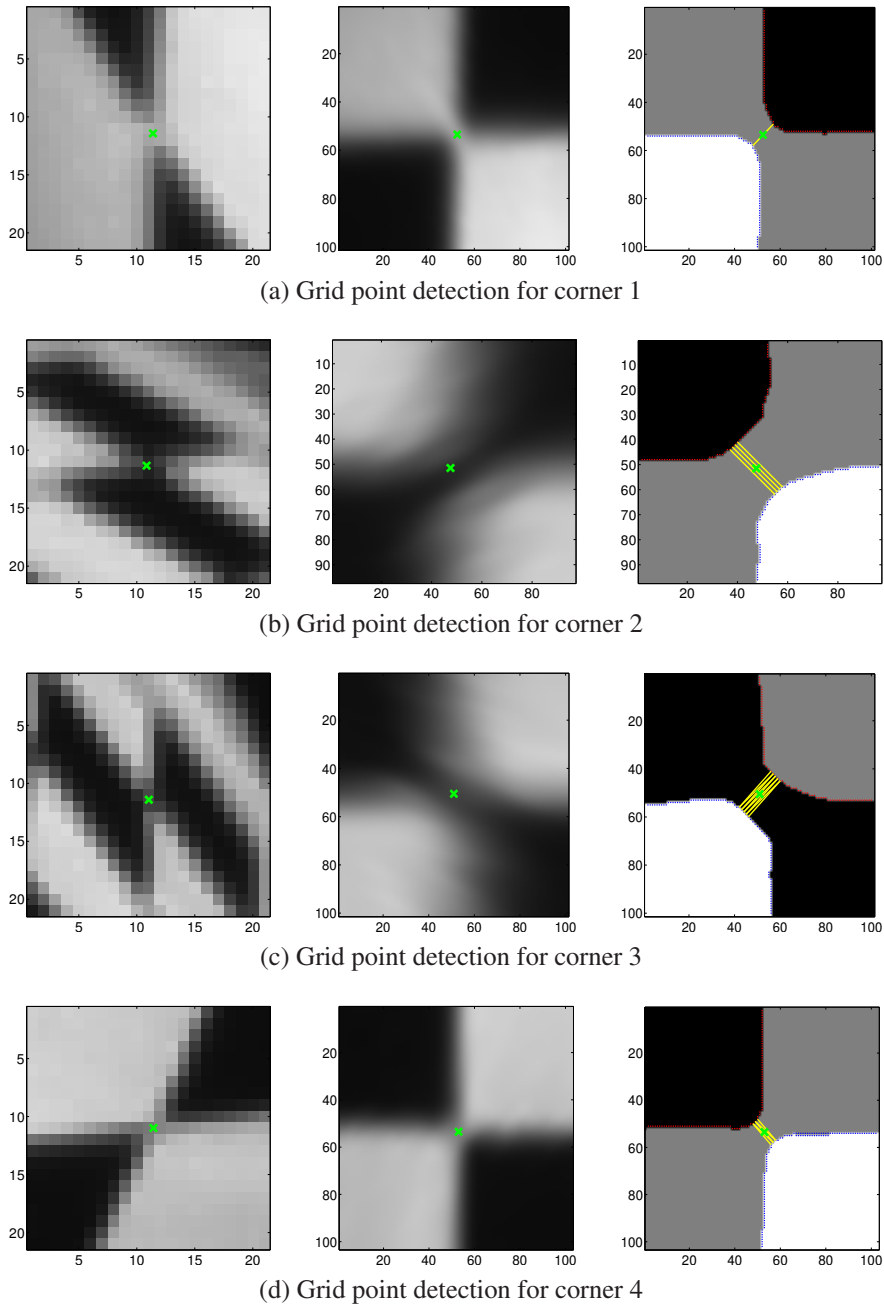


Figure 2.17: Example grid point detection for each of the four outermost grid points given the local patch surrounding each. The left column shows the local regions on the original fisheye image, the middle column shows the patch taken from the orthonormal perspective view, and the right shows the labelled regions after binary thresholding and dilation. The yellow lines in the right column are the line(s) of minimum distance between the separated regions. The green cross in all images shows the final positions of the grid points — this is taken to be the mean midpoint of the line(s) of minimum distance between the regions. The axes have units of pixels.



## 2.4.6 Calibration Example

The calibration algorithm is demonstrated here on the fisheye camera using the set of ten  $1024 \times 768$  pixel input images of the planar checkerboard pattern shown in figure 2.18. Ten input images have been used to ensure that data is available over the full field of view of the camera. Using only image 1 for example would not include grid points near the outermost periphery of the image. The position of the grid points in each image defined by their pixel coordinates  $\mathbf{u} = (u, v)^T$  are found using the grid point detection algorithm described in section 2.4.5. The camera was assumed to be central projection and was calibrated using the unified image model. The intrinsic parameters are the position of the principal point  $\mathbf{u}_0$  and the parameters of the camera model  $l, m$ . For this initial demonstration, the camera was modelled as having equal scaling in the  $u, v$  directions and zero shear.

The results for camera calibration are depicted visually for image 6 in figure 2.19 and given in the top row of table 2.5 (pg.79). Figure 2.19a shows the sets of fitted great circles  $G$  and  $G'$  on the sphere which intersect at antipodal points  $\pm\eta_G$  and  $\pm\eta_{G'}$  respectively. The blue line is the great circle  $G_{\eta_G, \eta_{G'}}$  on which the antipodal points  $\pm\eta_G$  and  $\pm\eta_{G'}$  are constrained to lie and defines the fronto-parallel horizon of the plane in space containing the calibration target. The same set of great circles are represented on the original fisheye image in figure 2.19b and on the reconstructed orthonormal perspective view in figure 2.19c. In all figures, the green crosses indicate the positions of the grid points found using the grid point detection algorithm. The same results shown in figures 2.19b and 2.19c can be found for each of the input images in figures A.1 through A.10 in appendix A. The relationship between the angle of colatitude  $\theta$  on the sphere versus radius  $r$  on the image plane from the principal point found from calibration is shown in figure 2.20.

Although the calibration algorithm minimises the objective function error  $\hat{\epsilon}$ , which is the sum of errors  $\epsilon_k$  for each image defined in equation 2.62, the reprojection errors for each image can be found which is useful for interpreting the results. These reprojection errors are the Euclidean distances measured in the fisheye image between the position of the grid points found during detection, and the position of the grid points defined as the intersection of the fitted great circles — the points of intersection of the fitted great circles on the sphere need to be mapped to the fisheye image.

Given the rotation matrix  $R_f$  and the set of all fitted great circles  $\tilde{G}$  and  $\tilde{G}'$  defined by the angles  $\zeta$  and  $\xi$  respectively, the intersections of all the great circles  $G$  and  $G'$  can be found. Recall that any great circle  $G_i$  can be mapped backed to the equator by a

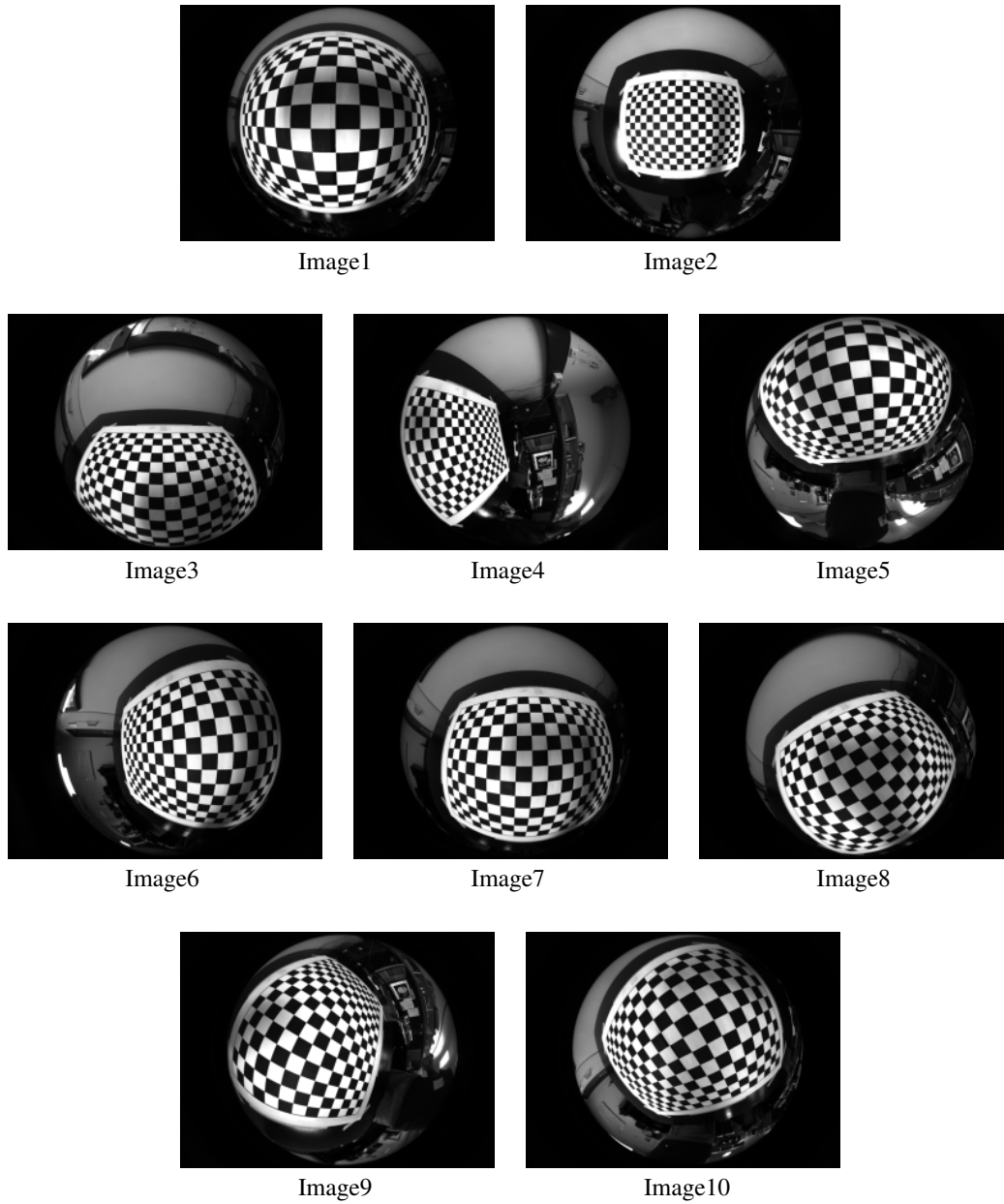
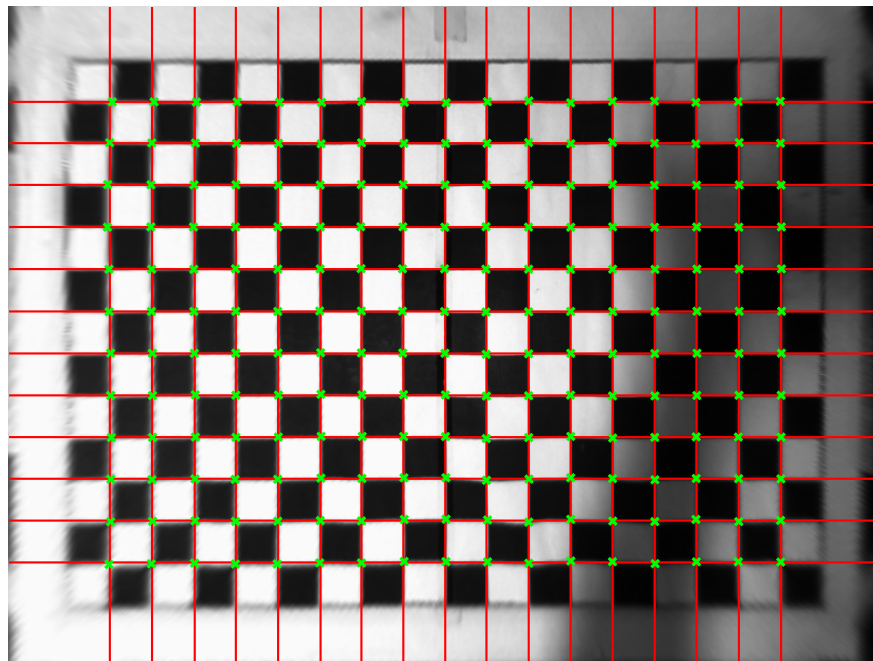
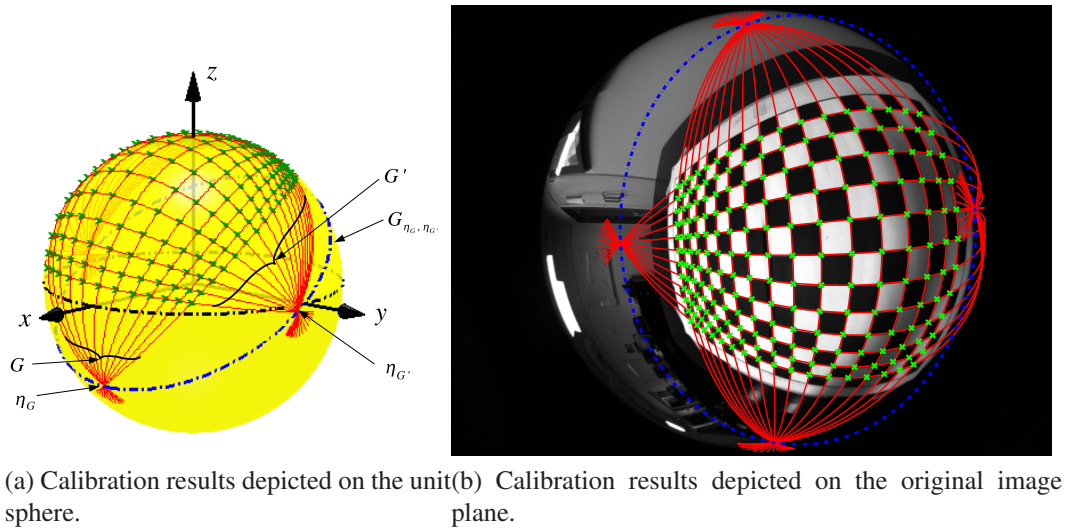


Figure 2.18: The 10 input fisheye images used for calibration. Each image is of size  $1024 \times 768$  pixels.

rotation  $R_y(\zeta_i)^T R^T$  and any great circle  $G'_j$  by a rotation  $R_x(\xi_j)^T R^T$ . If any great circle is defined as the intersection of a plane and the sphere, where the origin of the sphere lies in the plane, then a great circle can be defined by the normal to this plane. For  $G_i$  and  $G'_j$  these normals are

$$\mathbf{N}_{G_i} = (R_f(R_y(\zeta_i)(1, 0, 0)^T)) \times (R_f(R_y(\zeta_i)(0, 1, 0)^T)), \quad \|\mathbf{N}_{G_i}\| = 1 \quad (2.72)$$



(c) Calibration results depicted on the orthonormal perspective plane.

Figure 2.19: Calibration results for image 6. The green crosses show the position of the grid points found using the grid point detection algorithm. The red lines illustrate the fitted great circles on the sphere. The blue line ( $G_{\eta_G, \eta_{G'}}$ ) is the projection of the front-parallel horizon of the plane containing the planar checkerboard pattern on the sphere.

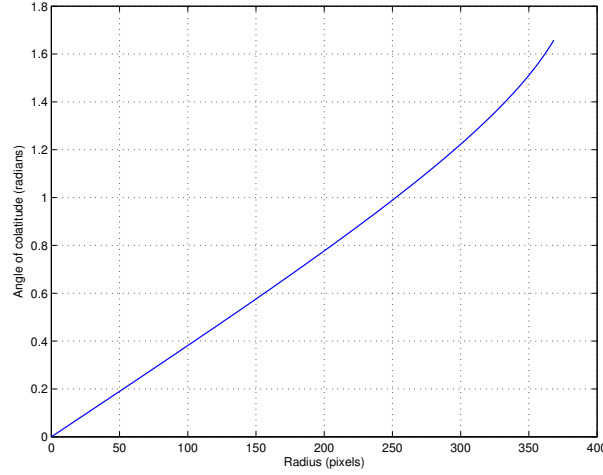


Figure 2.20: Unified image model function found from calibration. The figure shows the angle of colatitude on the sphere versus radius from the principal point in the fish-eye image.

and

$$\mathbf{N}_{G'_j} = (R_f(R_x(\xi_j)(1,0,0)^T)) \times (R_f(R_x(\xi_j)(0,1,0)^T)), \quad \|\mathbf{N}_{G'_j}\| = 1. \quad (2.73)$$

Note that it is necessary to define the normals using this approach since simply using  $\mathbf{N}_{G_i} = R_f(R_y(\zeta_i)(0,0,1)^T)$  for example would be subject to ‘gimbal lock’ if  $\zeta_i = 0$  as the first term of  $R_f$  is a z-axis rotation.

The intersection of two planes defined by the great circles  $G_i$  and  $G'_j$  is a line which passes through the centre of the sphere and intersects it at two antipodal points  $\eta_{G_i, G'_j}$ . These antipodal points are the intersections of the great circles  $G_i$  and  $G'_j$ :

$$\eta_{G_i, G'_j} = \pm(\mathbf{N}_{G_i} \times \mathbf{N}_{G'_j}), \quad \|\eta_{G_i, G'_j}\| = 1. \quad (2.74)$$

To determine which antipodal point is the correct estimate of the grid point position, both are mapped back to the wide-angle image (assuming this is possible) using the camera intrinsic values obtained from calibration. The one which is nearest to the grid point position found during detection is selected.

Figure 2.21 illustrates the reprojection errors of all grid points for each of the ten input images. Each of the marks represents, in each of the fisheye images, the difference in the position of the reprojected grid point coordinates (intersections of the fitted great circles) relative the coordinates of their corresponding grid points found during detection. The results show good correspondence between the estimated and detected grid point positions where the mean  $x, y$  difference in the relative positions for each

Model	Intrinsic values	Mean reprojection error (pixels)	Error ( $\hat{\epsilon}$ )
Unified	$l = 2.7899$ $u_0 = 528.1214$ $v_0 = 384.0784$ $m = 996.4617$ $s = 0$ $s_y = 1$	$median = 0.3568$ $mean = 0.4170$	0.00673
Unified (affine)	$l = 2.7902$ $u_0 = 528.1214$ $v_0 = 384.0786$ $m = 996.4617$ $s = 1.4824 \times 10^{-16}$ $s_y = 1.0000$	$median = 0.3578$ $mean = 0.4167$	0.00672

Table 2.5: Summary of the calibration results for the unified and unified (affine) camera models using the 10 input images in figure 2.18. Observe that a smaller error  $\epsilon$  on the sphere corresponds to a smaller reprojection error measured in the fisheye image.

image is very close to zero. The reprojection errors for all ten images are shown versus their distance from the principal point in figure 2.22a, and as a probability density function in figure 2.22b. It is observed in figure 2.22a that the reprojection errors remain approximately uniform for all radii  $r$ . All of these results suggest that the method of calibration is sound, and that the unified camera model is able to model the fisheye camera with a high degree of accuracy. The specific camera intrinsic values, mean reprojection error for all images, and the error  $\hat{\epsilon}$  are given in the top row of table 2.5.

The example shown assumed equal scaling in the  $u, v$  pixel directions and zero shear, where the matrix  $A$  in equation 2.5 is the  $2 \times 2$  identity matrix. To determine if there is some affine component in the sensor to image plane coordinate transform, the camera was calibrated again with the unified image model, using the same set of grid points, where an initial estimate for the affine matrix  $A$  was set to

$$A = \begin{bmatrix} 1 & s \\ 0 & s_y \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad (2.75)$$

where both  $s$  and  $s_y$  were included in the set of camera intrinsic parameters during calibration. This will be termed the unified (affine) model. The camera intrinsic values and errors  $\hat{\epsilon}$  for each model are given in table 2.5. Figure 2.23 shows for each model the boxplot of combined reprojection errors in all ten images.

The results in table 2.5 and figure 2.23 show that when compared to the unified model, the unified (affine) model is only able to achieve a reduction in the mean reprojection error of 0.0003 pixels. It can also be observed that the values  $s$  and  $s_y$  defined in equation 2.75 are very close to 0 and 1 respectively. For these reasons, in the following experiments where numerous models are compared, the sensor to image coordinate transform is assumed to have equal scaling and zero shear (matrix  $A$  defined

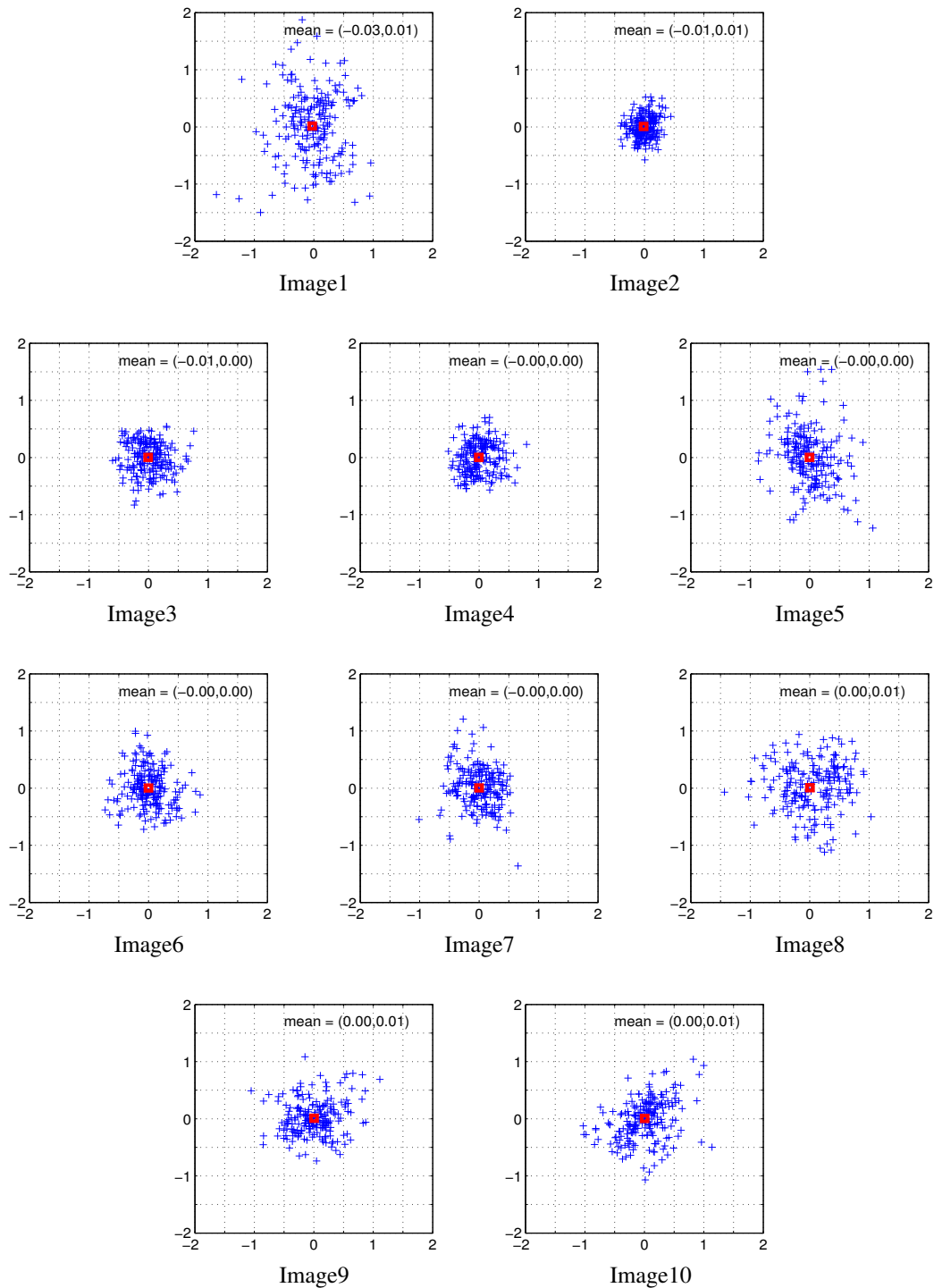
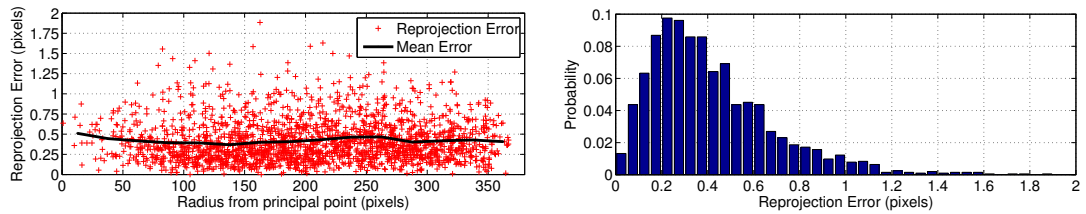


Figure 2.21: The relative positions in the fisheye images of the estimated grid point coordinates found during calibration (intersections of fitted great circles) relative to the coordinates of their corresponding points found during grid point detection. The red square in each is the mean difference in the relative positions — the mean  $x, y$  differences are given in the upper right corners. All axes have units of pixels.



(a) Reprojection errors versus radius from principal point (centre of distortion). The mean shown is the mean value of all reprojection errors over fixed intervals of 25 pixels.

(b) Probability density function of reprojection errors (mean = 0.417 pixels, median = 0.357 pixels).

Figure 2.22: Distribution of reprojection errors versus radius  $r$  from the camera's principal point, and the probability distribution of the reprojection errors.

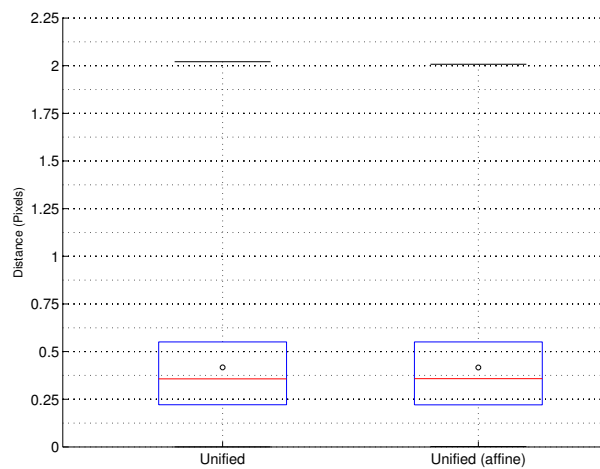


Figure 2.23: Boxplot comparison of the reprojection errors in the fisheye image plane for the unified and unified (affine) camera models. The mean values are indicated by the circle.

in equation 2.5 is fixed as the  $2 \times 2$  identity matrix).

## 2.4.7 Experiments and Results

The aim of these experiments is to identify, the camera model most suited to the fisheye camera. The most suitable model is determined to be the one with the smallest residual objective function error  $\hat{\epsilon}$  after calibration. In the following experiments only the ray-based models are considered, and camera is assume to have only radial distortion. Therefore, referring to table 2.4, the models considered and are:

- |                    |                           |   |
|--------------------|---------------------------|---|
| 1. Stereographic   | 4. Combination ([9])      | 7. Polynomial (1,3) ( $r = k_1\theta + k_2\theta^3$ ) |
| 2. Equiangular     | 5. Mičušík 1 ([164, 162]) | 8. Polynomial (1,3,5) ([101])                         |
| 3. Equisolid angle | 6. Mičušík 2 ([162])      | 9. Unified ([86])                                     |



The sine-law model is not used as it is not suited for a camera with in excess of a hemispherical field of view. The rational model [45] is also not considered as it requires calibrating for 16 variables — as many of the other models with fewer parameters in the table have been used with success for accurate camera calibration, only these are considered. A similar argument can be made for the model of Kanalla and Brandt [118, 119] which contains a high number of variables and includes both radial and tangential distortions. Although the radial component could be used by itself, it includes polynomial terms up to the ninth power. As noted by Basu and Licardie [14], this requires the use of iterative techniques to solve the roots of the polynomial and it becomes difficult to obtain an accurate initial estimate using non-linear calibration. With respect to polynomials, the model used by Xiong and Turkowski [243] is not used as Ho [101] was able to find more accurate calibration using their model for a number of different wide-angle cameras. However, to validate the claim that higher order polynomials are able to more accurately model the radial distortion, the results for the polynomial  $r = k_1\theta + k_2\theta^3$  are also found.

#### 2.4.7.1 Initial Model Estimates

The calibration algorithm attempts to find the camera intrinsic parameters and the extrinsic rotation  $R_f$  for each image which minimises the objective function error  $\hat{\epsilon}$ . This is achieved using non-linear optimisation, and a potential limitation of this approach is convergence on solutions for which  $\hat{\epsilon}$  is a local minima. Since the results using the unified image model appear to model the camera with high accuracy (albeit without comparison to other camera models), the extrinsic rotations  $R_f$  for each image and the position of the principal point  $\mathbf{u}_0$  found from calibration are used as the initial estimates for all other models. Furthermore, the radial distortion function found using the unified image model shown in figure 2.20 is used to initiate the estimates of the intrinsic parameters (specific to the radial distortion) for the remaining models before calibration. This is of particular importance for the polynomial models.

The initial estimates for each camera models intrinsic parameters are found as follows. For  $n = 200$  equally spaced radii on the image  $r \in [0, r_{max}]$ , the angles of colatitude  $\theta(r)$  are found using the unified image model results found in the calibration example. The maximum radius  $r_{max}$  is the radius on the image corresponding to an angle of colatitude of  $\theta = 190^\circ/2$  ( $190^\circ$  being the cameras maximum field of view). Then for any of the remaining camera models, define  $\tilde{\theta}(r_i)$  as the angle of colatitude corresponding to radius  $r_i$  on the image for the estimate of the model parameters. The initial estimates of the camera model parameters are found prior to calibration by minimising



the sum of squared errors

$$\varepsilon = \sum_{i=1}^n (\theta(r_i) - \tilde{\theta}(r_i))^2. \quad (2.76)$$

This is implemented as a non-linear optimisation. The initial estimates for each camera model are shown as a function of the angle of colatitude  $\theta$  versus radius on the image plane  $r$  in figure 2.24. The original function obtain from calibration using the unified image model is included in each figure for reference. Given that the unified image model was able to calibrate the camera with high accuracy, this is a valid means for obtaining an initial estimate of the camera model parameters prior to calibration.

### 2.4.7.2 Results and Discussion

The fisheye camera was calibrated for each camera model considered using all ten input images in figure 2.18. Table 2.6 shows for each model the calibrated values of the camera intrinsic parameters, the minimised objective function error  $\hat{\varepsilon}$ , and the mean and median reprojection error in the fisheye image for all ten input images (the reprojection errors in each image are combined into a single set before finding the mean and median). A box plot comparison of the reprojection errors for each camera model is presented in figure 2.25. Note that the ranking of the camera models using the calibration objective function error  $\hat{\varepsilon}$  corresponds to the same ranking using either the mean or median of the reprojection error measured on the fisheye image.

The calibration results suggest that the accuracy of the camera model defined by the minimised objective function error  $\hat{\varepsilon}$  in general improves with an increasing number of camera model intrinsic parameters. The stereographic, equiangular and equisolid models for example have only a single intrinsic parameter (excluding the position of the principal point  $\mathbf{u}_0$ ) and model the camera's radial distortion with the least accuracy. This observation is well supported in the literature and suggests that the 'ideal' pin-hole models described by Fleck [72] are not suited for real wide-angle fisheye cameras.

The two camera models of Mičušík [164, 162] include a parameter  $b$  which models deviations from the ideal equiangular model, and each of these converges to equiangular as  $b$  approaches zero. As expected, the addition of this term  $b$  gave improved accuracy over the ideal equiangular model. The combination model of Bakstein and Pajdla [9] models deviations from either the ideal stereographic or equisolid models. Again, as expected, this model gave improved accuracy over both the stereographic and equisolid models.

The three models found to most accurately model the fisheye camera in these ex-

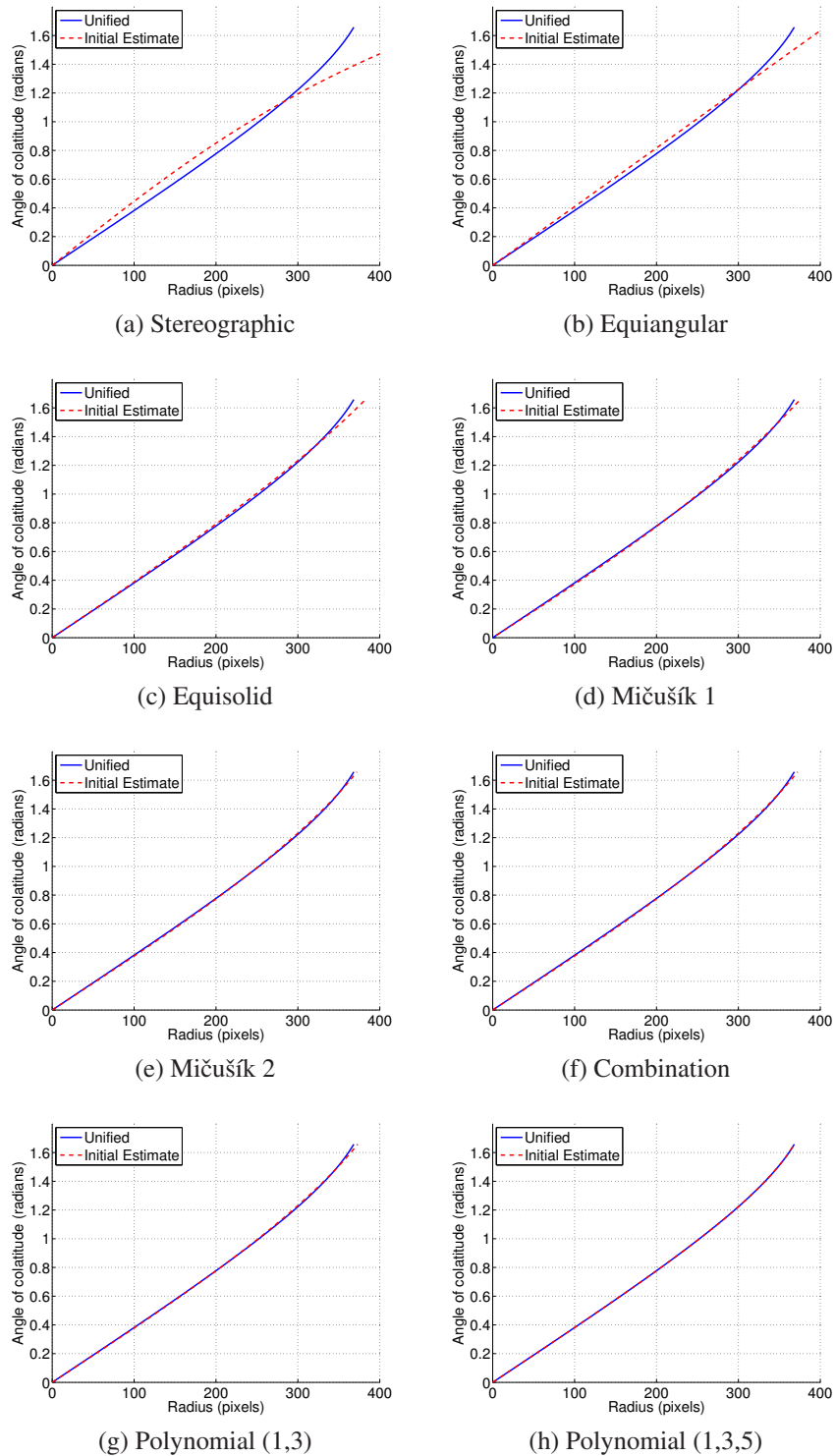


Figure 2.24: Initial estimates of the camera model functions obtained prior to calibration. These initial estimates for each model are found with respect to the calibration results for the unified image model in the example in section 2.4.6. Refer to table 2.4 for the specific camera model functions.

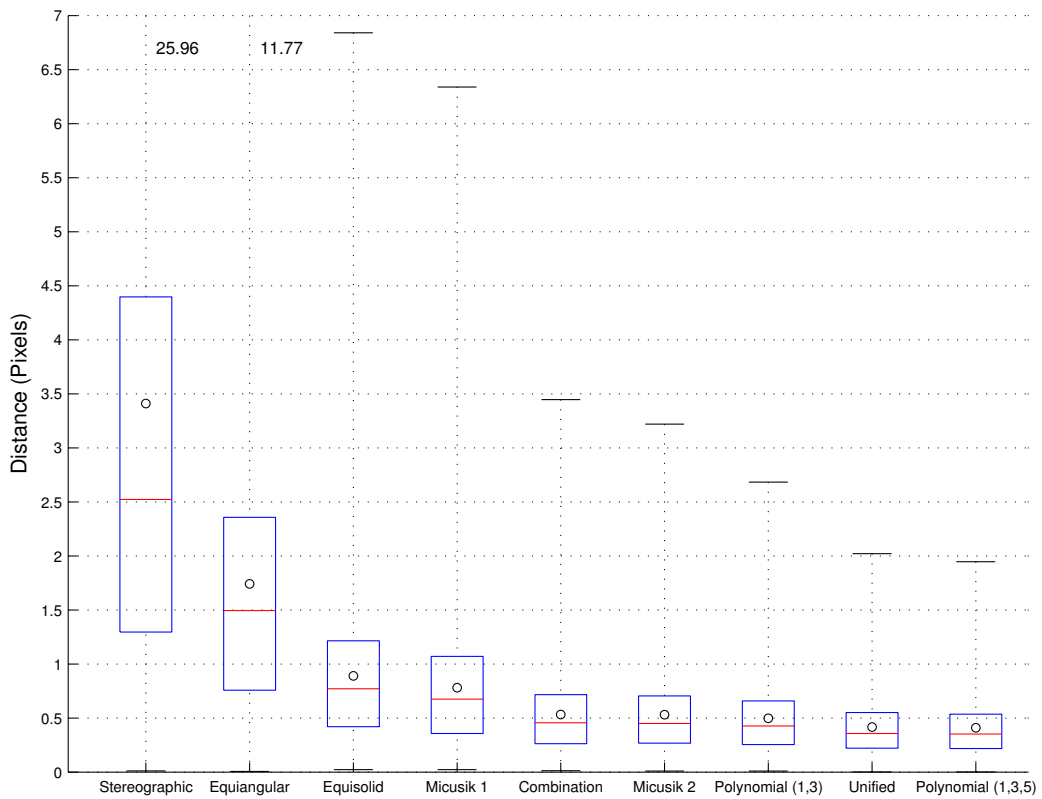


Figure 2.25: Boxplot of the calibration results for each ray-based fisheye camera model considered. The entries are sorted with respect to accuracy defined by the error  $\hat{\epsilon}$ . Notice that this ranking would be the same using either the mean or median reprojection distance errors measured in the fisheye images.

periments were the two polynomial models and the unified image model. This result supports arguments for the use of polynomial based models as they are capable of modelling a wide-range of cameras. The results indicate that there were improvements in accuracy using polynomial(1,3,5) over polynomial(1,3) which suggests that the addition of the higher order polynomial coefficient was able to better model the radial distortion. The polynomial(1,3,5) model outperformed all other models in these experiments followed closely by the unified image model. This validates the claim of Ying and Hu [244] that the unified model is able to accurately model image formation for many fisheye cameras.

When considering which model to use with the fisheye camera, one may simply argue that the model which the lowest error  $\hat{\epsilon}$  should be selected, which in this case is the polynomial(1,3,5) model. However, the inverse mapping from a radius on the image plane to the angle  $\theta$  requires the roots of the polynomial to be determined using iterative techniques. The unified image model in contrast can obtain a direct solution as seen in equation 2.10. Since there is only a very small improvement in accuracy

Model	Intrinsic values	Mean reprojection error (pixels)	Error ( $\hat{\epsilon}$ )
Stereographic	$u_0 = 521.2630$ $v_0 = 380.3094$ $k = 388.1975$	<i>median</i> = 2.5220 <i>mean</i> = 3.4097	0.60046
Equiangular	$u_0 = 532.1770$ $v_0 = 378.9425$ $k = 235.7712$	<i>median</i> = 1.4936 <i>mean</i> = 1.7401	0.15560
Equisolid	$u_0 = 530.0230$ $v_0 = 382.0045$ $k = 514.9977$	<i>median</i> = 0.7713 <i>mean</i> = 0.8894	0.04121
Mičušík 1	$u_0 = 528.5584$ $v_0 = 384.1524$ $a = 3.6088e^{-3}$ $b = -1.2540e^{-6}$	<i>median</i> = 0.6739 <i>mean</i> = 0.7805	0.03040
Bakstein	$u_0 = 528.4148$ $v_0 = 384.2398$ $a = 0.0037$ $b = 0.5143$ $c = 427.9583$ $d = 1.5663$	<i>median</i> = 0.4555 <i>mean</i> = 0.5320	0.01318
Mičušík 2	$u_0 = 528.2029$ $v_0 = 384.5021$ $a = 273.9097$ $b = -0.6453$	<i>median</i> = 0.4497 <i>mean</i> = 0.5312	0.01291
Polynomial (1,3)	$u_0 = 528.1757$ $v_0 = 384.4817$ $k_1 = 272.8297$ $k_2 = -17.7261$	<i>median</i> = 0.4269 <i>mean</i> = 0.4978	0.01108
Unified	$u_0 = 528.1214$ $v_0 = 384.0784$ $l = 2.7899$ $m = 996.4617$	<i>median</i> = 0.3568 <i>mean</i> = 0.4170	0.00673
Polynomial (1,3,5)	$u_0 = 528.1716$ $v_0 = 384.1241$ $k_1 = 264.9471$ $k_2 = -10.1055$ $k_3 = -1.9437$	<i>median</i> = 0.3515 <i>mean</i> = 0.4100	0.00665

Table 2.6: Summary of the calibration results for each ray-based fisheye camera model considered. The entries in the table are sorted with respect to accuracy defined by the error  $\hat{\epsilon}$ . Notice that this ranking would be the same using either the mean or median reprojection distances (error) defined in the fisheye image plane.

using the polynomial (1,3,5) model over the unified model, the unified model will be used for this fisheye camera for the remainder of this work.

## 2.5 Conclusions

A review of image formation with wide-angle central catadioptric and dioptric (fish-eye) cameras was presented in this chapter. This review was focused primarily on central projection cameras where all incoming rays intersect at a single effective view-point, and where the image can be back projected to a function on the unit view sphere. For central projection catadioptric cameras, the process of image formation is derived geometrically from the shape of the reflective surface (mirror). In contrast, the process of image formation for fisheye cameras cannot be easily derived from a geometric standpoint due to the complex nature of fisheye lens design. As a result, there are numerous empirical models used for describe image formation with fisheye cameras. These include pinhole models which define a mapping from the fisheye image to an undistorted perspective view, and ray-based model which define the mapping of scene

points to the fisheye image. The ray-based models were identified as being more flexible as they allow any pixel in the image to be back projected to a ray in space directly. Furthermore, an undistorted perspective image can be obtained for all ray-based models using a two step mapping via the sphere.

A review of some fundamental methods of camera calibration was then presented, including full-range, auto-calibration and plumb-line algorithms. A novel calibration algorithm was proposed based on plumb-line methods which uses the the position of vanishing points of coplanar sets of parallel lines in space for additional constraints. The calibration algorithm operates on the unit view sphere and was used to calibrate the fisheye camera used extensively in experiments throughout this work with a range of different ray-based camera models. From these results, the unified image model was selected as the ideal candidate and is used for the remainder of this work.



## Chapter 3

# Keypoint Detection, Description and Matching with Applications to Wide-Angle Images

*A review of keypoint detection, description and matching is presented in this chapter starting with the ‘classical’ methods suited for small-baseline motion. This follows with the wide-baseline methods which are suited for large change in camera pose and becoming increasingly popular for use in vision-based localisation applications. A review of the suitability of these wide-baseline methods is presented with respect to vision based-localisation applications, and the keypoint detection algorithms using scale-space analysis are identified as ideal candidates, in particular the Scale-Invariant Feature Transform (SIFT) of Lowe [142]. Both the classical and wide-baseline methods are designed primarily for use with perspective cameras and frequently applied ‘blindly’ to wide-angle images without accounting for the radial distortion. In many cases they are also applied directly to rectified log-polar and cylindrical panoramic images. The limitations of these approaches are discussed, and a potentially more ‘ideal’ approach is proposed based the suggestion of Daniilidis et al [56] that wide-angle image processing algorithms should be formulated as operations on the sphere. Re-formulating existing wide-baseline keypoint detection and description algorithms as operations on the sphere could therefore make them more suited for wide-angle image processing — this is the subject of chapter 4.*

## 3.1 Introduction

Camera egomotion can be estimated by observing the change in appearance of the environment between two images. As discussed in chapter 1, this can be measured quantitatively from the dense or sparse optical flow obtained from a set of keypoint correspondences between views. Many *classical* algorithms can be used for this purpose which are both designed and suited for small-baseline motion where there is a minimal change in appearance of the environment between views — the optical flow vectors may be only a few pixels in magnitude.

The ability to find correspondences across a wide-baselines change in camera pose has potential advantages for vision odometry and visual place recognition applications. However, this is a challenging task due to the large projective changes in the appearance of the environment between images. These include changes in rotation, scale, affine transformations (for planar objects) as well as large illumination variations. Fortunately there is an impressive body of literature which addresses specifically this problem, and there are a number of algorithms capable of detecting and describing keypoints in a manner invariant to rotation, scale and in some cases affine transformations. These include a number based on a scale-space framework [142, 64, 158, 160, 159, 17], using a family of scale-space images obtained via convolution of an image with sampled Gaussian functions of increasing scales [123, 134]. A number of alternative approaches have also been proposed by Kadir and Brady [112, 115, 113, 114], Tuytelaars and Van Gool [230], and Matas et al [151].

Wide-baseline keypoint detection algorithms are designed almost exclusively for use with perspective cameras and frequently use operators that are shift-invariant in the image plane. This means that if an image were to be smoothed for example, then the image would be convolved with a fixed shape smoothing kernel (e.g. a sampled Gaussian kernel of fixed size and scale). In many cases these algorithms are applied ‘blindly’ to wide-angle images without accounting for the radial distortion. However, the appearance of objects change considerably depending on their position in a wide-angle images due to the radial distortion of the camera. Applying operators that are shift-invariant in the image plane to wide-angle images is therefore not ideal. For a calibrated central projection wide-angle camera one could always convert it to a perspective image and process this perspective image using existing algorithms designed for perspective images (i.e. shift-invariant in the image plane). As noted by Daniilidis et al [56] the problem with this approach twofold. First, perspective projective is limited to less than a hemispherical field of view. Second, the interpolation required for



this mapping is computationally expensive and introduces artifacts. Another approach which has been used is to process the image using shift-invariant operators on a rectified log-polar or cylindrical panoramic image. This is again not ideal as the image is not perspective. Daniilidis et al [56] proposed that the ideal domain in which to formulate shift-invariant image processing algorithms is the sphere, where shift refers to a rotation. As any central projection wide-angle camera can be effectively mapped to the sphere, this approach is suited for all and is invariant to the radial distortion in an image. Furthermore, they discuss how these image processing algorithms formulated on the sphere can be implemented on the wide-angle image itself avoiding the need for interpolation of the original image function. Although they used this approach to find small-baseline optical flow, it is proposed that this general approach to image processing can be used for wide-baseline keypoint detection, description and matching with wide-angle images. More specifically, existing wide-baseline algorithms including those based on the scale-space framework can be reformulated as image processing algorithms on the sphere. This is made possible by the work of Bülow who derived the scale-space for functions on the sphere [31].

This chapter commences with a review of some popular classical methods used to find both dense and sparse optical flow. The limitations of these methods with respect to wide-baseline change in camera pose between views is then discussed. This follows with a review keypoint detection and description methods used to find correspondences across images separated by wide-baseline (large change in camera pose). This includes those based on the scale-space framework and a number of alternative approaches. A review of a number of comparative works is presented where it is concluded that the methods using scale-space analysis are well suited for applications to vision-based localisation. A discussion on the limitations of applying these blindly to wide-angle images, rectified perspective and various panoramic (cylindrical and log-polar) images is discussed. The ‘ideal’ approach proposed by Daniilidis et al [56] is discussed where image processing is formulated as an operation on the unit sphere. It is proposed that existing wide-baseline algorithms could be adapted to follow this approach. This paves the way for chapter 4 where an existing wide-baseline keypoint detection and description algorithm, the Scale-Invariant Feature Transform [142], is formulated as an image processing operation on the sphere and used for wide-baseline keypoint detection and matching with wide-angle images.

## 3.2 Classical (small-baseline) Techniques

### 3.2.1 Sparse Optical Flow

Camera egomotion can be estimated from the sparse optical flow, which is the change in position of a select group of keypoints between two images. There are two fundamental methods used to find the sparse optical flow:

1. Keypoint registration: Keypoints are detected in the first image. For each keypoint the image function in the local region surrounding it is selected as a template. The position in the second image where this template is most similar to the image is used to find the estimate of the keypoint location in the second image. Cross-correlation methods are popular for assessing this similarity.
2. Keypoint detection and matching: Keypoints are found independently in both images. A image function in the local region surrounding each keypoint is taken again as a template for matching. A similarity between a keypoint in the first image and all those in the second image is found, the most similar taken to be the corresponding keypoint (assuming the similarity is above some predefined threshold). This similarity can be found using directly the greyscale intensity values in the template and cross-correlation for example, or by producing a descriptor for each keypoint, which is a distinctive representation of the local image content in the template.

For both cases it is necessary to detect keypoints in an image, where a keypoint is defined as any salient point and often referred to as an interest point, feature or corner. With reference to figure 3.1, the general procedure for many keypoint detection algorithms is as follows. Take an input image, calculate the saliency of each pixel, then select keypoints as local extrema of the saliency function with values above some threshold. In many cases subpixel accuracy is achieved by interpolating the saliency values in the local region surrounding a keypoint. The algorithms used to detect keypoints differ primarily in their definition of saliency.

#### 3.2.1.1 Keypoint Detection

This section reviews a number of popular classical keypoint detectors. Although each differs in their definition of pixel saliency, a common theme for all is to assign a higher saliency to corner points in an image. Corner points have a large change in intensity

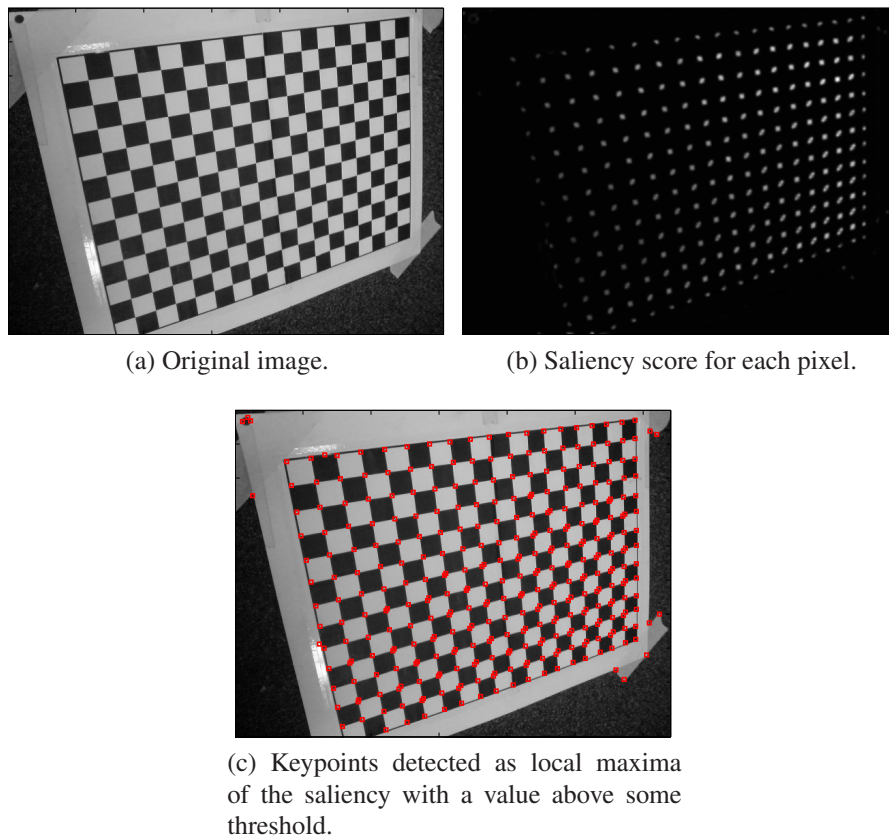


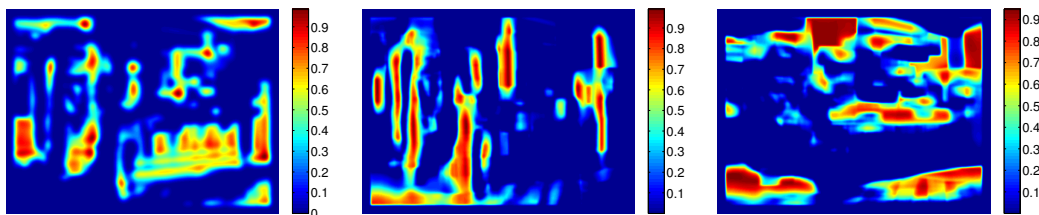
Figure 3.1: Example keypoint detection using the Harris/Plessey keypoint detection algorithm [94] discussed in section 3.2.1.1. The algorithm finds for each pixel a saliency measure (lighter values indicate higher saliency). A keypoints is as local maxima of the saliency function with a value above some predefined threshold. Notice that homogeneous regions and points along lines have small saliency values. Most algorithms are designed to find ‘corners’ such as the intersections of the checkerboard pattern used in this example.

values in orthogonal directions and are more distinctive than points in the image which lie on edges or homogeneous regions. This can be illustrated with respect to figure 3.2. The top image in figure 3.2a shows a point on a corner, edge, and homogeneous region. The square around each indicates a small local region (template). The results of zero normalised cross-correlation (see section 3.2.1.2) of each template is shown in figure 3.2b. Observe that the corner point is the most distinctive as it has only a high similarity at a small number of points in the second image. In contrast, the edge point has a high similarity along all vertical edges in the second image and the homogeneous region at many locations.

There is a large number of keypoint detection algorithms, and an exhaustive review will not be given here. Intensity based methods are most frequently used which operate directly on the greyscale image (i.e. the example in figure 3.1). However, there are



(a) Two images separated by a change in viewpoint. Image 1 (left) shows a corner point (blue), edge point (red) and homogeneous point (green). The square region surrounding each is used for the cross-correlation



(b) The results of zero normalised cross-correlation of the templates with image two for the corner point (left), edge point (middle) and homogeneous point (right).

Figure 3.2: Zero normalised cross correlation of a corner, edge and homogeneous point. Observe that the homogeneous point attains high similarity in many regions in the image, the edge point on many vertical edges, and the corner point at only a few regions. Corner points are preferred as they are generally more distinctive than edge and homogeneous points when finding correspondences via registration or matching.

a range of alternatives such as contour based methods. An example is the Curvature Scale-Space (CSS) algorithm of Mokhtarian and Suomela [165] which uses edge detection to find contours (isophotes) in the image from which the keypoints are found — the keypoint detector effectively operates on the contour image and not the original greyscale values. A review of some of these alternatives is given by Schmid et al [203].

One of the earliest keypoint detection algorithms was developed by Moravec [168]. Moravec proposed that for an image function  $I$ , a pixel at position  $u, v$  with image value  $I(u, v)$  is distinct if the minimum variance of the autocorrelation function centred about the pixel is above some threshold. The autocorrelation function  $f(u, v)$ , which is the cross correlation of the image function with itself, is

$$f(u, v) = \sum_{\mathbf{w}} [I(u_i, v_i) - I(u_i + \Delta u, v_i + \Delta v)]^2, \quad (3.1)$$

where  $u_i, v_i$  are the coordinates of a pixel within a window function  $\mathbf{W}$ , and  $\Delta u, \Delta v$  are discrete pixel shifts in the  $u$  and  $v$  directions respectively. The saliency of a pixel is defined as the minimum of the autocorrelation function shifted in eight cardinal direc-

tions (vertical, horizontal and diagonal) using a fixed sized binary window function  $\mathbf{W}$  in the order of 4 to 8 pixels half width. Keypoints are selected as pixels whose saliency value is a local extrema above some threshold.

Harris and Stephens [94] note that Maoravec's approach to keypoint detection is anisotropic as the autocorrelation function at a pixel is evaluated in only eight directions — simply rotation the image by some angle could alter considerably the keypoints found. A point on an edge for example may be assigned a high saliency value unless the autocorrelation function were assessed in the direction of the edge. They proposed an alternate keypoint detection algorithm which attempts to find an isotropic saliency metric. This method is frequently referred to as either the Plessey or Harris corner detector, the later being most frequently used in the literature and used for the remaining discussions. The key to their method is the use of the autocorrelation matrix  $A$  from which the approximate autocorrelation function can be evaluated in any arbitrary direction. This autocorrelation matrix, also referred to as the second moment matrix, has since been used as the basis for numerous keypoint detection algorithms [94, 76, 77, 209]. The autocorrelation matrix is derived based on the assumption that the image function  $I(u_i + \Delta u, v_i + \Delta v)$  at some small shift  $\Delta u, \Delta v$  in the  $u, v$  directions respectively from point  $I(u_i, v_i)$  can be approximated by the first order Taylor expansion about the point  $u_i, v_i$ :

$$I(u_i + \Delta u, v_i + \Delta v) \approx I(u_i, v_i) + (I_u(u_i, v_i) I_v(u_i, v_i)) \begin{pmatrix} \Delta u \\ \Delta v \end{pmatrix}, \quad (3.2)$$

where  $I_u(u_i, v_i)$  and  $I_v(u_i, v_i)$  are the first order derivatives in the  $u$  and  $v$  directions respectively evaluated at the pixel position  $u_i, v_i$ . Substituting into equation 3.1 gives

$$f(u, v) \approx \sum_{\mathbf{W}} \left[ I(u_i, v_i) - I(u_i, v_i) - (I_u(u_i, v_i) I_v(u_i, v_i)) \begin{pmatrix} \Delta u \\ \Delta v \end{pmatrix} \right]^2, \quad (3.3)$$

where  $\mathbf{W}$  is some window function. After expanding, equation 3.3 can be written as

$$f(u, v) \approx (\Delta u \Delta v) \begin{pmatrix} \sum_{\mathbf{W}} I_u(u_i, v_i)^2 & \sum_{\mathbf{W}} I_u(u_i, v_i) I_v(u_i, v_i) \\ \sum_{\mathbf{W}} I_u(u_i, v_i) I_v(u_i, v_i) & \sum_{\mathbf{W}} I_v(u_i, v_i)^2 \end{pmatrix} \begin{pmatrix} \Delta u \\ \Delta v \end{pmatrix}, \quad (3.4)$$

$$f(u, v) \approx (\Delta u \Delta v) A \begin{pmatrix} \Delta u \\ \Delta v \end{pmatrix}. \quad (3.5)$$

Harris and Stephens use a Gaussian window function

$$W = G(u, v; \sigma) = \exp(-(u^2 + v^2)/2\sigma^2) \quad (3.6)$$

and compute the first order derivatives as

$$I_u = \frac{\partial I}{\partial u} = I * (-1, 0, 1) \quad I_v = \frac{\partial I}{\partial v} = I * (-1, 0, 1)^T \quad (3.7)$$

where  $*$  is the convolution operator. The eigenvalues  $\lambda_1, \lambda_2$  of the autocorrelation matrix give the principal variance of the autocorrelation function in orthogonal directions, where the direction of each is defined by the corresponding eigenvectors. If the values of both eigenvalues are large, then a corner point or highly textured region has been found [209]. If only one is large, then there is a large variance in only one direction which indicates an edge response. If both are small, the local image function in the neighbourhood of the pixel is approximately homogeneous. The Harris detector defines the saliency of a pixel based on the eigenvalues of the autocorrelation matrix. Rather than compute the eigenvalues explicitly, the saliency, or ‘cornerness’,  $C$  of a pixel is evaluated as

$$C = \det(A) - k \text{trace}(A)^2 \quad (3.8)$$

where  $\det(A) = \lambda_1 \lambda_2$  is the determinant of  $A$ ,  $\text{trace}(A) = \lambda_1 + \lambda_2$  is the trace of  $A$ , and  $k$  is an empirical constant (0.04 for greyscale intensity values in the range 0-255). A high value of  $C$  is a corner response, negative value a edge, and a small value a pixel lying in a homogeneous region. A pixel whose saliency value  $C$  is above some threshold, and whose value is greater than that of its 8 neighbouring pixels, is selected as a keypoint.

Although the Harris corner detector has been used extensively for many computer vision applications, Schmid et al [203] suggest that there is still some anisotropic bias in the computation of the partial derivatives in equation 3.7. They propose an alternative *improved Harris* detector which computes the partial derivatives via convolution with derivative of Gaussian kernels  $G(\sigma)$  of standard deviation  $\sigma$ :

$$I_u = \frac{\partial I}{\partial u} = I * \frac{\partial G(\sigma)}{\partial u} \quad I_v = \frac{\partial I}{\partial v} = I * \frac{\partial G(\sigma)}{\partial v}. \quad (3.9)$$

The autocorrelation matrix has also been used for keypoint detection in a method first outlined by Tomasi and Kanade in [223] and formalised by Shi and Tomasi in [209]. This method is frequently referred to as either the Shi-Tomasi keypoint detector, or the KLT keypoint detector named after the authors of a series of works relating to methods



of optical flow estimation [209][223] — for this discussion the former is used. The Shi-Tomasi detector finds the eigenvalues of the autocorrelation matrix and uses as a saliency metric  $C$  the value of the minimum eigenvalue:

$$C = \min(\lambda_1, \lambda_2). \quad (3.10)$$

A pixel is selected as a keypoint if  $C > \lambda$ , where  $\lambda$  is a predefined threshold. Although the Shi-Tomasi keypoint detector does not distinguish between corner and edge responses based on the ratio of eigenvalues, the authors argue that any pixel with a minimum eigenvalue above some threshold is salient, being either a corner or highly textured local region which may be reliably tracked. In some respects the Shi-Tomasi algorithm can be considered an isotropic version of Morevec's algorithm as only the minimum eigenvalue of the autocorrelation matrix (the approximate minimum variance of the autocorrelation function) is used to define saliency.

Another early keypoint detector was proposed by Beaudet [195]. Rather than use the autocorrelation matrix as the basis for keypoint detection, the Hessian matrix  $\mathcal{H}$  was used, which is the square matrix of second order partial derivatives computed from the local image function surrounding each pixel:

$$\mathcal{H} = \begin{bmatrix} \frac{\partial^2 I}{\partial u^2} & \frac{\partial^2 I}{\partial u \partial v} \\ \frac{\partial^2 I}{\partial u \partial v} & \frac{\partial^2 I}{\partial v^2} \end{bmatrix} = \begin{bmatrix} I_{uu} & I_{uv} \\ I_{uv} & I_{vv} \end{bmatrix}. \quad (3.11)$$

Beaudet's method was termed DET, an isotropic measure of saliency defined as the determinant of the Hessian matrix

$$DET = \det(\mathcal{H}) = I_{uu}I_{vv} - I_{uv}^2 = \lambda_1\lambda_2 \quad (3.12)$$

where  $\lambda_1$  and  $\lambda_2$ , the eigenvalues of the Hessian matrix, are the principal curvatures of the image function (greyscale intensity values) in orthogonal directions. This metric can be interpreted geometrically as the Gaussian curvature of the image function centred about a pixel, where the Gaussian curvature is proportional to the product of the eigenvalues<sup>1</sup>. If DET is positive and large then the pixel is a corner point; local minima if both eigenvalues are positive and local maxima if both eigenvalues are negative. If DET is negative, then the local curvature is a saddle point. If DET is close to zero, the local region surrounding the pixel is either a homogeneous region with near constant image values or an edge response. Pixels are selected as keypoints if the DET value

<sup>1</sup>If  $\lambda_1$  and  $\lambda_2$  are the eigenvalues of the Hessian matrix, the Gaussian curvature is frequently defined as the product  $\lambda_1\lambda_2$ . As noted by Derich [60], the Gaussian curvature is often defined as  $\lambda_1\lambda_2/(\lambda_1^2 + \lambda_2^2)$ . Irrespective of the choice, the sign of the DET would not change as the denominator is always positive.

computed at the pixel is above some threshold.

Both first and second order partial derivatives are also been used in the early keypoint detector of Kitchen and Rosenfeld [122]. The Kitchen-Rosenfeld detector defines the saliency of each pixel as the rate of change of curvature of a contour line (isophote) passing through the pixel. This rate of change of curvature is the change of gradient direction multiplied by the local gradient magnitude and sometimes referred to as the level curve curvature [132, 134]. They derived the analytical expression for this curvature which is used to define the saliency  $C$  of a pixel as

$$C = \frac{I_{uu}I_v^2 + I_{vv}I_u^2 - 2I_{uv}I_uI_v}{I_u^2 + I_v^2}. \quad (3.13)$$

The first and second order partial derivatives computed at each pixel are obtained using the facet model computed from a second order polynomial fitted to the image function in the local 3, 5 or 7 pixel square region surrounding the pixel. Non-maxima suppression of the gradient magnitudes is applied before multiplication (computing  $C$ ). Those pixels with a saliency value  $C$  above some threshold are selected as keypoints. As discussed by Schmid et al [203], an identical saliency metric  $C$  in equation 3.13 was derived for keypoint detection in a separate work by Dreschler and Nagel [62].

The use of image derivatives is prone to errors due to noise, particularly for the keypoint detectors of Beaudet and Kitchen-Rosenfeld which use second order partial derivatives. This can result in false positive keypoint detection which limits the ability to find correct and reliable keypoint correspondences between views. Smith and Brady [171] proposed a novel keypoint detector termed ‘SUSAN’ which operates without using derivatives making the method less sensitive to image noise — SUSAN is an acronym for Smallest Univalued Segment Assimilating Nucleus. A circular mask is taken about each pixel, the centre pixel being the ‘nucleus’ of the mask. The USAN for the mask is then found which contains all pixels of similar greyscale intensity as the nucleus. The USAN contains important properties used for keypoint detection. If the USAN is large, the pixel lies within a near homogeneous region. If it is approximately half the size of the mask, the pixel is on an edge. If it is small, the pixel lies on a corner. The saliency of each pixel is defined by the size of the USAN, where keypoints are found by finding the Smallest USAN’s.

Although an exhaustive review of classical keypoint detectors has not been presented here, the Harris corner detector has ranked consistently well in comparative works and used successfully in visual odometry applications [181, 46]. Tissainayagam and Suter [222] compared the performance of the Kitchen-Rosenfeld, Plessey, Shi-



Tomasi and Smith (SUSAN) keypoint detectors with respect to keypoint stability and localisation accuracy using indoor and outdoor scenes. Keypoint stability is the ability to detect the same keypoint in different images, and localisation accuracy is a measure of a keypoint's pixel position found relative to that of its estimated true position. Using both static and moving image sequences with and without added Gaussian noise (and illumination variations for the outdoor scene resulting from change in intensity and viewpoint of the natural light source), the Harris and Shi-Tomasi keypoint detectors performed best with respect to both stability and localisation accuracy.

Another comparison of keypoint detectors was made by Schmid, Mohr and Bauckage [203, 204], comparing the Harris keypoint detector and their modified version discussed previously to a number of other classical methods not discussed here — a summary of each can be found in their work. The comparison was based on two criteria, and these were the repeatability of the keypoints and the image content of the local regions surrounding the keypoints. Repeatability was measured as the percentage of keypoints found between two images, and information content by the entropy of the greyscale intensity function in the local region surrounding each keypoint. Keypoints with high information content (entropy) are desirable as the local region is in general very distinctive permitting accurate keypoint registration or matching between views. Results were found for changes in rotation, scale, illumination variations, viewpoint change and camera (image) noise. Their improved Harris detector discussed previously was found to perform better than the original and shown to give improved or comparable performance to the others with respect to both repeatability and image content.

To summarise, the Harris keypoint detector remains one of the most robust and popular keypoint detectors for small-baseline applications. As will be seen in later discussions in section 3.3.1, the principles it uses for keypoint detection have since been adapted for scale-invariant keypoint detection. Hessian based methods such as that of Beaudet have also since been adapted for scale-invariant keypoint detection.

### 3.2.1.2 Finding Keypoint Correspondences: Registration and Matching

As discussed, there are two methods that can be used to find sparse optical flow. The first is registration, detecting keypoints in the first image and searching in the other image for their most likely position based on the similarity of local regions. The second is matching, detecting keypoints in both images separately and then finding the most similar keypoint in the other image.

Area based methods are typically used to measure similarity for registration, and as noted by Banks [10] have a number of advantages compared to other methods used to describe local image content. They are simple to implement, fast and suited to textured regions. Area based methods use the image pixel values (typically greyscale) in the template to obtain a similarity measure. Let  $T_1$  be the template of a keypoint in the first image with local coordinates  $x, y$ , and  $T_2$  be the second image with pixel coordinates  $u, v$  — the template  $T_1$  is the local image function surrounding a keypoint. One of the simplest similarity measures is the sum of squared distances  $SSD$  which is computed at any pixel location  $u, v$  in the second image as

$$SSD(u, v) = \sum_{x,y} (T_1(x, y) - T_2(u + x, v + y))^2, \quad (3.14)$$

where  $\sum_{x,y}$  is the summation over all pixels in the template. From the inner product of  $SSD$  the general expression for cross-correlation  $CC$  is derived:

$$CC(u, v) = \sum_{x,y} T_1(x, y) T_2(u + x, v + y). \quad (3.15)$$

A limitation of general cross correlation is a lack of invariance to changes in intensity offset and scale across images resulting from lighting variations and viewpoint change, a problem most evident in outdoor environments. Normalised cross correlation  $NCC$  attempts to provide illumination scale invariance by dividing with the variance in the templates being matched to give

$$NCC(u, v) = \frac{\sum_{x,y} T_1(x, y) T_2(u + x, v + y)}{\sqrt{\sum_{x,y} T_1(x, y)^2 \sum_{x,y} T_2(u + x, v + y)^2}}. \quad (3.16)$$

A further improvement which accounts for effects of illumination variation, illumination scale change and offset is zero mean normalised cross correlation  $ZNCC$ . Let  $\bar{T}_1$  be the mean intensity value in the template  $T_1$ , and  $\bar{T}_2'$  be the mean intensity value in the local template  $T_2'$  — this local template is simply the region in the second image spanned by the template  $T_1$  when assessing similarity.  $ZNCC$  subtracts from each template the mean intensity values and divides by the normalised variance of the templates to give

$$ZNCC(u, v) = \frac{\sum_{x,y} (T_1(x, y) - \bar{T}_1) (T_2(u + x, v + y) - \bar{T}_2')}{\sqrt{\sum_{x,y} (T_1(x, y) - \bar{T}_1)^2 \sum_{x,y} (T_2(u + x, v + y) - \bar{T}_2')^2}}. \quad (3.17)$$

Although the methods of cross correlation described use pixel intensity values, these

methods are applicable to any information contained within the local region templates. Intensity gradients or colour information for example could be used. Furthermore, in many practical applications with small change in appearance between images it is not necessary for a keypoint to obtain a similarity metric for all pixels in the second image. A more computationally efficient approach is to obtain a similarity measure around the estimated position of the keypoint in the second image. Jung and Lacroix [111] for example predict the position of each keypoint in the second image given the known 3D position obtained with stereo vision. They obtain the *ZNCC* similarity measure using a  $9 \times 9$  template within a  $41 \times 41$  region centred around the predicted position. They note that this reduces both the computational expense of registration and potential of finding an incorrect estimate of the keypoints position. If the expected change is within only a few pixels then the predicted position could simply be set as the keypoint position in the first image. If subpixel accuracy is required in the estimate of the final keypoint position the similarity function can be interpolated [111].

The second approach used to find the sparse optical flow is keypoint detection and matching which requires a means for assessing similarity between two keypoints. Methods of cross-correlation can again be used for this purpose, the cross-correlation being between the templates for each keypoint taken as a fixed sized local region around each. For example, keypoint matching using Harris corners and *ZNCC* has been used by Corke [46] and Jung and Lacroix [110], and Harris corners and *NCC* used by Nistér [181]. An alternate approach for assessing similarity is to produce first for each keypoint a descriptor, which is a vector that encodes the local image content in the template surrounding the keypoint. The similarity is then found by comparing descriptors. One advantage of using descriptors is a greater invariance to viewpoint change between images when assessing similarity. In particular, many descriptors are invariant to the orientation of the template. A more detailed discussing on keypoint descriptors is reserved for section 3.3.3, however, it is important to recall again that the size of the template is set as some fixed size for all keypoints when using classical (small-baseline) methods.

### 3.2.2 Dense Optical Flow

Dense optical flow finds for each pixel in the first image an estimate of its position in the second (assuming it is within the image). This can be achieved by registration of every pixel in the first image for example using methods of cross-correlation. A more elegant method of registration was proposed by Lucas and Kanade [144] using greyscale intensity gradients. Their method is based on the assumption that the local

image content surrounding the same scene point in two images are related by a translation, the magnitude and direction of this translation being the optical flow vector for the scene point.

Let  $I$  and  $J$  be images 1 and 2 respectively with greyscale intensity values  $I(\mathbf{u})$  and  $J(\mathbf{u})$  at coordinate  $\mathbf{u} = (u, v)^T$ . If both images were related by a linear translation  $\mathbf{h} = (h_u, h_v)^T$ , then  $I(\mathbf{u})$  would be equal to  $J(\mathbf{u} + \mathbf{h})$ . Lucas and Kanade [144] define the error  $\varepsilon$  for the estimate of the translation  $\mathbf{h}$  as the L2 norm

$$\varepsilon = \sum_W [J(\mathbf{u} + \mathbf{h}) - I(\mathbf{u})]^2, \quad (3.18)$$

where  $W$  is a local region surrounding the pixel location  $\mathbf{u}$  in image 1 and  $\mathbf{u} + \mathbf{h}$  in image 2. They assume that the local image function  $J(\mathbf{u})$  in the region surrounding  $\mathbf{u}$  is linear, where  $J(\mathbf{u} + \mathbf{h})$  can be approximated from the first order Taylor series expansion of  $J$  about  $\mathbf{u}$ :

$$J(\mathbf{u} + \mathbf{h}) \approx J(\mathbf{u}) + \mathbf{h}^T \frac{\partial J}{\partial \mathbf{u}}(\mathbf{u}) \quad \text{where} \quad \frac{\partial J}{\partial \mathbf{u}}(\mathbf{u}) = \left( \frac{\partial J}{\partial u}(\mathbf{u}), \frac{\partial J}{\partial v}(\mathbf{u}) \right)^T. \quad (3.19)$$

An estimate for  $\mathbf{h}$  is obtained by substituting 3.19 in 3.18 and setting the derivative to zero:

$$0 = \frac{\partial \varepsilon}{\partial \mathbf{h}} \approx \frac{\partial}{\partial \mathbf{h}} \sum_W \left[ J(\mathbf{u}) + \mathbf{h}^T \frac{\partial J}{\partial \mathbf{u}}(\mathbf{u}) - I(\mathbf{u}) \right]^2 \quad (3.20)$$

$$= \sum_w 2 \frac{\partial J}{\partial \mathbf{u}}(\mathbf{u}) \left[ J(\mathbf{u}) + h \frac{\partial J}{\partial \mathbf{u}}(\mathbf{u}) - I(\mathbf{u}) \right] \quad (3.21)$$

$$(3.22)$$

from which

$$\mathbf{h} = \frac{\sum_W \left( \frac{\partial J}{\partial \mathbf{u}}(\mathbf{u}) \right)^T [I(\mathbf{u}) - J(\mathbf{u})] w(\mathbf{u})}{\sum_W \left( \frac{\partial J}{\partial \mathbf{u}}(\mathbf{u}) \right)^T \left( \frac{\partial J}{\partial \mathbf{u}}(\mathbf{u}) \right) w(\mathbf{u})}. \quad (3.23)$$

The value  $w(\mathbf{u})$  is a weighting factor evaluated at each point  $\mathbf{u}$  in the window defined in one dimension as  $w(u) = (|J'(u) - I'(u)|)^{-1}$ . An iterative scheme is used to converge on a solution for  $\mathbf{h}$ .

The problem with computing reliable dense optical flow was illustrated previously in figure 3.2. Points within homogeneous regions in the image are similar to many regions in the other image within a close proximity. As a result a reliable estimate for  $\mathbf{h}$  is difficult to obtain. For points on an edge a reliable estimate for  $\mathbf{h}$  can be found

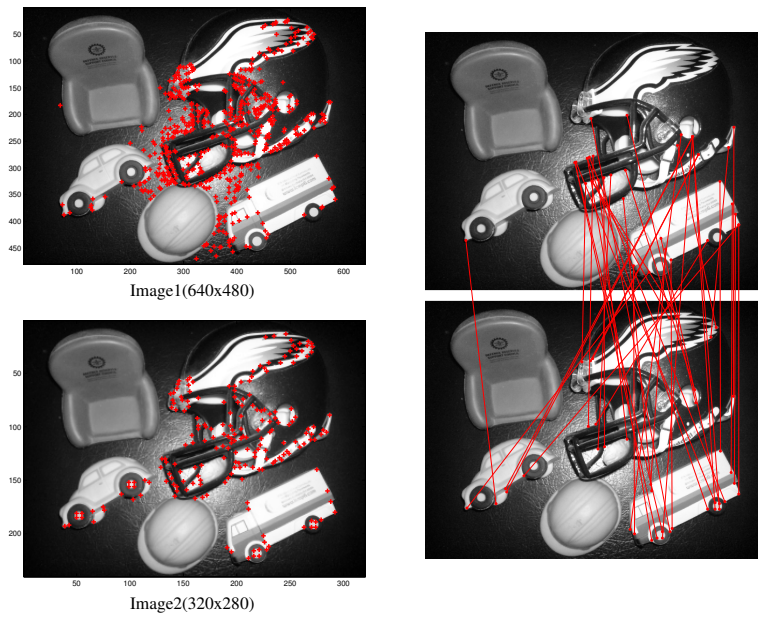
only in the direction orthogonal to the edge. For this reason, the Shi-Tomasi keypoint detector has been used to select only a small set of salient points to track. This forms the so called Kanade-Lucas-Tomasi (KLT) keypoint tracking algorithm [144, 209, 223] which finds sparse optical flow. KLT is a robust method for obtaining sparse optical flow and can be used to successfully *track* keypoints over multiple frames. Although the authors suggest several means for including more generalised projective changes than a simple translation [144], KLT is most suited for small viewpoint changes between images.

### 3.2.3 Limitations for Wide-baseline Motion

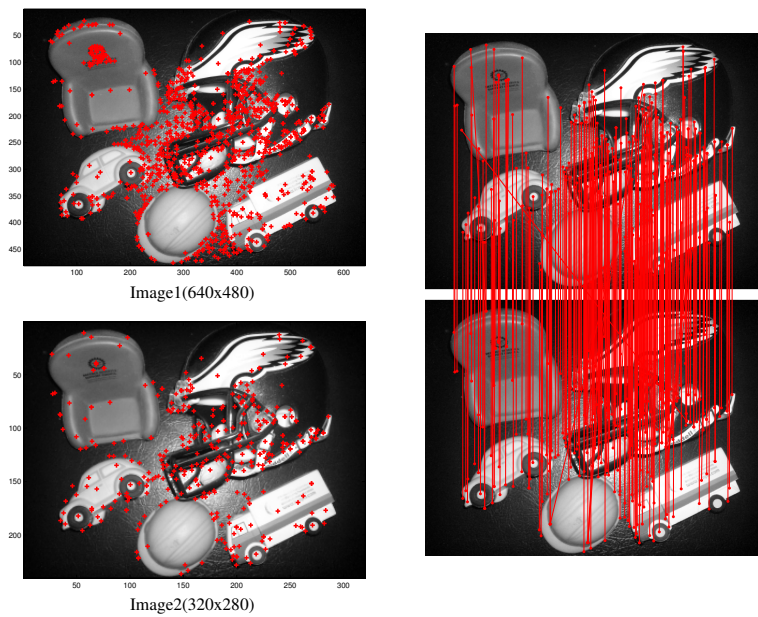
The classical methods used to detect keypoints and find optical flow via registration or matching are suited for small-baseline motion where there is minimal change in the appearance of the scene between views. For wide-baseline motion there can be considerable projective changes between views, including rotation, affine transforms (for planar objects) and scale change where the apparent size of regions in the scene change in the image.

Scale change in particular becomes problematic when attempting to find reliable optical flow between views using classical algorithms. This occurs as classical methods use fixed sized operators to find keypoints, and fixed sized support regions (templates) for keypoint registration, description and matching. To illustrate, figure 3.3a shows the result of keypoint detection and matching using the Harris corner detector and ZNCC matching (threshold 0.7) between local  $31 \times 31$  templates surrounding each keypoint for two images. For this example the second image is a half resolution linear interpolation of the first (scale change factor of 2). Notice that only a minimal number of correct correspondences are found. Even if the same keypoints were found in the two images, the size of the template (keypoint support region) surrounding each keypoint is the same size in both images. As no account is made for the change in resolution, the templates for the same keypoint in image 1 and 2 can span different regions of the scene. Figure 3.3b shows the result using the Scale-Invariant Feature Transform (SIFT) of Lowe [142], which is a wide-baseline algorithm including a method of keypoint detection and description, and will be discussed in section 3.3. Observe that SIFT is capable of finding many correct correspondences irrespective of the change in image resolution.

One could argue that keypoint tracking could be used to increase the change in pose between views used for egomotion estimation, that is, rather than find correspondences



(a) Keypoint detection and matching using Harris corners and ZNCC (31 x 31 pixel template).



(b) Keypoint detection and matching using SIFT.

Figure 3.3: Keypoint detection and matching results using Harris corners and ZNCC (a) and SIFT (b). The leftmost column shows the keypoints detected in the images, and the rightmost column shows the set of keypoint correspondences. Only the position of the SIFT keypoints is shown — each keypoint has a unique ‘characteristic’ scale which defines the size of its support region.



over a wide-baseline the keypoints are tracked over multiple small-baselines before computing camera egomotion. For methods using keypoint detection and matching in particular, the ability to track keypoints requires that the same keypoint are found in all images which is difficult. Although tracking may be applicable for visual odometry applications, the ability to find keypoint correspondences across wide-baselines is required for visual place recognition. Consider a camera returning to previously visited location. Unless the camera returns to almost exactly the same location and pose, there will be significant projective differences between the images taken by the camera at each of these locations. As discussed, the classical methods of keypoint detection and matching are not suited for this scenario. Most algorithms used for visual place recognition use methods of keypoint detection and description that are suited for finding correspondences between images separated by wide-baseline changes in pose [50, 51, 52].

### 3.3 Wide-Baseline Techniques

The key to finding corresponding keypoints between two images separated by a wide-baseline is the ability to detect and describe the same keypoints in a way that is invariant to the apparent change of appearance of the scene between the images. As discussed scale change is one of the greatest challenges.

It is beneficial to discuss here the concept of *scale* described by Koenderink [123] using as an example a scene containing a group of buildings. At a fine scale the features in the scene may be the corners of windows and buildings, at a higher scale the windows and doors themselves, and at an ever higher scale the buildings. With respect to an image of the scene, the ‘outer scale’ is the highest scale at which features can be found and is dependent on the cameras field of view. The ‘inner scale’ is dependent on the image resolution and dictates the features observable in the image. Consider then two images of the scene, one taken close to the buildings and one far away. As the Harris corner detector uses fixed sized image operators to compute the greyscale gradients, it operates on only one level of resolution. It may therefore only be able to find keypoints on the corners of windows and doors in the first image and only the entire windows and doors in the second. This limits the ability to find keypoint correspondences between the images. Koenderink [123] considers that unless one knows which features need to be found, to retain all relevant information an image needs to be considered simultaneously on all levels of resolution — applying the Harris corner detector to a blurred version of the first image for example would potentially allow the

doors and windows to be detected. The success of wide-baseline methods is primarily their ability to find features (keypoints) in the image over a range of scales, that is, at different levels of resolution. These are often referred to as *scale-invariant* keypoint detectors. As will be seen, they are able to also set the keypoint support region used for description in a scale-invariant manner. Note also that they are invariant to image rotations about the cameras principal axis. In some cases these methods can also detect and describe keypoint in an affine invariant manner. These are termed scale and affine invariant keypoint detectors.

Wide-baseline keypoint detection and description algorithms can be broadly classified into two categories. The first are those which utilise scale-space theory [133], an embedding of an image in a one parameter family of scale-space images (Gaussian smoothed versions of the original image). Those to be discussed include:

- Scale-space primal sketch and automatic scale selection of Lindeberg et al [132, 134, 136, 137]
- Scale Invariant Feature Transform (SIFT) of Lowe [142]
- Multi-scale Harris of Schmid et al [64]
- Harris-Laplace and Hessian-Laplace of Mikolajczyk et al [158]
- Harris-Affine and Hessian-Affine of Mikolajczyk et al [160, 159]
- Speeded-Up Robust Features (SURF - 'Fast Hessian') of Bay et al [19, 17]

The second use alternate image processing algorithms without the use of scale-space images. Those to be discussed include:

- Scale-saliency of Kadir and Brady [112, 115, 113, 114]
- Tuytelaars and Van Gool [228, 229, 230]
- Maximally Stable Extremal Regions (MSER) of Matas et al [151]

This section starts with a review of the scale-space approaches followed by the alternate approaches with respect to keypoint detection. This follows with a review of keypoint descriptors and matching. Finally, a number of works comparing the relative performance of keypoint detection and description algorithms is presented from which those most suited for visual-based localisation are identified.



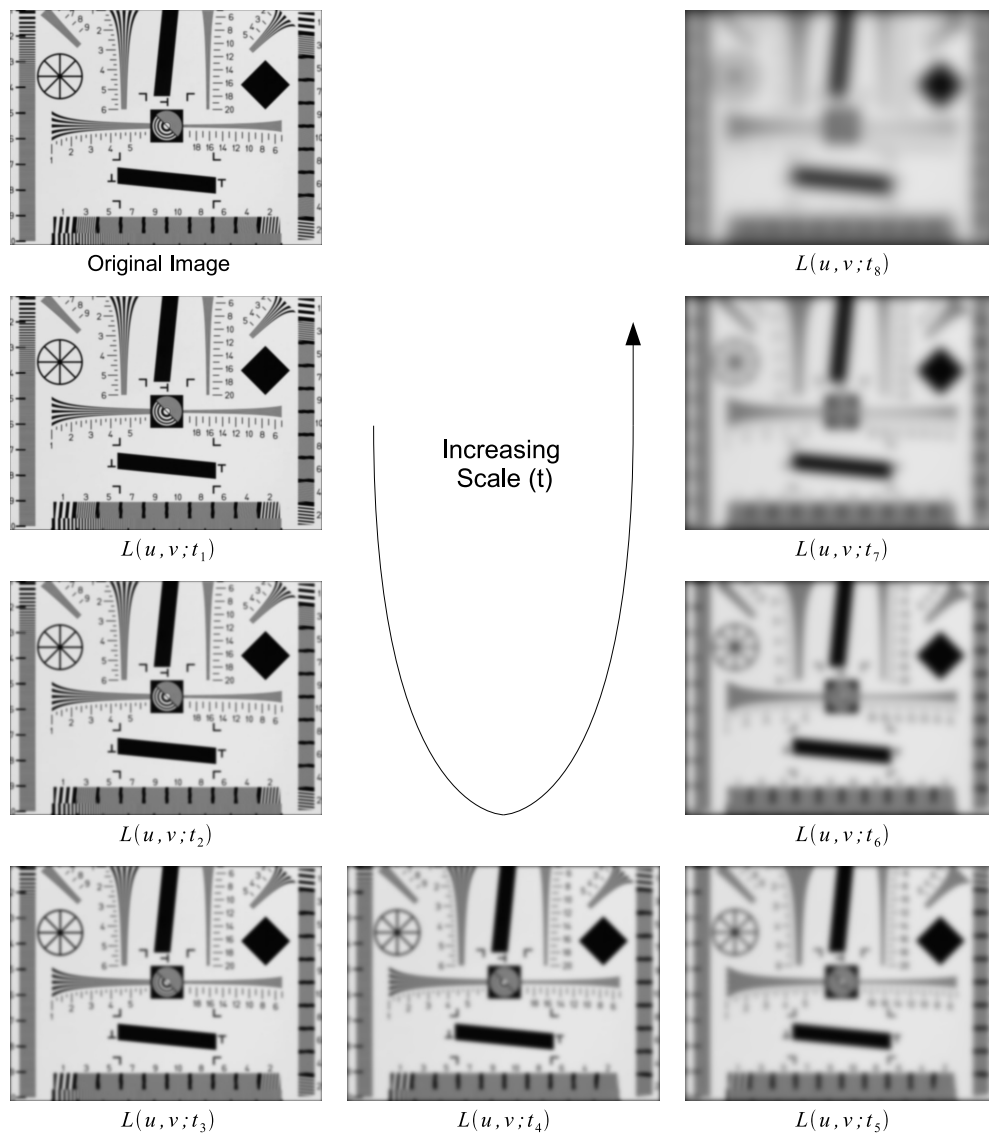


Figure 3.4: Example set of derived scale-space images  $L(u, v; t)$  with scale parameter  $t$  derived from the primal image  $I(u, v)$ . As the scale parameter  $t$  increases the fine detailed features in the image are suppressed and large features become more apparent.

### 3.3.1 Scale-Space Approaches

An image can be represented at multiple levels of scale by deriving a one parameter family of *scale-space* images  $L(u, v; t)$  from an original 'primal' image  $I(u, v)$  [134]. These scale-space images are obtained by convolving the primal image with a Gaussian of scale (variance)  $t$ . Figure 3.4 shows a series of scale-space images derived from an input primal image for increasing scales  $t$ . Observe that as the scale increases the fine detail in the image is suppressed. The scale-space approaches to image processing use this family of derived scale-space images for keypoint detection.

The term scale-space was popularised in western literature by Witkin [242] and Koenderink [123] who derived the unique solution as Gaussian convolution for one and two dimensional continuous signals respectively. However, as noted by Weickert et al [237] the same solution for 1D signals was first proposed in the work of Iijima (1959 — transcript in Japanese) [106]. Two dimensional signals are of interest here, and Koenderink [123] derived that the linear scale-space representation of some continuous two-dimensional signal  $L(x, y) \in \mathbb{R}^2$  satisfies the heat diffusion equation (with constant thermal conductivity)

$$\partial_t L = \frac{1}{2} \Delta L. \quad (3.24)$$

The variable  $t$  is a reference to time and  $\Delta$  the Laplacian operator

$$\Delta L = \frac{\partial^2 L}{\partial x^2} + \frac{\partial^2 L}{\partial y^2}. \quad (3.25)$$

The Green's function (fundamental solution) of the heat equation at time  $t$  is the Gaussian

$$g(x, y; t) = \frac{1}{2\pi t} e^{-\left(\frac{x^2+y^2}{2t}\right)}, \quad \int_{x,y} g(x, y; t) = 1, \quad (3.26)$$

and is the unique scale-space operator. Given some continuous function  $f \in \mathbb{R}^2$ , if the initial condition  $L(\cdot; t=0)$  at time  $t=0$  is the original function  $f$ , the scale-space representation  $L(\cdot; t)$  of  $f$  at time  $t$  is the convolution of  $f$  with the Gaussian  $g(\cdot; t)$ :

$$L(\cdot; t) = g(\cdot; t) * f, \quad (3.27)$$

where  $*$  is the convolution operator. The same notation as Lindeberg [134] is used here where  $L(\cdot; t)$  is defined to mean  $L(x, y; \sigma) \forall x, y \in \mathbb{R}^2$ . This solution states that some 2 dimensional heat distribution  $f$  evolves to  $L(\cdot; t)$  after time  $t$ . With respect to image processing, the function  $f$  is an image, and the parameter  $t$  is frequently referred to as scale and not time.

The unique solution for the Gaussian function has been derived from a range of scale-space axioms by a number of authors, including Witkin [242], Koenderink [123], Babaud et al [6], Yuille and Poggio [246] and Florack et al [74, 73], and Lindeberg [134, 135, 133] who provides a comprehensive review in his work. These axioms are a set of fundamental requirements for the scale-space representation of a signal. The solution of Koenderink [123] was based on the following axioms:

1. Causality: No spurious detail is created for increasing scale. Any image at higher scale is a simplification of another at a lower scale — all values can be derived

(traced) from the primal image.

2. Homogeneity: The convolution (diffusion) process is shift-invariant — with respect to the heat equation the function is assumed to have constant thermal conductivity. The scale-space operator remains fixed during convolution.
3. Isotropy: The diffusion process has no preferred direction — the diffusion kernel needs to be rotationally symmetrical

A number of other axioms have been used. Lindeberg presents two axioms which, coupled with the requirement of a continuous scale parameter, are sufficient for deriving the Gaussian as the unique scale-space operator [135, 134]. The first is a modified formulation of causality referred to as ‘non-enhancement of local extrema’, and the second the semigroup property. The former states that for linear scale-space, any non-degenerate local minima or maxima does not decrease or increase respectively under diffusion. This property can be formalised using the Laplacian  $\Delta$  of a function evaluated at some point, which is the product of the principal curvatures  $\lambda_1, \lambda_2$  of the function and equal to the trace of the Hessian matrix  $\mathcal{H}$ . If both curvatures are positive or negative the point is a local minima or maxima respectively. The non-enhancement of local extrema constraint can then be written

$$\text{Minima: } \partial_t L < 0 \quad \text{if} \quad \Delta L < 0 \quad (3.28)$$

$$\text{Maxima: } \partial_t L > 0 \quad \text{if} \quad \Delta L > 0, \quad (3.29)$$

whereby  $\text{sign} \partial_t L = \text{sign} \Delta L \Rightarrow \partial_t L \Delta L > 0$ . This has as the simplest solution

$$\partial_t L = \alpha \Delta L \quad (3.30)$$

for some  $\alpha > 0$ , and has the same form as the heat diffusion equation in 3.24.

The semi-group property is the other axiom used by Lindeberg to derive the Gaussian as the unique scale-space operator [135, 134], the Gaussian being a commutative semigroup operator. In short, it states that performing successive diffusion operations should be the same as performing a single diffusion. This is written by Lindeberg as:

$$L(\cdot; t_2) \stackrel{\text{def}}{=} h(\cdot; t_2) * f \quad (3.31)$$

$$= (h(\cdot; t_2 - t_1) * h(\cdot; t_1)) * f \quad (\text{semi-group}) \quad (3.32)$$

$$= h(\cdot; t_2 - t_1) * (h(\cdot; t_1) * f) \quad (\text{associative}) \quad (3.33)$$

$$= h(\cdot; t_2 - t_1) * L(\cdot; t_1). \quad (3.34)$$

The semi-group property is useful for efficient computation of scale-space images during image processing [142]. As the scale-parameter increases so does the size of the Gaussian function and computational expense of convolution with the primal image. Using the semigroup property, the scale-space representation of an image can be obtained via convolution of any scale-space image at a smaller scale with a Gaussian of scale  $t = t_2 - t_1$ , where  $t_2$  is the required scale and  $t_1$  the scale of some other scale-space image  $L(\cdot; t_1)$ .

Most derivations of scale-space is for continuous signals. A formal treatment for discrete signals was studied by Lindberg [131] which is of significance for image processing. Lindeberg derives the discrete analog of the Gaussian function as the unique scale-space kernel for discrete functions. The scale-space representation of a discrete function is the convolution of the function with this kernel. This kernel differs from the sampled Gaussian function which does not adhere to the semi-group property for all ratios of scales  $t_1$  and  $t_2$ . However, in practice the sampled Gaussian is most frequently used for image processing, and was the method used to obtain the scale-space images in figure 3.4. Define  $G(x, y; t)$  as the sampled Gaussian function  $g(x, y; t)$  at integer (pixel) positions  $x, y$ . In image processing literature, the sampled Gaussian function is more frequently parameterised as  $G(x, y; \sigma)$ , where  $\sigma = \sqrt{t}$  is the standard deviation of the sampled Gaussian function:

$$G(x, y; \sigma) = \frac{1}{2\pi\sigma^2} e^{-\frac{(x^2+y^2)}{2\sigma^2}}. \quad (3.35)$$

The scale-space representation  $L(u, v; \sigma)$  of some primal image  $I$  is

$$L(\cdot; \sigma) = G(\cdot; \sigma) * I, \quad (3.36)$$

which is evaluated at each pixel position  $u, v$  as

$$L(u, v; \sigma) = \sum_{x=-n}^{x=n} \sum_{y=-n}^{y=n} G(x, y; \sigma) I(u+x, v+y), \quad x, y \in \{-n, -n+1, \dots, n-1, n\}, \quad (3.37)$$

where  $n$  is the integer support size of the sampled Gaussian. A suitable normalisation is applied to the sampled Gaussian such that  $\sum_{x,y} G(x, y; \sigma) = 1$ . In addition to the semi-group property, the separability of the sampled Gaussian is exploited for efficient computation of scale-space images. Convolution of an image with a sampled 2D Gaussian can be implemented more efficiently with respect to computation by successive convolutions with a one-dimensional sampled Gaussian  $G_x(\cdot; \sigma)$  in the  $x$  direction and

a one-dimensional sampled Gaussian  $G_x(\cdot; \sigma)$  in the  $x$  direction as

$$L(\cdot; t) = G_x(\cdot; \sigma) * (G_y(\cdot; \sigma) * I). \quad (3.38)$$

### 3.3.1.1 Scale-Space Primal Sketch and Automatic Scale Selection

In addition to his work regarding scale-space theory, Lindeberg proposed a number of scale-space keypoint detection algorithms. One is the *scale-space primal sketch* [132, 134], a method which uses a small set of primitives for the detection of salient blob-like structures in scale-space corresponding to salient (significant) structures in an image. With respect to a greyscale image, a blob is any distinctive region of light or dark greyscale intensity value with respect to the surrounding image function. Light blobs can be found for example by applying a binary threshold to the grey level values of an image. The appearance of large blob like structures in an image is enhanced with increasing scale. See for example the appearance of the dark diagonal blob towards the right of figure 3.4 which is most distinctive with respect to the other image structure in the scale-space images at the largest scale.

Lindeberg finds the ‘base-level’ and spatial extent of all blobs in a given scale-space image via binary thresholding of the grey level values, as illustrated in figure 3.5. Notice in the figure that blobs merge and create new blobs as the threshold is reduced. The base-level is the greyscale threshold at which the blob merges with another, and the spatial extent is its boundary (edge), both of which define for each a 3D volume. This information is used to construct a grey-level ‘blob-tree’ for the scale-space image. Rather than identify salient blobs using only one scale-space image which is sensitive to image noise, Lindeberg finds blob-trees for a number of scale-space images and constructs a blob-tree in scale-space — each blob is a 4D entity having a volume at each scale. For the set of blobs found in a scale-space image, one would expect to see the same blobs in the scale-space images at a slightly higher or lower scale, that is, they are ‘connected’ in scale-space. Lindeberg considers that a singularity occurs when this is not the case and is the result of annihilation, merging, splitting and creation of blobs. The range of scales over which a blob is connected in scale-space is used to define its scale-space lifetime, and coupled with its spatial extent and contrast, is used to define its saliency. This saliency can then be used to select the  $n$  most salient blobs in the image. For each detected blob, the scale at which the normalised volume is a maximum is found and the spatial extent of the blob at this scale used to define the support region.

The scale-space primal sketch is used for focus-of-attention. The blobs found sig-

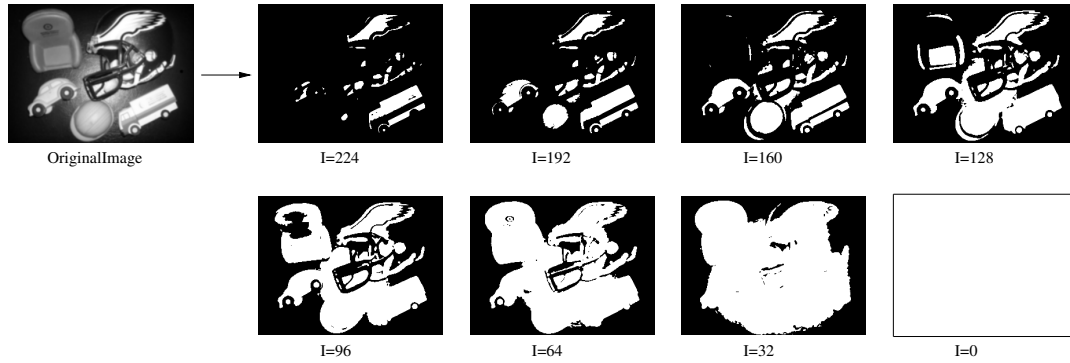


Figure 3.5: Detection of blob like structures in a greyscale image via binary thresholding of the grey level value. As the threshold level is reduced, light blobs appear and merge to create new blobs. In this example,  $I$  is the greyscale threshold level where the input image has greyscale intensity values in the range 0-255.

nify interesting regions in the image and are not keypoints themselves. These interesting regions can for example be used for focused edge detection and segmentation. More relevant to this discussion is their use for keypoint detection (Lindeberg uses the term junction). For a blob whose normalised volume is a maxima at some scale  $\sigma$ , Lindeberg [132, 134] finds the grey level curvature  $k$  for the scale-space image  $L(\cdot; \sigma)$  as

$$k = |L_{uu}L_v^2 + L_{vv}L_u^2 - 2L_{uv}L_uL_v|. \quad (3.39)$$

Keypoints are found by detecting ‘curvature blobs’ in this curvature image, where the spatial extent of each blob defines its support region. Lindeberg notes that this process is suited mainly to finding potential candidate keypoints as the localisation accuracy of blobs is poor. Alternate methods for keypoint localisation need to be used at finer scales for improved accuracy.

In later work Lindeberg considers automatic scale selection for keypoints [136, 137] which has since become the basis for many scale-invariant keypoint detection algorithms [142, 158, 160, 156]. The idea is to select keypoints as local extrema in scale for some some scale-normalised saliency metric assessed from differential operators. Scale normalisation is necessary since the magnitude of the gradients computed in a scale-space image at a large scale will in general be smaller than those computed in a scale-space image at a small scale. This property is related to the non-enhancement of local extrema principle. Lindeberg derives then a normalised measure of image derivatives which are invariant to scale. Consider two images  $I(\mathbf{x}) = I'(\mathbf{x}')$  where the coordinates are related by some scale factor  $s$ :  $\mathbf{x}' = s\mathbf{x}$ . The scale space images are then related by  $L(\mathbf{x}; t) = L'(\mathbf{x}'; t')$  where  $t' = s^2t$ . For perfect scale-invariance under a

rescaling of the image, the  $n^{\text{th}}$  order spatial derivatives satisfy the constraint

$$\partial_{\mathbf{x}^n} L(\mathbf{x}; t) = s^n \partial_{\mathbf{x}^n} L'(\mathbf{x}'; t'), \quad (3.40)$$

where  $\sigma = s$ .

A number of keypoint detectors based on this principle were proposed by Lindeberg [136, 137]. The first is used to finding keypoints associated with corner and edge points using a scale-normalised version of the Kitchen-Rosenfeld metric, a measure of the rate of change of curvature of a contour line defined by the saliency metric  $k_{norm}$  computed at some scale  $\sigma$  as

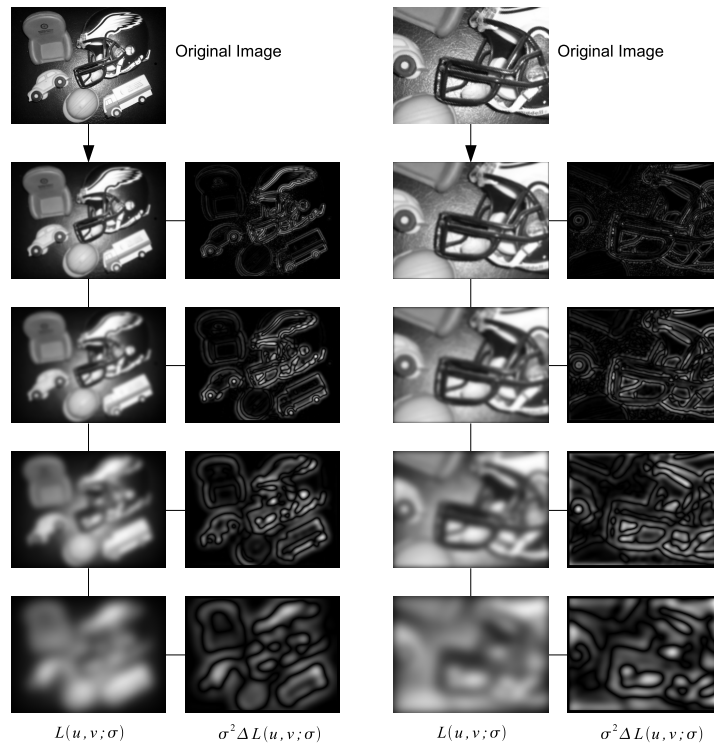
$$k_{norm} = \sigma^2 (L_{uu}L_v^2 + L_{vv}L_u^2 - 2L_uL_vL_{uv}). \quad (3.41)$$

For each candidate keypoint found as a local extrema in  $k_{norm}$  over scale, a secondary step is used to improve the localisation of the keypoint in scale-space where full details are given in [136]. The second method is used for detection of keypoints associated with blob like structures using the scale-normalised Laplacian of Gaussian  $\Delta G$  to define the saliency  $k_{norm}$  at scale  $\sigma$  as

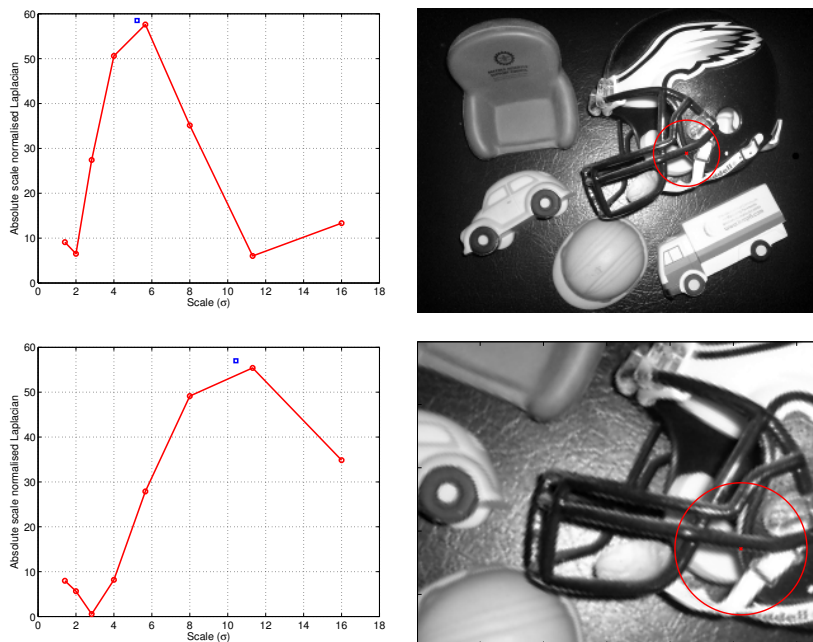
$$k_{norm} = \sigma^2 I * \Delta G(\cdot; \sigma) = \sigma^2 [L_{uu}(\cdot; \sigma) + L_{vv}(\cdot; \sigma)]. \quad (3.42)$$

For either saliency, the scale at which the extrema occurs is the ‘characteristic’ scale  $\sigma$  of the keypoint which is used to assign the keypoint support region radius. To illustrate, figure 3.6a shows for two images of the same scene the absolute value of the scale-normalised Laplacian for each calculated over a range of scales. These two images have a viewpoint change differing in rotation about the camera’s principal axis and scale change. Figure 3.6b shows, for the same keypoint (i.e. the same scene point) in the two images, the scale-normalised Laplacian of Gaussian computed at each scale for this point. The square mark is the interpolated position of the local extrema of the function in scale which defines the characteristic scale for each keypoint. The circular support region has been overlaid for each keypoint with a radius  $r$  proportional set proportional to the characteristic scales of the keypoints. Observe that although the radii are different, each support region encloses the same region of the scene. This concept of automatic scale selection is a powerful tool for wide-baseline keypoint detection and matching and is the basis for many state of the art algorithms.





(a) For each image, the left column is the set of scale-space images and the right column is the scale-normalised Laplacian evaluated at the same scales as the scale-space images (the absolute values are shown for illustrative purposes).



(b) Automatic scale selection for the same keypoint in the two images.

Figure 3.6: Automatic scale-selection using the scale-normalised Laplacian. The characteristic scale is found by searching for local extrema in the scale-normalised Laplacian over scale. The blue square is the interpolated position of the extrema.



### 3.3.1.2 Multi-scale Harris, Harris-Laplace, Hessian-Laplace, Harris-Affine and Hessian-Affine

An approach to matching images with different resolutions (scales) was presented by Dufournaud, Schmid and Horaud in [64]. They developed a multi-scale Harris keypoint detector which is used to find scale-normalised Harris keypoints independently in a number of scale-space representations of a high resolution image. These keypoints are matched to a set of Harris corners found in a low resolution image using a homography between images to evaluate the correct correspondences. The scale at which the greatest number of correct correspondences is found represents the relative scale (resolution) between the images. A limitation of this method is the inability to index features, that is, store feature descriptors which may be compared to any other image at any resolution without having to compute first the relative scale.

Mikolajczyk and Schmid [158] developed the Harris-Laplace keypoint detection algorithm, an extension of the multi-scale Harris detector with automatic scale selection [136]. The algorithm was initially applied to object recognition and image retrieval, where the use of automatic scale selection eliminated the need to store a unique set of keypoints at each level of scale-space. For some input image, they find the scale-space representation of the image at 17 scales, starting at  $\sigma = 1.5$  and increasing by a value of 1.5. They search first for candidate keypoints as being local extrema in the scale-normalised Harris corner strength above some threshold compared to the nearest 8 neighbours in the current scale-space image, as illustrated in figure 3.7a. The scale-normalised Harris corner strength is

$$C(\sigma_L, \sigma) = \det(A(\sigma_L, \sigma)) - k \times \text{trace}(A(\sigma_L, \sigma))^2, \quad (3.43)$$

where  $A$  is the auto correlation matrix

$$A(\sigma_L, \sigma) = \sigma^2 G(\cdot; \sigma_l) * \begin{bmatrix} L_u^2(\cdot; \sigma) & L_u L_v(\cdot; \sigma) \\ L_u L_v(\cdot; \sigma) & L_v^2(\cdot; \sigma) \end{bmatrix}. \quad (3.44)$$

The Gaussian  $G(\cdot; \sigma_l)$  of integral scale  $\sigma_l$  is used to integrate the values of the auto-correlation matrix over a small local region surrounding each pixel.

For each candidate keypoint, the scale-normalised Laplacian is used to assess if the candidate keypoint is a local extrema in scale by comparing it to the same pixel position in the adjacent scale space images, as shown in figure 3.7b. The scale-normalised Laplacian is

$$\Delta = \sigma^2 (L_{uu}(\cdot; \sigma) + L_{vv}(\cdot, \sigma)). \quad (3.45)$$

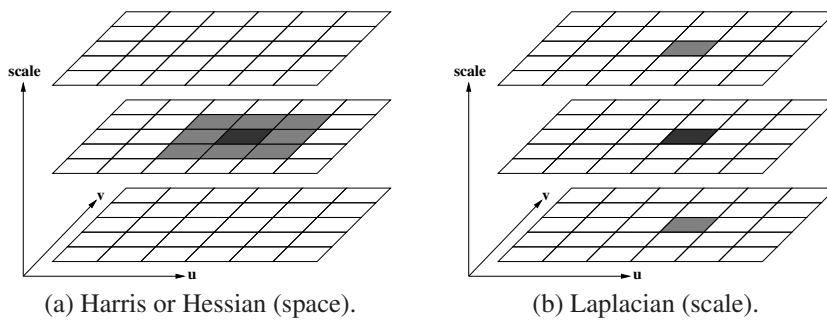


Figure 3.7: Local extrema for the Harris-Laplace and the Hessian-Laplace algorithms. Keypoints are found first as local extrema compared to their nearest neighbours in the current space-image (a). A keypoint is a local extrema in scale if the scale-normalised Laplacian of the keypoints is a local extrema compared to the scale-normalised Laplacian of the same pixel in the adjacent scale-space images (b).

Assuming the candidate keypoint is a local extrema, the characteristic scale is  $\sigma$ . The radius of the support region for the keypoint is set proportional to this characteristic scale. They state that the scale normalised Harris corner measure is used to find the position of candidate keypoints as it provides high repeatability for 2D localisation in images subject to image rotations, illumination transformations, and perspective deformation. It is not used for automatic scale selection as it rarely attains a maxima over scales. The Laplacian is more robust for this purpose with respect to the repeatability of keypoints found between different images of the same scene. A similar keypoint detector termed Hessian-Laplace was proposed by Mikolajczyk [157] using the scale-normalised determinant of Hessian to select candidate keypoints as local extrema above some threshold with respect to the nearest 8 neighbours in the current scale-space image. Automatic scale-selection is achieved again using the Laplacian of Gaussian with respect to the same pixel location in the adjacent scales.

A scale and affine invariant version of the Harris-Laplace and Hessian-Laplace algorithms, termed Harris-Affine and Hessian-Affine respectively, were developed by Mikolajczyk et al [159, 160, 156]. Keypoints are first found using the Harris-Laplace or Hessian-Laplace keypoint detectors. Affine-invariance is then found using an iterative normalisation of the autocorrelation matrix  $A$  first explored and used for scale and affine invariant keypoint detection by Lindeberg and Garling [138] and later by Baumberg [16]. The iterative normalisation scheme is analogous to a local search in affine scale-space (convolution of an image with an elliptical Gaussian having different scales in orthogonal directions [133]) for which the ratio of eigenvalues  $\lambda_1, \lambda_2$  of the computed autocorrelation matrix are unity. The approximate scale and location of keypoints are found first using the Harris-Laplace or Hessian-Laplace algorithms as a full search in affine scale-space is not practical with respect to computational expense.

The ratio of eigenvalues is used as it is an estimate of the anisotropic shape of the local image function [138], where any anisotropic region is assumed to be an affine transformed version of an isotropic region [160]. Hence, the search in affine scale-space finds the affine transform which converts the anisotropic image structure to an isotropic structure. Unlike the circular support region for scale-invariant methods, the support regions for the Harris-Affine and Hessian-Affine algorithms are ellipses.

Scale and affine invariant keypoint detection algorithms have also been developed by Baumberg [16] and Schaffalitzky and Zisserman [201]. Baumberg finds Harris corners independently in scale space-images and then uses the iterative normalisation of the autocorrelation matrix to achieve affine invariance. However, this normalisation is only for the shape of the affine transform and not the scale or position of the keypoint. The fact that only the shape of the affine transform and not the scale or keypoint location are adjusted during this normalisation limits the performance [230], particularly for large affine deformations where the position of the keypoint can be affected significantly as a result of the initial location being found using circular operators [160]. As no automatic scale selection is used the same keypoints can be found in neighbouring scale-space images which has disadvantages during matching [160]. The method of Schaffalitzky and Zisserman [201] accounts for this by finding keypoints using the Harris-Laplace algorithm. However, similarly to Baumberg only the shape of the affine transform is found using the iterative normalisation of the autocorrelation matrix which limits its performance with respect to keypoint localisation for large affine deformations. In contrast, the Harris-Affine and Hessian-Affine iterative for the shape and scale of the affine deformation as well as the position of the keypoint during normalisation making it more robust for large affine deformations. The Harris-Affine algorithm has been shown to provide a greater repeatability of keypoints in planar images subject to viewpoint change than both the method of Baumberg in [159] and Schaffalitzky and Zisserman in [160].

### 3.3.1.3 The Scale-Invariant Feature Transform (SIFT)

The Scale-Invariant Feature Transform (SIFT) was first proposed by Lowe in [143] and refined in following works by Brown and Lowe [29] with the final algorithm given in [142]. SIFT has been used for a range of applications including vision-based SLAM [205], augmented reality [211] and construction of scenes mosaics [28]. Although SIFT includes both a means of scale-invariant keypoint detection and description, this section will discuss only the keypoint detection phase.

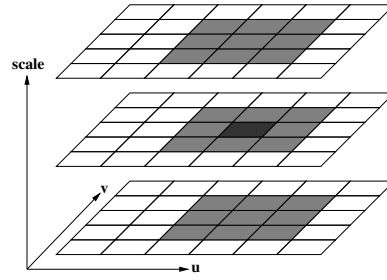


Figure 3.8: SIFT selects candidate keypoints as local extrema compared to the 26 nearest pixels in the current and adjacent DoG images.

SIFT uses as a saliency metric difference of Gaussian (DoG) images  $D(\cdot; \sigma)$  obtained by subtracting neighbouring scale space images

$$D(\cdot; \sigma) = (G(\cdot; k\sigma) - G(\cdot; \sigma)) * I \quad (3.46)$$

$$= L(\cdot; k\sigma) - L(\cdot; \sigma) \quad (3.47)$$

where  $k$  is some multiplicative factor. A pixel is selected as a candidate keypoint if it has an absolute DoG value above some threshold and is a local extrema in the DoG images compared to the nearest 26 neighbouring pixels in the current and adjacent scales, as shown in figure 3.8.

Lowe [142] shows that the difference of Gaussian function  $G(\cdot; k\sigma) - G(\cdot; \sigma)$  is a close approximation to the scale-normalised Laplacian  $\sigma^2 \Delta G$  [136, 134] which finds predominantly significant blob like structures in an image. This approximation was illustrated by Lowe [142] by writing the heat diffusion equation parameterised by  $\sigma = \sqrt{t}$  as

$$\frac{\partial G}{\partial \sigma} = \sigma \Delta G \quad (3.48)$$

which can be approximated from the difference of nearby functions as

$$\frac{\partial G}{\partial \sigma} \approx \frac{G(\cdot; k\sigma) - G(\cdot; \sigma)}{k\sigma - \sigma} \quad (3.49)$$

at scales  $k\sigma$  and  $\sigma$ . From back substitution Lowe shows that

$$G(\cdot; k\sigma) - G(\cdot; \sigma) \approx (k - 1)\sigma^2 \Delta G, \quad (3.50)$$

where the approximation error converges to zero as  $k$  approaches 1. As the factor  $(k - 1)$  is constant for all scales, Lowe argues that if scale-space images are used for increasing scales separated by a constant factor  $k$  then the approximation will have no influence on the robustness on keypoint detection. Selecting local extrema in the DoG

images is therefore similar to selecting local extrema in the scale-normalised Laplacian. This has been validated in experiments of Mikolajczyk and Schmid [158] with respect to the repeatability of scale-invariant keypoint detection for each respective method.

For efficient computation of scale-space images, an octave based approach to image processing is used as illustrated in figure 3.9. An octave is a halving of a scale-space image which increases the scale  $\sigma$  by a factor of two. The original image is first doubled in size and pre-smoothed by a Gaussian with standard deviation  $\sigma = 1.6$ . An octave is a doubling of the standard deviation of the initial convolution  $2\sigma$ . An integer number of  $s = 3$  scales per octave is used where  $k = 2^{1/s}$ . For the purpose of extrema detection, a total of six scale-space images are found per octave from which 5 difference of Gaussian images are found. The middle 3 are then used for extrema detection. The third scale-space image from the top is then subsampled by a factor of two (taking every second pixel in row and column) and is the start image for the next octave. The number of octaves used is dependent on the size of the original image. The process is terminated when the size of the images in a given octave falls below some threshold.

The location and scale of each candidate keypoint is refined using a method of 3D quadratic interpolation introduced by Brown and Lowe in [29] which they state improves significantly the repeatability of keypoints found between images of the same scene. The interpolated position is found by finding the position of the extrema of the second order Taylor expansion of the  $3 \times 3 \times 3$  DoG function about the candidate keypoint position. To illustrate this process, consider the local 1D difference of Gaussian (DoG) function  $D$  about a pixel at the origin. The approximate DoG function evaluated at some shift  $x$  from the pixel can be estimated from the second order Taylor expansion

$$D(x) = D + D'x + D''x^2 \quad (3.51)$$

where  $D' = \frac{\partial D}{\partial x}$  and  $D'' = \frac{\partial^2 D}{\partial x^2}$  are the first and second order derivatives of  $D$  evaluated at the origin. Setting the derivative of equation 3.51 to zero, the interpolated position  $\hat{x}$  at which the function  $D$  is a maxima can be found:

$$0 = D' + D''\hat{x}, \quad \hat{x} = -\frac{D'}{D''}, \quad (3.52)$$

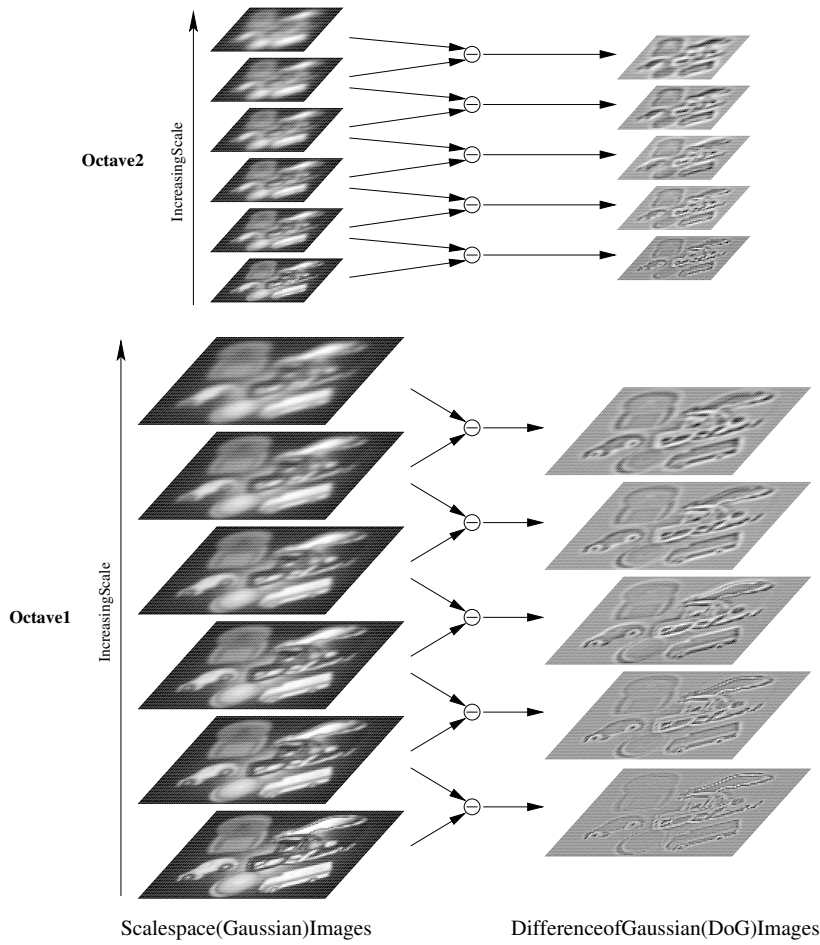


Figure 3.9: Octave approach use by SIFT for efficient image processing. SIFT uses three scales per octave which requires 6 scale-space image to be produced each octave. Five DoG images can then be found for each octave, and the middle three are used to select candidate keypoints.

where, using 3.51, the estimated maxima of  $D$  evaluated at  $\hat{x}$  is

$$D(\hat{x}) = D + D'\hat{x} + \frac{1}{2}D''\hat{x}^2 \tag{3.53}$$

$$= D + D'\hat{x} + \frac{1}{2}D'' \left( -\frac{D'}{D''} \right)^2 \tag{3.54}$$

$$= D + D'\hat{x} - \frac{1}{2}D'\hat{x} \tag{3.55}$$

$$= D + \frac{1}{2}D'\hat{x}. \tag{3.56}$$

The second order Taylor expansion of the three dimensional DoG function  $D$  about a candidate keypoint is

$$D(\mathbf{x}) = D + \frac{\partial D}{\partial \mathbf{x}}^T \mathbf{x} + \frac{1}{2} \mathbf{x}^T \frac{\partial^2 D}{\partial \mathbf{x}^2} \mathbf{x}, \tag{3.57}$$

where  $\mathbf{x} = (x, y, t)^T$  is an offset from the position of the candidate keypoint in scale-space.  $\frac{\partial D}{\partial \mathbf{x}}$  and  $\frac{\partial^2 D}{\partial \mathbf{x}^2}$  are the  $3 \times 1$  matrix of first order partial derivatives and  $3 \times 3$  matrix of second order partial derivatives of the DoG function respectively, both evaluated at the position and scale of the candidate keypoint. By setting the derivative of equation 3.57 to zero, the position  $\hat{\mathbf{x}}$  of the extrema in scale-space relative to the candidate keypoint position in scale-space is evaluated as

$$\hat{\mathbf{x}} = -\frac{\partial^2 D}{\partial \mathbf{x}^2}^{-1} \frac{\partial D}{\partial \mathbf{x}}, \quad (3.58)$$

and the estimate of  $D(\hat{\mathbf{x}})$  is

$$D(\hat{\mathbf{x}}) = D + \frac{1}{2} \frac{\partial D}{\partial \mathbf{x}}^T \hat{\mathbf{x}}. \quad (3.59)$$

This new estimate is then used to test again if the absolute DoG value of the keypoint is above the required threshold. SIFT uses an iterative scheme for the quadratic interpolation of a keypoint. If either of the pixel coordinates of  $\hat{\mathbf{x}}$  exceed 0.5, the candidate keypoint position is shifted accordingly and the process is repeated.

As the DoG images are a close approximation to the scale-normalised Laplacian, the saliency metric used to select the candidate keypoints considers only the sum of principal curvatures of the image function. The candidate keypoints can therefore be found on edges which is not desirable. A means for rejecting keypoints on edges (edge removal) was presented by Lowe in [142] using the Hessian matrix  $\mathcal{H}$  of the DoG function computed at each candidate keypoint position  $\mathbf{u}$  and scale  $\sigma$ , where

$$\mathcal{H}(\mathbf{u}; \sigma) = \begin{bmatrix} D_{uu}(\mathbf{u}; \sigma) & D_{uv}(\mathbf{u}; \sigma) \\ D_{uv}(\mathbf{u}; \sigma) & D_{vv}(\mathbf{u}; \sigma) \end{bmatrix}. \quad (3.60)$$

The eigenvalues  $\lambda_1, \lambda_2$  of the Hessian matrix are the principal curvatures of the DoG function evaluated at  $D(\mathbf{u}; \sigma)$ . Using the property  $\text{trace}(\mathcal{H}(\mathbf{u}; \sigma)) = \lambda_1 + \lambda_2$  and  $\det(\mathcal{H}(\mathbf{u}; \sigma)) = \lambda_1 \lambda_2$ , SIFT considers that a keypoint is not on an edge if the ratio of the maximum to minimum eigenvalues are below a threshold  $r = \frac{\lambda_1}{\lambda_2}$ :

$$\frac{\text{trace}(\mathcal{H})^2}{\det(\mathcal{H})} = \frac{(\lambda_2 + r\lambda_2)^2}{\lambda_2^2 r} = \frac{(1+r)^2}{r}. \quad (3.61)$$

A value  $r = 10$  is used where a keypoint is assumed not to be an edge response if

$$\frac{\text{trace}(\mathcal{H})^2}{\det(\mathcal{H})} < \frac{(1+r)^2}{r}. \quad (3.62)$$



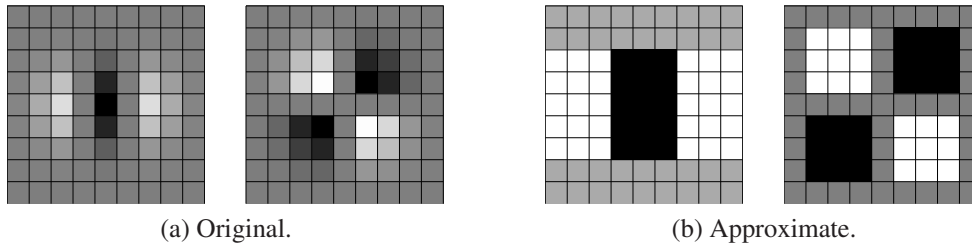


Figure 3.10: Box filter approximates of second order derivative of Gaussians  $G_{uu}$  and  $G_{uv}$  used by SURF.

Those candidate keypoints that are not edge responses are selected as SIFT keypoints and are defined by a pixel location and scale  $\sigma$ . The radius of the support region for each keypoint is set proportional to its scale  $\sigma$ .

### 3.3.1.4 Speeded up Robust Features (SURF) - ‘Fast Hessian’

A scale-invariant keypoint detection and description algorithm was developed by Bay et al [19, 17] called Speeded-Up Robust Features (SURF). The keypoint detection stage discussed here is often referred to as the ‘fast Hessian’ keypoint detector. The saliency metric used for keypoint detection is based on the Hessian matrix  $\mathcal{H}(\mathbf{u}; \sigma)$ , evaluated at some pixel position  $\mathbf{u}$  and scale  $\sigma$  as

$$\mathcal{H}(\mathbf{u}; \sigma) = \begin{bmatrix} L_{uu}(\mathbf{u}; \sigma) & L_{uv}(\mathbf{u}; \sigma) \\ L_{uv}(\mathbf{u}; \sigma) & L_{vv}(\mathbf{u}; \sigma) \end{bmatrix}. \quad (3.63)$$

One way to find the values  $L_{vv}(\cdot; \sigma)$  would be to convolve the image with the sampled and cropped second order derivative of Gaussian  $G_{vv}(\cdot; \sigma)$ . The fast Hessian detector finds approximate estimates for the values of the Hessian matrix using integral images [235] and box filters. These box filters are approximations of the sampled and cropped second order derivative of Gaussian functions computed at scale  $\sigma$  as illustrated in figure 3.10 for a scale  $\sigma = 1.2$ . Using simple box filters and integral images allows fast correlation of the image with the box filters to obtain the estimates of  $L_{uu}(\cdot; \sigma)$ ,  $L_{uv}(\cdot; \sigma)$  and  $L_{vv}(\cdot; \sigma)$ , denoted by  $\tilde{L}_{uu}(\cdot; \sigma)$ ,  $\tilde{L}_{uv}(\cdot; \sigma)$  and  $\tilde{L}_{vv}(\cdot; \sigma)$  respectively, whose processing time is independent of the size of the box filters. This eliminates the need for the octave based approach of SIFT where the image size is halved each octave, a process that can limit the accuracy of keypoint localisation at high octaves.

The approximate Hessian matrix  $\mathcal{H}_{approx}$  is found at a number of scales, obtained



at any scale  $\sigma$  as

$$\mathcal{H}_{approx}(\mathbf{u}; \sigma) = \begin{bmatrix} \tilde{L}_{uu}(\mathbf{u}; \sigma) & \tilde{L}_{uv}(\mathbf{u}; \sigma) \\ \tilde{L}_{uv}(\mathbf{u}; \sigma) & \tilde{L}_{vv}(\mathbf{u}; \sigma) \end{bmatrix}. \quad (3.64)$$

This is used to find the saliency metric  $C$ , referred to as the blob response map, based on its determinant as

$$C(\cdot; \sigma) = \tilde{L}_{uu}\tilde{L}_{vv} - (w\tilde{L}_{uv}^2) \approx \det(\mathcal{H}_{approx}), \quad (3.65)$$

where  $w = 0.9$  is a weighting factor. As all box filters used are normalised to have equal Forbenius norm, no scale-normalisation (i.e. multiplication of  $C$  by  $\sigma^2$ ) is required [19, 17]. A keypoint is selected as a local maxima in  $C$  compared to the 26 nearest pixels in the current and adjacent scales. A Hessian based saliency metric is used as it was found in [160, 161] to give good repeatability with respect to keypoint detection in scale-space. The location and scale of each keypoint is then interpolated using the same 3D quadratic scheme proposed by Brown and Lowe [29] which was used for SIFT. The keypoint support region is set proportional to the characteristic scale  $\sigma$ . As a final note, there are two versions of the fast-Hessian keypoint detector; FH-9 which uses the original sized image, and FH-15 which first doubles the image size. These numbers denote the width of the box filters used at the lowest scale.

## 3.3.2 Alternate Approaches

### 3.3.2.1 Scale-Saliency

A novel scale-invariant keypoint detector developed for target tracking and recognition was presented by Kadir and Brady in [113] and the dissertation of Kadir [112]. Their method, referred to here as scale-saliency, was inspired by the work of Gilles [88] who suggested that the complexity of the local image content within the neighbourhood of a pixel could be used as a measure of its saliency. Gilles used as a quantitative measure of the complexity of this region the Shannon entropy of the probability distribution function (PDF) of greyscale intensity values. A region with high complexity has a near uniform PDF with high entropy and is therefore assigned a large saliency value. This idea was extended to scale-invariant keypoint detection by Kadir and Brady.

The scale-saliency algorithm operates as follows. For each pixel in the image at some position  $\mathbf{u} = (u, v)^T$ , a series of PDF's of the greyscale intensity values within circular regions of scale  $s$  surrounding the pixel are found, where  $s$  is the diameter of the region with values in the range 7-43 pixels. The PDF's have descriptor (bin) values

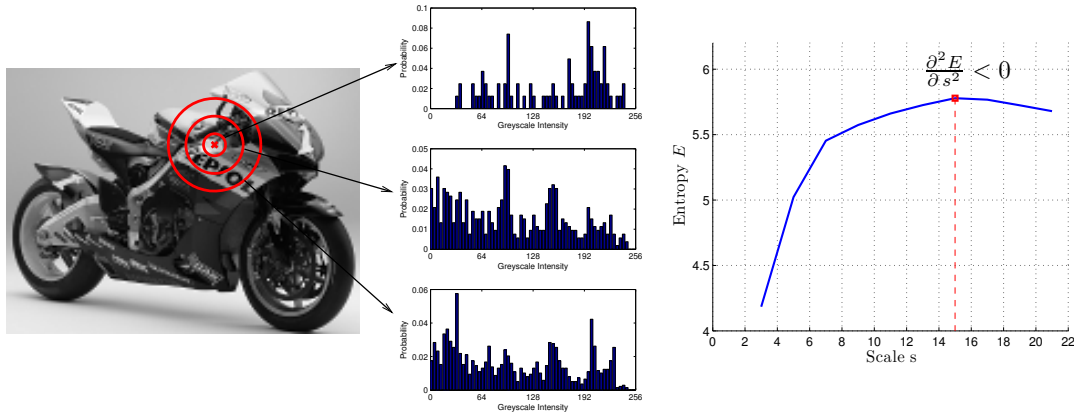


Figure 3.11: Automatic scale-selection for the scale-saliency keypoint detector. The PDF's of the greyscale intensity values surrounding a pixel within circular regions of scale (diameter)  $s$  are found. The entropy of the PDF's is then obtained, and a characteristic scale found as an extrema of the entropy function over scale.

$D$ , and  $p_D(s, \mathbf{u})$  is the PDF value at some scale  $s$  for a descriptor value  $D$ . The entropy  $\mathcal{E}$  is measured from the PDF at each scale  $s$  as

$$\mathcal{E}(s, \mathbf{u}) \triangleq \int_{i \in D} p_D(s, \mathbf{u}) \log_2 p_D(s, \mathbf{u}) . di. \quad (3.66)$$

A type of automatic scale-selection is used which finds a vector of scales  $\mathbf{S}$  for which the entropy  $\mathcal{E}$  is a local maxima over scale:

$$\mathbf{S} \triangleq \left\{ s : \frac{\partial^2 \mathcal{E}(s, \mathbf{u})}{\partial s^2} < 0 \right\}, \quad (3.67)$$

as illustrated in figure 3.11. If  $\mathbf{S}$  is not empty, then the pixel is assigned a saliency value  $\mathcal{Y}(\mathbf{S}, \mathbf{u}) \in \mathbb{R}^3$  (scale and image space) defined as

$$\mathcal{Y}(\mathbf{S}, \mathbf{u}) \triangleq \mathcal{E}(\mathbf{S}, \mathbf{u}) \times \mathcal{W}(\mathbf{S}, \mathbf{u}) \quad (3.68)$$

where  $\mathcal{W}(\mathbf{S}, \mathbf{u})$  is a weighing function which measures the dissimilarity of  $p_D(s, \mathbf{u})$  over scale:

$$\mathcal{W}(s, \mathbf{u}) \triangleq s \cdot \int_{i \in D} \left| \frac{\partial}{\partial s} p(s, \mathbf{u}) \right| . di, \quad (3.69)$$

where  $s$  is an element of the vector  $\mathbf{S}$ . The final set of keypoints are found by thresholding and clustering the sparse population of saliency values  $\mathcal{Y}$  in scale-space. A scale and affine invariant version of this algorithm was developed later by Kadir and Brady in [114] which uses elliptical support regions parameterised by scale, rotation and aspect ratio.

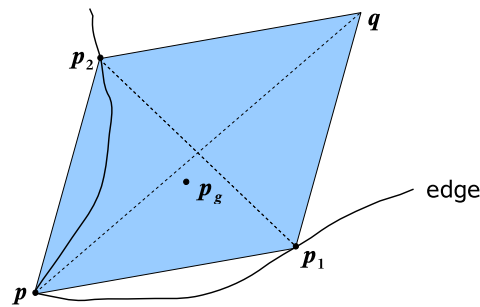


Figure 3.12: Geometry based method. A parallelogram is used to enclose an affine invariant region based on the greyscale intensity function along line segments originating from corner points.

### 3.3.2.2 Tuytelaars and Van Gool

Two scale and affine invariant feature detectors have been proposed by Tuytelaars and Van Gool [230], both of which may be used together in an opportunistic way to increase the number of keypoints detected and matched between images. The first is the geometry-based method first described in [228], and the second the intensity-based method first described in [229].

The geometry-based method detects in an image Harris corner points and edges using the Canny edge detector [36]. The Harris corners are considered to be ‘anchor points’. For a Harris corner  $\mathbf{p} = (u, v)^T$  detected on or near an edge, let  $\mathbf{p}_1$  and  $\mathbf{p}_2$  be two points moving away from the corner point in different directions along the edge. The two vectors  $\mathbf{p}_1 - \mathbf{p}$  and  $\mathbf{p}_2 - \mathbf{p}$  define a region in the image enclosed by a parallelogram with corners  $\mathbf{p}, \mathbf{p}_1, \mathbf{p}_2, \mathbf{q}$  and centre of gravity  $\mathbf{p}_g$ , as shown in figure 3.12. The position of the points  $\mathbf{p}_1$  and  $\mathbf{p}_2$  are selected over some fixed interval by searching for the minima of several functions for which the centre of gravity is close to, or coincident with, the diagonals of the parallelogram — there are some variations in this process dependent on the edge being straight or curved. The final parallelogram defines a scale and affine invariant region in the image. Unlike most wide-baseline keypoint detection algorithms discussed, the geometry-based method uses edges which have also been used for alternate wide-baseline keypoint detection algorithms by Bay et al [18] and Goedeme et al [90].

A potential limitation of the geometry-based method is the fact that the Harris corners and Canny edges are detected at a single scale. If there is a large scale change between images there is no guarantee that the same corner points and edges would be found. This may limit the overall ability to find keypoint correspondences between images. Tuytelaars and Van Gool note that the ability to find the same edges in particular

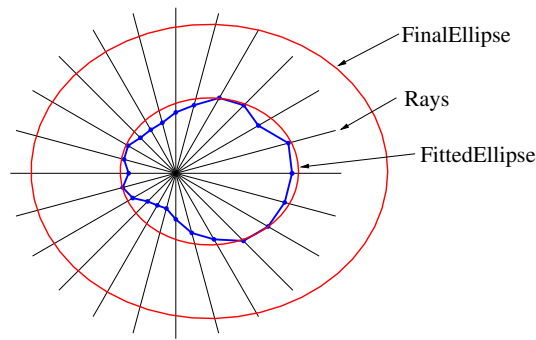


Figure 3.13: The intensity-based algorithm finds the point on each ray originating from an anchor point for which  $f_I(t)$  in 3.70 is a maxima. These points are then connected and an ellipse is fitted. This ellipse is then doubled in size, and this final ellipse defines the support region for the keypoint.

is the greatest source of error [230]. Furthermore, the use of the Harris corner detector in general finds keypoints near depth discontinuities in the scene [230].

The intensity-based algorithm is designed for keypoint detection on planar surfaces and does not require the detection of Harris corners or edges. It selects as anchor points local extrema of the greyscale image function which they claim occur predominantly on planar surfaces. They extend rays from the anchor point and assess the intensity function  $f_I(t)$  along each ray:

$$f_I(t) = \frac{\text{abs}(I(t) - I_0)}{\max\left(\frac{\int_0^t \text{abs}(I(t) - I_0) dt}{t}, d\right)}, \quad (3.70)$$

where  $t$  is the Euclidean arc-length along the ray,  $I(t)$  the intensity at position  $t$ ,  $I_0$  the intensity extrema and  $d$  a small number used to prevent division by zero. A scale and affine invariant region for the keypoint is found by connecting the points along each ray at the distance  $t$  for which the function  $f_I(t)$  is a maxima. This maxima typically occurs when there is a large variation in the intensity function along a ray. An ellipse is then fitted to this region using moments and doubled in size, as illustrated in figure 3.13. The intensity function within this ellipse is then used to produce a descriptor for the keypoint. They note that the intensity-based method is not able to localise the position of the anchor points as accurately as the geometry-based method which uses Harris corners capable of subpixel accuracy. However, they found that inaccuracies in the anchor point positions has little effect on the shapes of the final ellipses.

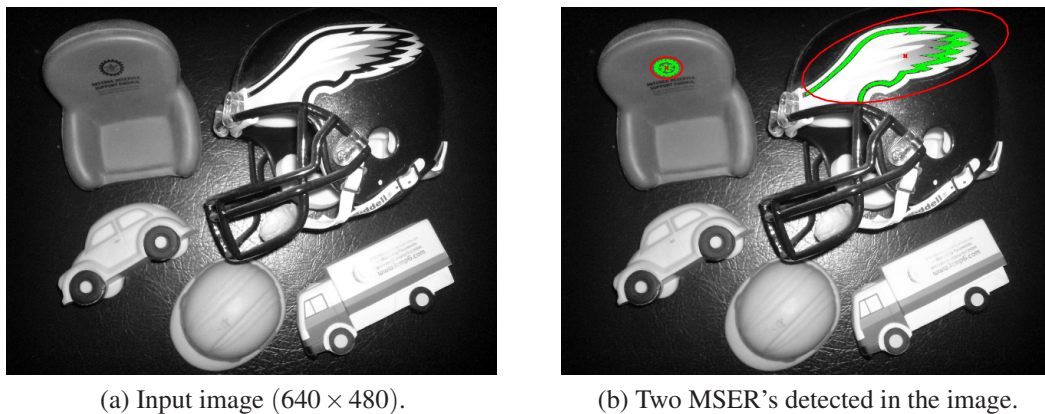


Figure 3.14: Example of two MSER keypoints detected in the image on the left. The position of each keypoint is marked with a cross. The ellipses are the scale and affine invariant support regions of the keypoints.

### 3.3.2.3 Maximally Stable Extremal Regions (MSER's)

Matas et al [151] proposed a means for scale and affine invariant keypoint detection. They define keypoints as Maximally Stable Extremal Regions (MSER's), which are regions in the image that remain connected over a margin (range) of greyscale intensity thresholds. The effect of thresholding the greyscale intensity values of an image was illustrated previously in figure 3.5 (pg. 112). As the greyscale intensity threshold is reduced blobs appear in the image. MSER's are blobs which remain stable with respect to shape and size over a margin of thresholds and can be very small or large regions in the image. As the threshold is varied from high to low the blobs are light regions in the image (MSER+), and as the threshold is varied from low to high the blobs are dark regions in the image (MSER-). A convex hull (ellipse) is fitted to each MSER which defines the scale and affine invariant support region for the keypoint whose position is located at the centroid of the ellipse. Although the original MSER algorithm was designed for use with greyscale images, the concept has been extended to MSER detection in colour images using each of the red, green and blue colour channels by Forssén [75].

### 3.3.3 Keypoint Descriptors

A keypoint descriptor encodes the local image content within a keypoint's support region. They are used to assess similarity between keypoints in different images which is used to find correspondences via matching. Descriptors are also used for many visual place recognition algorithms which will be discussed in chapter 5. Although

some descriptors use colour information [230, 21, 167, 187], they most frequently use information derived from the greyscale image function. Distribution based descriptors operating on greyscale images are used extensively for both scale and scale and affine invariant keypoint description. They describe the local image content using histograms, popular choices being the SIFT descriptor of Lowe [142], Gradient Location Orientated Histogram (GLOH) of Mikolajczyk and Schmid [126] and the SURF descriptor of Bay et al [17], each detailed in this section. The popularity of these methods is well founded due to their performance in comparative works.

SIFT was found to perform best in the comparison in [155] with respect to a range of greyscale descriptors including steerable filters, differential invariants, moments, complex filters and cross correlation. These descriptors were generated using a variety of keypoints (Harris, Harris-Laplace, DoG, Harris-Affine) for viewpoint changes between planar images subject to changes in image rotation, scale, affine transforms and illumination changes. The performance metric used was Receiver Operator Characteristics (ROC), a measure of the detection rate of correct keypoint correspondences versus false detection rate for varying thresholds on the distance between descriptors (Euclidean for SIFT, Mahalanobis distance for the rest). The ROC statistics were found by comparing an image to all others in a database. Correct correspondences were found between different images based on the position and shape of the keypoint support regions and the known homography between the images.

A more extensive comparison was made in [161] with additional descriptors including PCA-SIFT [120] shape context and spin-images — a brief summary of each can be found in [161]. The GLOH descriptor developed in [161] was also compared to the others. Recall vs 1-precision statistics were used to assess the performance of each descriptor with respect to their ability to match correctly keypoints between images detected using a range of algorithms including Harris-Affine and Hessian-Affine. Recall is the ratio of the number of correct correspondences versus the total number of correct correspondences for changing threshold on descriptor similarity. The metric 1-precision is the ratio of the number of false correspondences versus the number of total correspondences. They used again planar scenes and different image transformations including changes in scale, rotation, viewpoint, image blur, JPEG compression and illumination. The same method used in [155] was used to validate correct correspondences. GLOH was found to give the best performance followed closely by SIFT.

Although similar detailed comparisons have not been made with SURF descriptors, Bay et al [17] reported improvements over SIFT and GLOH descriptors evaluated for

SURF keypoints using a subset of the dataset and same experimental procedure of Mikolajczyk et al in [161].

Before discussing the details of SIFT, GLOH and SURF, each could be applied to any scale or scale and affine invariant keypoint. The size and shape of this support region is dependent on the keypoint detection algorithm used. The support region for scale-invariant keypoints such as SIFT and SURF is a circle whose size is set proportional to the characteristic scale  $\sigma$  of the keypoint. For scale and affine-invariant keypoints such as Harris-Laplace, Hessian-Laplace, and MSER, the support region is an ellipse. For the scale and affine invariant keypoints the local image content within the support region is typically mapped to a fixed sized patch by an affine transform, as illustrated in figure 3.15a for two MSER keypoints. Although the conversion to a fixed sized patch is not always required when computing descriptors for some scale-invariant keypoints, this step is sometimes used. When this is the case, the mapping to the fixed sized patch is a simple rescaling of the local image content as shown in figure 3.15b for two SIFT keypoints. A fixed sized patch of size  $41 \times 41$  pixels is frequently used [120, 155, 161].

### 3.3.3.1 SIFT descriptor

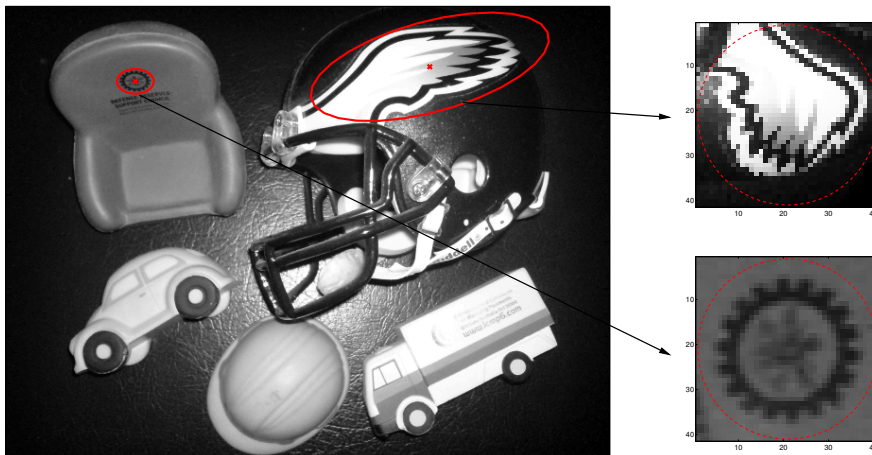
As discussed, SIFT includes a means for scale-invariant keypoint detection and description. For a SIFT keypoint detected in the DoG image  $D(\cdot; \sigma)$ , the gradient magnitude  $m$  and orientation  $\theta$  are calculated at any given pixel location  $\mathbf{u} = (u, v)^T$  in the scale space image  $L(\cdot; \sigma)$  as

$$m(u, v) = \sqrt{(L(u+1, v) - L(u-1, v))^2 + (L(u, v+1) - L(u, v-1))^2}, \quad (3.71)$$

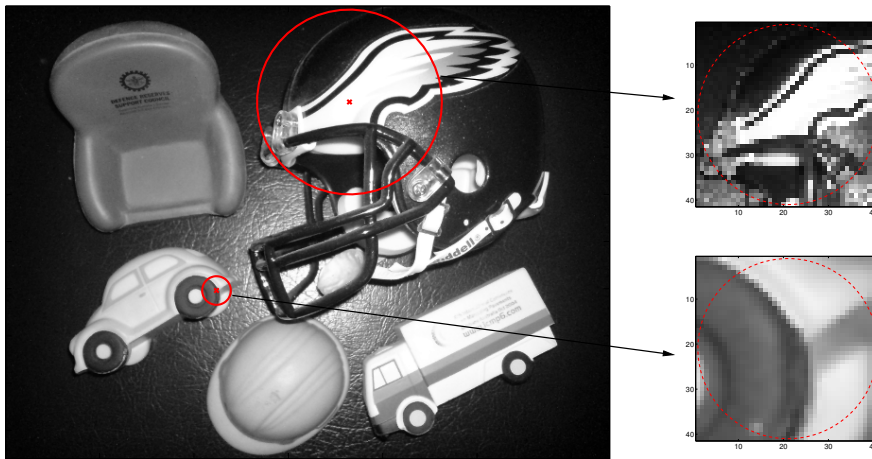
$$\theta(u, v) = \arctan \left( \frac{L(u, v+1) - L(u, v-1)}{L(u+1, v) - L(u-1, v)} \right). \quad (3.72)$$

A gradient orientation histogram is then constructed which contains 36 bins with angles spanning 360 degrees. Each pixel within the local region surrounding the keypoint contributes to the histogram, where each sample is weighted by its gradient magnitude and distance from the keypoint by a Gaussian function with standard deviation equal to 1.5 times the characteristic scale  $\sigma$  of the keypoint. The peak of the histogram defines the keypoint orientation which is used to make the descriptor rotationally invariant. A keypoint can have multiple orientations if any other non-adjacent peaks are within 80 percent of the maxima. If this occurs a separate descriptor is obtained for each orientation. A parabolic interpolation of the orientation histogram is used to counterfeit





(a) MSER keypoints (scale and affine-invariant).



(b) SIFT keypoints (scale-invariant).

Figure 3.15: The local image content within a keypoint's support region is frequently mapped to a fixed sized patch from which the keypoint's descriptor is evaluated. For scale and affine invariant keypoints such as MSER (a), this mapping is an affine transform. For scale invariant keypoint such as SIFT (b), it is a rescaling.

quantisation errors in the estimate of the keypoint orientations(s).

A SIFT descriptor is found for each keypoint orientation using the procedure illustrated in figure 3.16. The local region surrounding a keypoint is divided up into a  $4 \times 4$  array of cells centred at the position of the keypoint and aligned with the keypoint orientation. A gradient orientation histogram is found for each cell in the array, each histogram having 8 bins spanning angles from 0 to 360 degrees. Each pixel contributes as a sample value its gradient magnitude weighted by its distance from the keypoint by a Gaussian with standard deviation set proportional to the characteristic scale of the keypoint — the gradient orientation is found with respect to the keypoint orientation. Each sample contributes to the histograms in the nearest 4 cells using a linear inter-



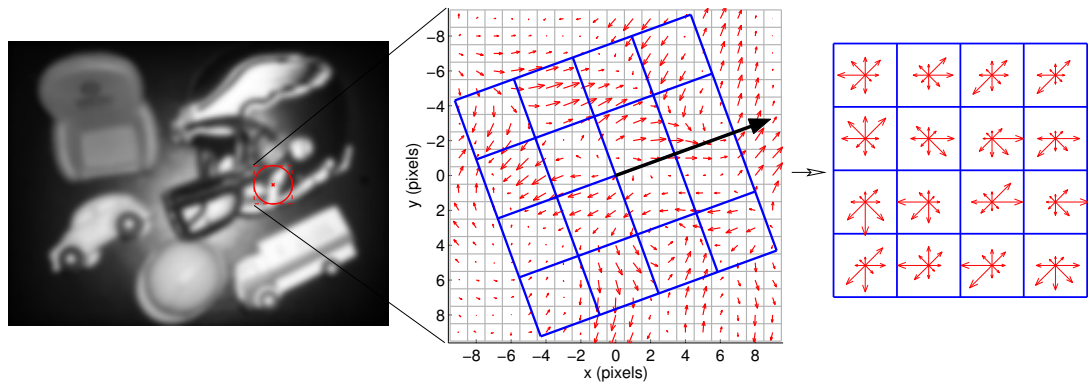


Figure 3.16: The general procedure used to evaluate a SIFT descriptor. The local region within a keypoint’s support region is divided up into a  $4 \times 4$  index cell array aligned with orientation of the keypoint (black arrow). An 8 bin gradient orientation histogram is found in each cell (right figure) from the gradient magnitudes and orientations (red arrows in middle figure) computed for each pixel within the support region. The 128 element SIFT descriptor is the concatenation of the histograms in each cell.

polarization to improve robustness to errors in keypoint scale and location and projective variations in the local image content. Each sample added to a histogram is also linearly interpolated to reduce quantisation errors. Concatenating the  $4 \times 4$  array of 8 element histograms yields the 128 element SIFT descriptor. To account for affine illumination variations, the descriptor is normalised to unit length. These illumination changes affect all pixels uniformly so normalisation to unit length accounts for this constant multiplication of the gradients as they are found from pixel differences. Non-uniform illumination is more problematic and can cause saturation. In an attempt to account for this, after the first normalisation any elements with values greater than 0.2 (empirical) are set to 0.2 and the vector re-normalised to unit length. The SIFT descriptor can also be computed for a keypoint mapped to a fixed sized patch. In such a case, the width of the cells and the standard deviation of the Gaussian functions used for distance weighting are set relative to the size of the fixed sized patch.

### 3.3.3.2 Gradient Location-Orientation Histogram (GLOH)

Mikalajczyk and Schmid [161] proposed a modified SIFT descriptor using a log-polar index bin array as shown in figure 3.17. Their descriptor is called GLOH, and acronym for Gradient Location-Orientation Histogram which they have used for keypoint description with Hessian-Affine and Harris-Affine keypoints, converting first the local support region to a fixed sized  $41 \times 41$  patch — see for example figure 3.15a. The log-polar array is divided into 17 index cells using radii of 6, 11, and 15 pixels and 8 uniform orientations (the middle cell is not divided). This log-polar array is then orientated with

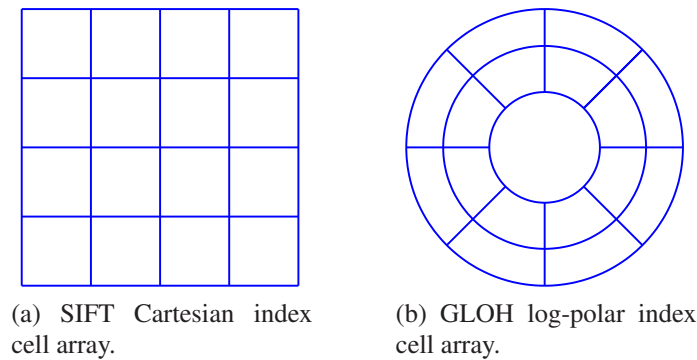


Figure 3.17: In contrast to SIFT which uses a Cartesian index cell array, GLOH uses a log-polar index cell array.

respect to the keypoint orientation which is found using the same method as SIFT. A gradient orientation histogram is found in each cell using the same method as SIFT, however, GLOH uses 16 orientation bins for each histogram. Concatenating all of the 16 bin histograms in each of the 17 cells gives a 272 bin histogram. The size of this histogram is reduced using principal component analysis where the covariance matrix is learned offline using training images. The 128 eigenvectors corresponding to the 128 largest eigenvalues of the covariance matrix are selected as the patch eigenspace. Projecting the 272 bin histogram onto the patch eigenspace produces the 128 element GLOH descriptor, which is the same length as the SIFT descriptor. Although GLOH was found to perform better than SIFT in [161], dimensionality reduction using PCA adds increased computational expense.

### 3.3.3.3 SURF descriptor

As is the case for SIFT, SURF includes both a method of keypoint detection and description [19, 17]. The SURF descriptor shares similarities with both the SIFT and GLOH descriptors as it is based on the distribution of intensity change within a keypoint's support region. However, SURF uses the distributions of first order Haar wavelet responses rather than gradients as they are efficient to compute using the same integral images used during SURF keypoint detection. Figure 3.18 illustrates two Haar wavelets used to compute a response  $d_u$  and  $d_v$  in the  $u, v$  coordinates respectively.

The SURF descriptor finds first an orientation for a keypoint having some characteristic scale  $\sigma$ . The first order Haar wavelet responses  $d_u, d_v$  are found within the  $6\sigma$  region surrounding the keypoint at sample points on a Cartesian grid spaced  $\sigma$  apart. The size of the Haar wavelet is  $4\sigma$ . These responses  $d_u, d_v$  are then weighted with a sampled Gaussian, centred at the keypoint position, with standard deviation  $2\sigma$ . A

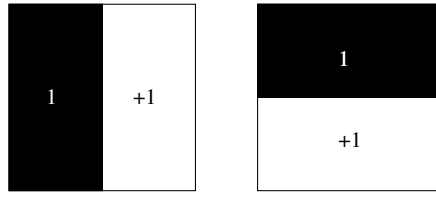


Figure 3.18: First order Haar wavelets used to compute a response  $d_u$  (left) and  $d_v$  (right) in orthogonal directions.

sliding circular window is rotated about the keypoint position with an angular range of  $\pi/3$  radians. The sum of responses  $d_u, d_v$  within the window at any position defines an orientation vector. The keypoint orientation is determined by the longest of these vectors.

Given the keypoint orientation, an equally space  $4 \times 4$  cell index array (figure 3.17a) centred at the keypoint position is constructed and aligned with the keypoint orientation. The outermost square of this cell array has width  $20\sigma$ . The Haar wavelet responses  $d_x$  and  $d_y$  relative to the keypoint orientation are then found at  $5 \times 5$  equally spaced sample points on a Cartesian grid in each of the cells using first order Haar wavelets of width  $2\sigma$ . These responses are then weighted based on their distance from the keypoint with a sampled Gaussian of standard deviation  $3.3\sigma$ . A 4 element vector  $\mathbf{v}$  for each cell is found as

$$\mathbf{v} = [\Sigma d_u, \Sigma d_v, \Sigma |d_u|, \Sigma |d_v|] \quad (3.73)$$

where  $|d_u|$  and  $|d_v|$  are the absolute values of  $d_u$  and  $d_v$  respectively. Concatenating the vectors for all cells produces the 64 element SURF descriptor which is normalised to a unit vector for invariance to image contrast.

An alternate version of the SURF descriptor was also considered in the same work [17]. By finding the summations of the responses  $d_u$  and  $|d_u|$  separately for  $d_v < 0$  and  $d_v \geq 0$ , and the summations of  $d_v$  and  $|d_v|$  separately for  $d_u < 0$  and  $d_u \geq 0$ , an 8 element vector for each cell is found. The resulting SURF descriptor is then 128 elements in length. The standard SURF descriptor is frequently referred to as SURF-64, and the extended version just described SURF-128. Although they observed in comparisons SURF-128 to be more distinctive, its increased length adds to computation time during keypoint matching. As part of their matching scheme, they index keypoints based on the sign of the Laplacian. A keypoint can only be matched to another if they have equal signed Laplacians — this relates to the non-enhancement of Local extrema principle in scale-space theory [134]. This indexing improves both matching speed and robustness to false positives [17].

### 3.3.4 Keypoint Matching

Keypoint matching is the process by which corresponding keypoints in different images are found. A keypoint is matched to another if their descriptors are sufficiently similar. Let  $\mathbf{a}$  and  $\mathbf{b}$  be any two  $n$  element column vector descriptors. A popular similarity score is the Euclidean distance  $d$  between descriptors

$$d = \sqrt{\sum_{i=1}^n (\mathbf{a}_i - \mathbf{b}_i)^2}, \quad (3.74)$$

although other similarity metrics based on Chi-squared ( $\chi^2$ ) statistics, Kullback-Liebler divergence and cosine angle between vectors can be used. Only corresponding entries in each vector are compared using each of these similarity metrics.

For histogram based descriptors there can be sources of errors due to quantisation of the bins. Similarity measures such as Earth Movers Distance (EMD) [197] and diffusion distance [139] consider inter bin correlations. EMD treats each element of the descriptor as a pile of dirt, the amount of dirt being proportional to the descriptor value. EMD defines the dissimilarity of two descriptors as the energy required to move the piles of dirt in one descriptor to resemble exactly the other, where the energy is the product of the amount of dirt moved by the distance. The diffusion distance proposed by Ling and Okada [139] has improved computation efficiency compared to EMD. It defines first the difference between the descriptors  $\mathbf{d}_0 = \mathbf{a} - \mathbf{b}$  as an isolated heat field at time  $t = 0$ . The dissimilarity of descriptors is the integral

$$d = \int_0^{\hat{t}} (\mathbf{d}_t)_{L1} dt, \quad (3.75)$$

where  $(\mathbf{d}_t)_{L1}$  is the  $L1$  norm of the diffused heat field  $\mathbf{d}$  at time  $t$ . However, EMD and diffusion distance are not ideally suited for descriptors such as SIFT, which is the concatenations of several histograms, as the dissimilarity measure would be dependent on the order in which the histograms in each cell are concatenated.

The elements of some descriptor are different measurable properties of the information content within a keypoint's support region. Each element of the SIFT descriptor for example is the same measurable property (i.e. the first order gradient magnitudes). A descriptor using steerable filters for example could contain information including the first, second and higher order greyscale intensity gradients. In such cases the Mahalanobis distance is a more suitable metric compared to Euclidean distance for example. Assuming that the covariance matrix  $\Sigma$  of descriptor values has been learned offline,

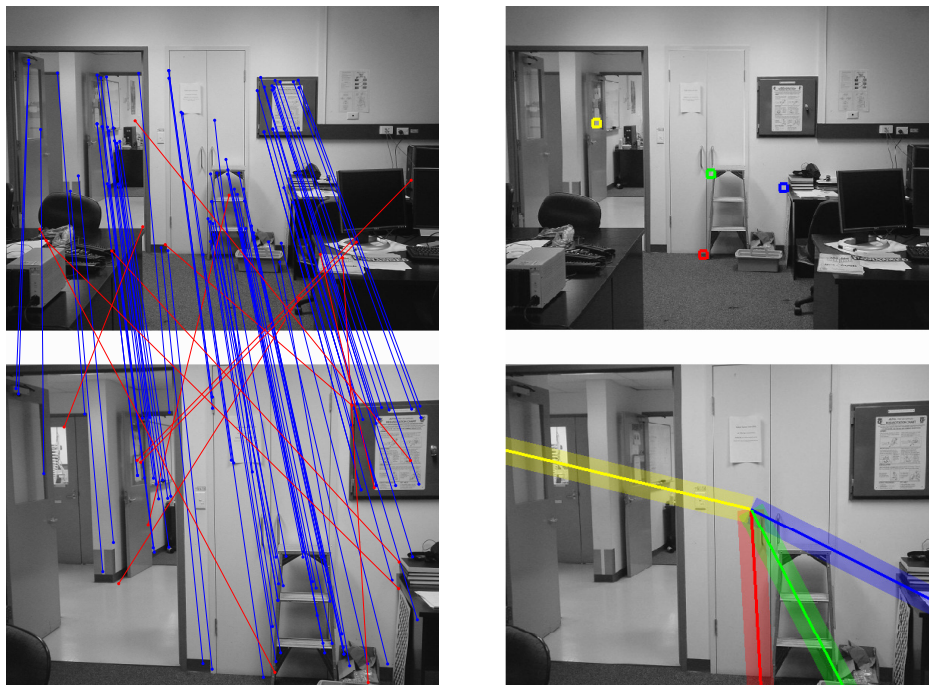
the Mahalanobis distance is

$$d = \sqrt{(\mathbf{a} - \mathbf{b})^T \Sigma^{-1} (\mathbf{a} - \mathbf{b})}. \quad (3.76)$$

Irrespective of the distance measure used, some threshold needs to be used to decide if a corresponding keypoint has been found. Thresholding Euclidean distance for example becomes problematic in the case where the scene contains repeatable patterns. Keypoints may be found on the upper left corner of all windows given an image of a building for example, where the appearance of the local region, and hence the descriptor, for each keypoint is very similar. A keypoint in one image can therefore be very similar to many others in another image, and this makes it difficult to determine with a high degree of certainty if a correct keypoint correspondence has been found. An approach that has become popular is to use an ambiguity metric to measure keypoint similarity. For a keypoint in one image, the nearest two keypoints in the other image are found based on the Euclidean distances between their descriptors, where the nearest of these is taken to be the potential keypoint correspondence. The ambiguity (dissimilarity) of this potentially corresponding pair of keypoints is the ratio of smallest to largest of these two distances. This approach has been used by Baumberg [16], who state that in their experience, the ambiguity of a match is more important than the matching strength based solely on the original similarity measure (e.g. Euclidean distance) for the purposes of finding sparse optical flow used for egomotion estimation. The use of the ambiguity metric based on Euclidean distance has also been suggested by Lowe [142] as the ideal approach when using SIFT keypoints and descriptors, and by Bay et al [19, 17] when using SURF keypoints and descriptors. Additionally, a mutual consistency check can also be used to improve the robustness of matching [181], whereby a keypoint match is accepted only if keypoint 1 is most similar to keypoint 2 in the other image compared to all others and vice-versa.

Keypoint correspondences found between two different images can be used to estimate the epipolar geometry between the views from which camera egomotion and the position of scene points can be recovered. Although a more detailed discussion regarding epipolar geometry is reserved for chapter 5, it is described by the fundamental matrix  $F$  for uncalibrated cameras, and the essential matrix  $E$  for calibrated cameras. The accuracy of the estimates of  $F$  and  $E$  are affected by the presence of incorrect correspondences. To obtain an accurate estimate for  $F$  or  $E$  in the presence of outliers, RANSAC (RANdom SAmple Consensus) [70] is typically used. A random set of correspondences is selected and  $F$  or  $E$  estimated — the number of correspondences selected is the minimum number required to estimate  $F$  or  $E$ . The remaining set of

correspondences which agree with the estimate  $F$  or  $E$  are then found. A new estimate for  $F$  or  $E$  is then estimated and the process iterated. Figure 3.19a illustrates, for a set of initial correspondences, those found to be correct (blue) and those found to be incorrect (red) using RANSAC and epipolar constraints. The reason for discussing here epipolar constraints is the fact that it can be used for guided matching. With respect to figure 3.19b, the fundamental matrix  $F$  found defines a mapping for each point in the first perspective image to an epipolar line in the second perspective image. Each point in image 2 maps also to an epipolar line in image 1. After an estimate of the epipolar geometry between views has been found, a guided matching process can be used. For a keypoint in image 1, only those keypoints in the second image within the vicinity of its epipolar line in the second image would be considered as potential candidates.



(a) Correct (blue) and incorrect (red) correspondences.

(b) Epipolar constraints. Each keypoint in image 1 is constrained to lie on its epipolar line in the second image.

Figure 3.19: The epipolar geometry between views can be used for guided matching. A keypoint in image 1 is constrained to lie on its corresponding epipolar line in the second image. For a keypoint in image 1, only those keypoint in the second image near its epipolar line in the second image would be considered as potential candidates.

### 3.3.5 Suitability for vision-based localisation

From the range of wide-baseline keypoint detection, description and matching algorithms discussed, it is of interest to identify those most suited for vision based local-



isation applications. This section reviews a number of comparative works that have compared primarily the relative performance of keypoint detection algorithms.

Mikolajczyk et al compared a number of wide-baseline keypoint detection algorithms in [160]. They used as a performance metric the repeatability score between image pairs introduced in [203] for image pairs subject to changes in zoom (scale and rotation), viewpoint, illumination and jpeg compression. The repeatability score is defined as the ratio of the number of correct correspondences and the minimum of the number of keypoints detected in the images. Images of either planar scenes or 3D scenes with fixed camera position were used such that all image pairs were related by a homography. Using a homography allowed the position and support region of a keypoint in image 1 to be transferred to the second image — both keypoints are in the same frame of reference. A correspondence was considered correct if, in the same frame of reference, the Euclidean distance between keypoint positions was within a specified tolerance and the overlap of keypoint support regions was within some threshold. A number of scale invariant detectors were compared, including the automatic scale selection methods of Lindeberg [136, 137] using the scale-normalised Laplacian and Hessian, difference of Gaussian (DoG) proposed by Lowe [143]<sup>2</sup>, and the Harris-Laplace algorithm. Additionally, the scale and affine invariant Harris-Affine algorithm was also used. For images subject to change in scale and rotation, overall all the scale-invariant algorithms outperformed the Harris-Affine keypoint detector, with Harris-Laplace having the best performance followed closely by Hessian. The Laplacian and DoG algorithms were found to provide similar performance. This is expected as the DoG function is an approximation of the Laplacian of Gaussian function. Both the Laplacian and DoG algorithms include no means for edge removal which limits their performance with respect to Hessian-Laplace for example. They attributed the relatively poor performance of Harris-Affine to the fact that it is designed to handle complex image transformations. In contrast, the scale-invariant algorithms are designed specifically to handle scale change only. For the case of large viewpoint change, Harris-Affine and Harris-Laplace were found to have a similar performance for viewpoint changes up to 40 degrees. In excess of 40 degrees the performance of Harris-Laplace degraded much more significantly than Harris-Affine.

A comparison of scale and affine invariant keypoint detectors is given by Mikolajczyk et al in [156]. They used a similar experimental procedure to that in [160], measuring performance based on the repeatability of keypoints between planar or 3D

---

<sup>2</sup>This is the original version of Lowe, using simply the Difference of Gaussian images for keypoint detection in scale-space without keypoint interpolation or edge removal. A pyramid structure is used for efficient computation and not the octave based approach used in [142].

scenes related by a homography. A keypoint correspondence is considered correct if the overlap between the regions in the same frame of reference is within some threshold. Results were obtained for image transformations including changes in viewpoint, scale, blur, jpeg compression and illumination. The scale and affine invariant keypoint detectors compared were Harris-Affine, Hessian-Affine, the geometry-based and intensity-based methods of Tuytelaars and van Gool, MSER, and the affine scale-saliency method of Kadir and Brady. Overall, for many of the image sequences and transformations MSER was found to provide the best performance (with the exception of image blur) with affine scale-saliency ranking consistently low. However, with respect to the number of absolute correct correspondences between views, MSER was found to perform relatively poorly to many of the others. In contrast, Harris-Affine and Hessian-Affine ranked consistently high in performance and were able to find many more overall correct correspondences between views than MSER which is of benefit with respect to visual odometry applications. As a final note, as the change in viewpoint between images was increased, the performance of all keypoint detectors degraded at similar rates.

The comparisons of Mikolajczyk et al in [160, 156] use planar scenes and image pairs related by a homography. Fraundorfer and Bishop compare the relative performance of a number of wide-baseline keypoint detection algorithms using more complex non-planar scenes in [78]. As the images used were not related by a homography, they used as ground truth viewpoint constraints derived from image triplets (trifocal tensor). The same repeatability metric used in [160, 156] was used, where correct keypoint correspondences were identified based on the Euclidean distance between keypoint positions in the image plane and the overlap between keypoint support regions in the image plane (this requires transferring the positions and support regions of keypoints in one image to the other). The keypoint detectors compared included the standard Harris detector, MSER, DoG (version in [143]), Harris-Affine and Hessian-Affine. The first test used two sequences of images with viewpoint changes ranging from 0 to 90 degrees. The first sequence used predominantly planar objects (boxes on a turntable) and the second a complex scene (room). MSER was observed to give superior performance to other scale and affine invariant algorithms such as Harris-Laplace and Harris-Affine with the box sequence, although Harris-Affine and Hessian-Affine found many more correspondences. For the complex room sequence, the relative performance of MSER, Harris-Affine and Hessian Affine were similar with respect to repeatability. MSER found fewer correspondences again which supports the observations in [156]. The second experiments used larger sequences of images subject to arbitrary viewpoint change. MSER was observed to be superior to Harris-Affine and



Harris-Laplace which differs from the observations made in [156] and suggest there may be some bias using predominantly planar scenes related by a homography.

It is of benefit here to summarise the comparisons discussed. There are two distinctive wide-baseline keypoint detection methods; those which are scale-invariant, and those which are scale and affine invariant. Of the scale and affine invariant methods evaluated in [156, 78], MSER, Hessian-Laplace and Harris-Laplace ranked consistently well. Although MSER appears to be the most robust with respect to keypoint repeatability, an observation most evident in the comparison in [78], it is unable to find as many correct correspondences between views as Harris-Affine and Hessian-Affine. This is a disadvantage with respect to visual odometry applications where it is important to obtain a sufficiently large number of correspondences for accurate egomotion estimation. The relative results between the scale invariant and Harris-Laplace algorithm in [160] is of interest here. They observed that in general the scale invariant algorithms gave improved performance over Harris-Laplace for viewpoint changes less than 40 degrees. Even for wide-baseline visual odometry the viewpoint change between images is highly unlikely to exceed 40 degrees. Furthermore, the results for viewpoint change used images of planar scenes, and the scale and affine-invariant keypoint detection algorithms are designed specifically for this. Fraundorfer [78] for example found differing results for predominantly planar versus 3D scenes.

Moreels and Perona [169] also observed that the performance of scale and affine-invariant keypoint detectors such as Harris-Affine and Hessian-Affine (both coupled with various descriptors) deteriorated quickly for viewpoint changes in excess of 30 degrees for keypoint matching in 3D scenes from image databases. They used two performance criteria, both evaluated with respect to matching of keypoints in one image to a database of keypoints found in many other images. The first was the stability rate of a keypoint versus viewpoint change, where the stability is the fraction of keypoints in an image successfully matched. The second was Receiver Operating Characteristics (ROC), using as ground truth the epipolar geometry between triplets of calibrated images. In both cases, the similarity metric used during matching was the ambiguity ratio of Euclidean and Mahalanobis distances. As noted by Lowe [142], one limitation of the scale and affine-invariant methods is the fact that they are not truly affine invariant. Harris-Affine for example selects the approximate scale and position of a keypoint in a non-affine manner using Harris-Laplace and then searches locally in affine-scale space. Lowe also argues that the affine frames used during detection are more sensitive to image noise which limits their repeatability with respect to the scale-invariant algorithms. This is evident in the results of Mikolajczyk et al in [160] where Harris-Laplace gave improved performance compared to Harris-Affine for viewpoint changes

less than 40 degrees. The scale-invariant methods can therefore be considered then as suitable choices for general use.

Unfortunately, the comparison of scale-invariant keypoint detectors in [160] does not include the SIFT detector or Fast-Hessian detector used by SURF. A difference of Gaussian (DoG) method is compared, but does not include the keypoint interpolation or edge removal schemes used by SIFT in [142]. A comparison of the SIFT and Fast Hessian (FH-9 and FH-15) keypoint detectors as well as Harris-Laplace and Hessian-Laplace is presented by Bay et al in [17]. The same experimental methodology, repeatability measure and image sequences used in [160] were used. For change in image viewpoint, FH-15 was found to show some improvements over the other methods for one of the two the image sequences used. For changes in both image scale and blur, FH-15 in general outperformed the other keypoint detectors. For changes in viewpoint and scale between images, FH-9, SIFT and Hessian-Laplace gave comparable performance. Harris-Laplace was observed to perform poorly compared to the others for all image transformations (viewpoint, scale and blur). In the same work the accuracy of keypoint localisation for each of the detectors were also compared with respect to calibration and 3D scene reconstruction. The relative rankings were FH-15, FH-9, SIFT, Harris-Laplace and Hessian-Laplace. This results can be explained by the fact that, unlike the other detectors, fast Hessian and SIFT both interpolate the position and scale of keypoints. SIFT uses an octave based approach to image processing which limits the accuracy of interpolation for keypoints detected at high octaves. Fast Hessian on the other hand does not suffer from this problem.

The full versions of SIFT and SURF both include a method of keypoint detection and description, and a relative comparison of these full versions was presented by Bauer et al in [15]. Although several implementations of SIFT were used, the results discussed here refer to the binary version of Lowe [142]<sup>3</sup>. Two versions of SURF were used, one using the original sized images (SURF) and the other using the double sized images (SURF-d). They compared the performance with respect to the total number and ratio of correct correspondences found between image pairs using the Euclidean distance between descriptors for matching. These image pairs were taken from a small sequences of images in outdoors scenes subject to changes in rotation, scale, image noise, lighting and viewpoint. Overall, SIFT was found to give slightly better performance compared to both SURF and SURF-d with respect to the ratio of correct correspondences and the total number of correct correspondences. However, the authors consider both version of SURF to be superior to SIFT as they they have

---

<sup>3</sup>Available <http://people.cs.ubc.ca/~lowe/keypoints/>

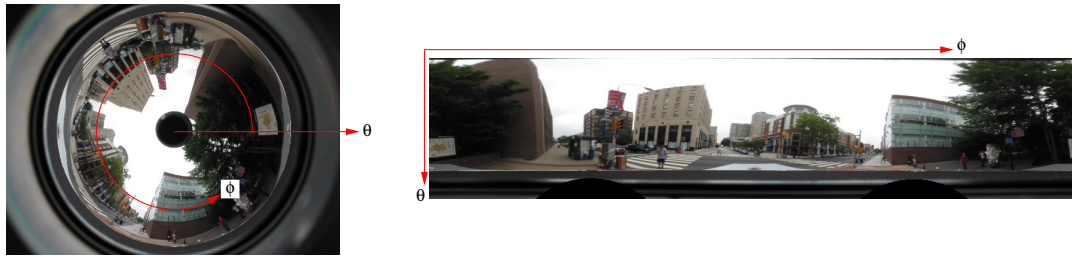


Figure 3.20: A wide-angle catadioptric image converted to a log-polar panoramic image.  $\theta$  is an angle of colatitude and  $\phi$  and angle of longitude (see figure 2.4, pg. 36).

considerably better runtime efficiency. They argue that this is an acceptable tradeoff for a reduction in both the ratio and number of correct correspondences compared to SIFT.

The relative performance of the full versions of SIFT and SURF was also compared by Valgren and Achim in [233]. The experiments used 7 wide-angle catadioptric image sequences taken in a university campus at different periods of the year. Each image was first converted to a log-polar panoramic image, as illustrated in figure 3.20. SIFT and SURF keypoints and descriptors were then found in each of the log-polar panoramic images. The 64 element and 128 element SURF descriptors were found for each of the SURF keypoints (SURF-64 and SURF-128). In the first experiments, one of the sequences was taken as the reference sequence. Then for each image in all the sequences, the most similar image in the reference sequences was found, the most similar image being that with the maximum number of keypoint correspondences found using the Euclidean ambiguity metric for the descriptors and the matching thresholds recommended for SIFT by Lowe [142] and for SURF by Bay et al [17]. By using the odometry logged for each sequence to validate if a correct image had been retrieved, SURF-128 was shown overall to provide the best performance followed by SIFT. The second experiment selected, in each sequence, images taken at the same location but different viewpoints. The relative performance was judged based on the ratio of correct correspondences and total number of correspondences found. SURF-128 was found again to give the best performance, followed by SURF-64. However, when compared to SURF-128 and SURF-64, SIFT was in general able to find at least twice as many correct correspondences.

Without giving an exhaustive discussion of all the comparisons of the full versions of SIFT and SURF presented in the literature, the general consensus is that both give similar performance in many applications with respect to metrics such as repeatability of keypoints, recall versus 1-precision and database image retrieval [17, 15, 233, 172]. SIFT in general detects more keypoints in an image which allows more correspon-

dences to be found between image pairs. This is an advantage for visual odometry and structure from motion applications as the accuracy of each in general improves with the number of correspondences used. However, SURF has a distinct advantage with respect to computation time. Although there is no way to answer exactly which is the best, any of the scale-invariant methods based on the scale-space framework can detect and match keypoints between images separated by a wide-baseline change in pose (i.e. Harris-Laplace, Hessian-Laplace, SIFT and SURF). With respect to descriptors, the selection is dependent to some extent on the keypoint detector used. Bay et al [17] for example found the relative ranking of descriptors for SURF keypoints in their experiments to be SURF, SIFT then GLOH. This contradicts the results of Mikolajczyk et al [161] who found in their experiments the GLOH descriptor to outperform the SIFT descriptor using scale and affine-invariant keypoints. If either SIFT or SURF were used for keypoint detection, then it would be logical to use the descriptor designed specifically for each. In addition, if the full versions of SIFT or SURF were used for keypoint detection and description, then it would be logical to use the ambiguity metric for keypoint matching as recommended by Lowe (SIFT) [142] and Bay et al (SURF) [17].

### 3.4 Wide-baseline Keypoint Matching with Wide-angle Images

As discussed in chapter 1, the ability to find keypoint correspondences between wide-angle images separated by a wide-baseline change in camera pose has potential advantages for vision-based localisation. The key to most wide-baseline keypoint matching algorithms is use of scale, or scale and affine invariant keypoint detection and description algorithms — for convenience, these will simply be referred to as wide-baseline keypoint detection and description algorithms. Unfortunately, the state of the art wide-baseline keypoint detection and description algorithms such as SIFT and SURF are designed for use with perspective images. Considering that wide-angle images are characterised as having extreme radial distortion, it seems intuitive that some account should be made for this distortion during keypoint detection and description. This raises the question of how the radial distortion should be handled.

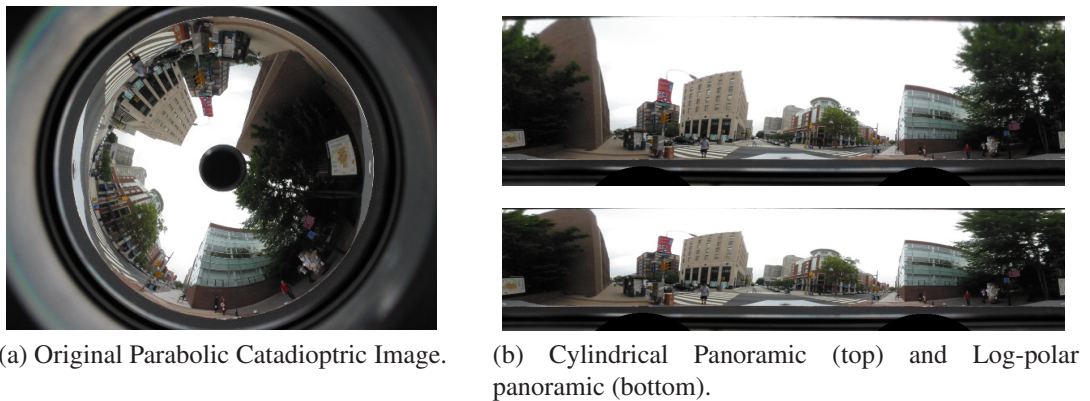


Figure 3.21: ‘Rectified’ log-polar and cylindrical panoramic images produced from a calibrated parabolic catadioptric camera.

### 3.4.1 Existing Approaches

The most basic approach to wide-angle image processing is to apply directly existing image processing algorithms to wide-angle images. This can be considered as a ‘blind’ application as no attempt is made to account for the radial distortion during keypoint detection or description. Classical algorithms such as KLT have been applied blindly to wide-angle images obtained with catadioptric cameras [48, 89, 234] and fisheye cameras [213, 102, 57]. Wide-baseline keypoint detection and description algorithms have also been applied blindly to wide-angle images. Scaramuzza et al for example applied blindly SIFT to catadioptric images [200, 198], and they noted that the number of false positive correspondences found between catadioptric images pairs exceeded the number typically found using perspective images [200].

Another approach used is to apply existing algorithms to *rectified* log-polar or cylindrical panoramic images. This approach is used almost exclusively for downward facing catadioptric cameras and rarely used with fisheye cameras. An example log-polar panoramic image was show previously in figure 3.20. A cylindrical panoramic image is obtained as the perspective projection of the image on the view sphere to a cylinder wrapped around the view sphere. It appears similar to a log-polar panoramic image as shown in figure 3.21. Examples of classical methods such as KLT applied to rectified panoramic can be found in [125, 32]. Examples of wide-baseline algorithms applied to rectified panoramic images can be found for SIFT in [233, 172] and for SURF in [233, 172]. Rectified panoramic images such as those shown in figure 3.21b still contain some distortion from the ideal (non-deformed) perspective projection. Converting any wide-angle image to a log-polar or cylindrical panoramic image is simply a mapping from one deformed space to another.



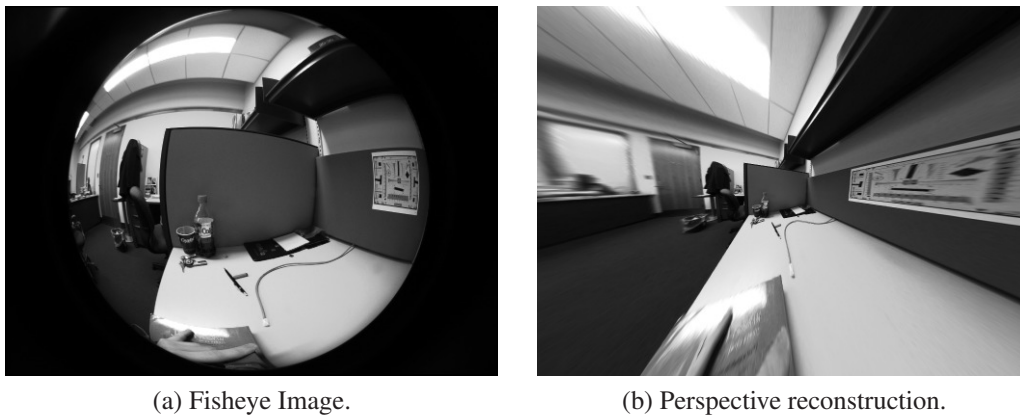


Figure 3.22: A fisheye image converted to a perspective image of the same size. Observe that information near the centre of the fisheye image is lost. Information near the periphery of the fisheye image becomes stretched out via interpolation with introduces artifacts in the perspective image.

Recall that any calibrated wide-angle image can be mapped via the sphere to the perspective plane. It seems a valid choice to then convert any wide-angle image to a perspective image and apply existing wide-baseline keypoint detection and description algorithms to this perspective image. However, Daniilidis et al [56] note several limitations of this approach. Firstly, perspective projection is limited to less than a hemispherical field of view making this approach unsuited to omnidirectional wide-angle cameras. Secondly, converting a wide-angle image to a perspective image is computationally expensive and introduces severe artifacts due to interpolation of the wide-angle image values. This is illustrated in figure 3.22 which shows a fisheye image converted to the perspective image. To include most of the cameras field of view, the pixels near the periphery of the fisheye image are effectively stretched out, and those near the centre of the image compressed. If the perspective image has the same number of pixels as the wide-angle image, then much of the information near the centre of the fisheye image is annihilated, and information near the periphery of the wide-angle image is artificially created through interpolation. Daniilidis et al [56] argue that image processing algorithms should always be applied to the original image values. This suggests that, to account for image distortion, some specialist approach to image processing designed specifically for wide-angle images is required.

It should be noted here that the discussions regarding alternate image processing methods are limited to *single image* methods. For example, a wide-angle image could be converted to multiple narrow angle of view perspective images, each covering different regions in the scene. Doing this would eliminate the 180 degree angle of view constraint, and would also potentially limit interpolation artifacts. However, a more

suitable option would be to use a specialised camera system which consists of an array of multiple perspective cameras (these systems are commercially available). This option is more suitable as no interpolation is required. As mentioned, the thesis is focussed on single image methods, in which case converting a highly distorted wide-angle image to a single perspective view is problematic for the reasons discussed. In particular, even if specialised techniques could be used to minimise interpolation artifacts, there is no way to avoid the fact that the angle of view of a perspective image is limited to less than 180 degrees (i.e. limited to less than a full hemisphere).

### 3.4.2 Methods Designed for Wide-Angle Images

There are some examples in the literature of image processing algorithms designed for use with wide-angle images. Briggs et al [26] considered scale-invariant keypoint detection with wide-angle images. Their algorithm operates on a rectified panoramic image. However, they note that constructing a linear scale-space as convolution of the image with Gaussians of increasing scale is not correct as the image is not perspective. They choose to therefore first convert each rectified panoramic image to a one-dimensional image  $I(\phi)$  by averaging the pixel intensity values for each angle of longitude  $\phi$  over the middle rows of the rectified panoramic image. The scale-space  $L(\phi; \sigma)$  for the image  $I(\phi)$  is then obtained by convolution with a one-dimensional sampled Gaussian  $G(\phi; \sigma)$ :

$$L(\phi; \sigma) = I(\phi) * G(\phi; \sigma), \quad G(\phi; \sigma) = \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{\phi^2}{2\sigma^2}}. \quad (3.77)$$

This convolution ‘wraps around’ at the edges of the one-dimensional image  $I(\phi)$  at angles 0 and 360 degrees. Keypoints are found in the scale-space images using the difference of Gaussian (DoG) approach of Lowe [142], where the difference of Gaussian image is  $D(\phi; \sigma_i) = L(\phi; \sigma_{i+1}) - L(\phi; \sigma_i)$ . Keypoints are selected as local extrema in the 3x3 pixel neighbourhood in the set of DoG images. This local neighbourhood includes the adjacent pixels at the current scale as well as the three nearest pixels in each of the adjacent DoG images (similar to the neighbourhood in figure 3.8 for the DoG function restricted to one spatial dimension). The limitation of this method is the use of one-dimensional images which makes it most suited for planar camera motion. They argue that the use of one-dimensional images is suitable when using a downward facing catadioptric camera traversing through a flat indoor environment, where the environment contains minimal texture (e.g. corridor walls). However, the same is not true for outdoor environments which are highly textured and information rich. The

ability to capture an information rich representation of a scene is one of fundamental reasons why vision is used, and it is critical to the success of tasks such as visual place recognition. It can be concluded therefore that this is not the ideal approach.

As discussed, wide-baseline keypoint detection algorithms are designed almost exclusively for use with perspective cameras. The image processing algorithms are shift-invariant in the image plane, where the local intensity gradient of an image for example would be computed using the same derivative of Gaussian kernel at all positions in the image. However, the appearance of objects in a wide-angle image can change considerably depending on their position in the image due to the camera's radial distortion. Applying operators that are shift-invariant in the image plane is therefore not ideal. Daniilidis et al [56] noted that if a wide-angle image is mapped to the sphere, the appearance of objects in the scene in this spherical image will not be affected by camera distortion. Furthermore, if two wide-angle images taken by the same camera at different orientations are mapped to spherical images, in theory these spherical images differ only by a change in rotation (i.e. a rotational shift). This concept of rotational shift-invariance is illustrated using a practical example in figure 3.23. They therefore proposed that in order to account for image distortion, the ideal domain in which to formulate shift-invariant image processing algorithms is the sphere, where shift refers to a rotation. This concept is the inspiration for the methods of keypoint detection and description with wide-angle images developed in chapter 4.

Daniilidis et al [56] used their concept of image processing on the sphere to compute optical flow between catadioptric images separated by a small change in camera pose. They found their approach to give improved accuracy in egomotion estimates compared to the equivalent 'blind' method of image processing. Importantly, the image processing operations were formulated on the sphere and implemented on the wide-angle image itself which removes then need for any interpolation of the original image function. This process can be illustrated by considering their approach to Gaussian convolution which is illustrated in figure 3.24. They define first an equivalent Gaussian function  $G_S$  on the sphere as the stereographic projection of the two-dimensional Gaussian to the sphere. The function centred at the pole is defined in polar coordinates at angle of colatitude  $\theta$  and longitude  $\phi$  by

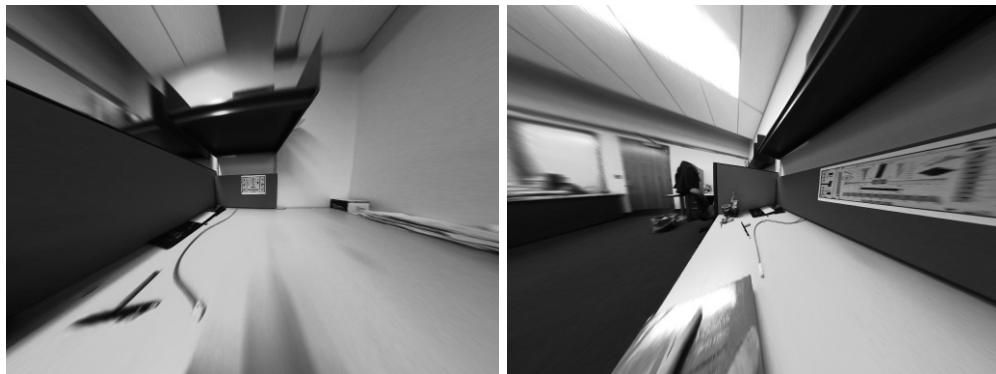
$$G_S(\theta, \phi) = \frac{1}{2\pi\sigma^2} e^{-1\frac{1}{2\sigma^2} \cot^2 \frac{\theta}{2}}. \quad (3.78)$$

The Gaussian smoothed image is defined as the convolution of the spherical image and  $G_S$ , where this convolution takes place on the sphere — a formal discussion regarding the convolution of two functions on the sphere is presented in chapter 4. This con-





(a) Original fisheye image plane



(b) Mapped to perspective image plane



(c) Mapped to the unit sphere

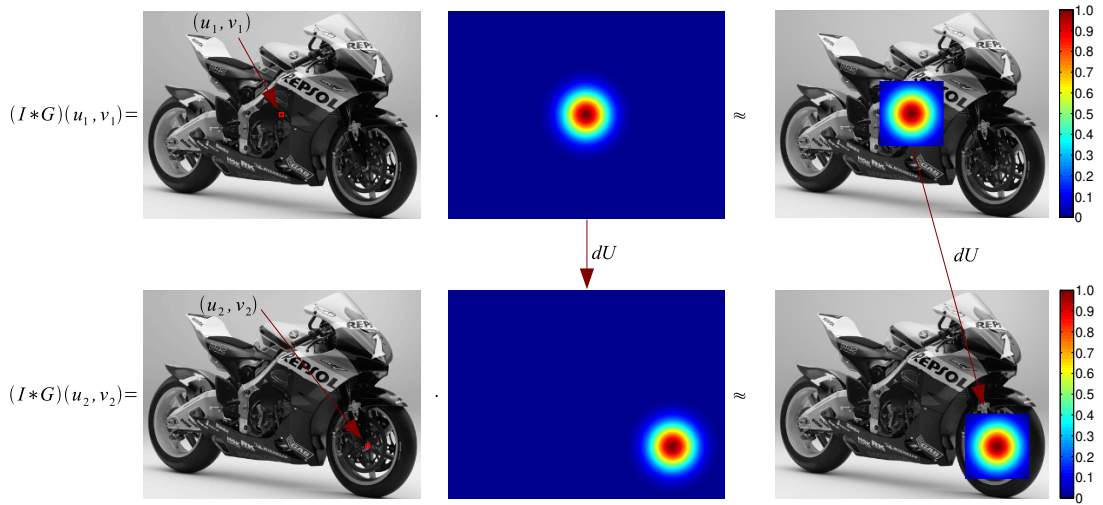
Figure 3.23: Rotational shift invariance for different image representations. (a) shows two fisheye images separated by a change in camera rotation. (b) shows these fisheye images converted to perspective images, and (c) shows these images mapped to the unit radius sphere (the spherical image on right has been rotated by a small angle). The spherical images are equivalent up to a change in rotation.

volution is implemented on the image itself, and is evaluated at a pixel position  $\mathbf{u}$  as follows. For a central projection camera, the image function at pixel location  $\mathbf{u}$  maps to a unique point  $\eta(\theta, \phi)$  on the view sphere.  $G_S$  is rotated so that it is centred at  $\eta$  and is then projected to a kernel on the wide-angle image. The result of convolution is the discrete summation of the pointwise product of this kernel and the wide-angle image values within this kernel. Observe in figure 3.24 that the shape of the kernel used to evaluate the convolution at two different positions in the image changes considerably — this kernel is not shift-invariant in the image plane.

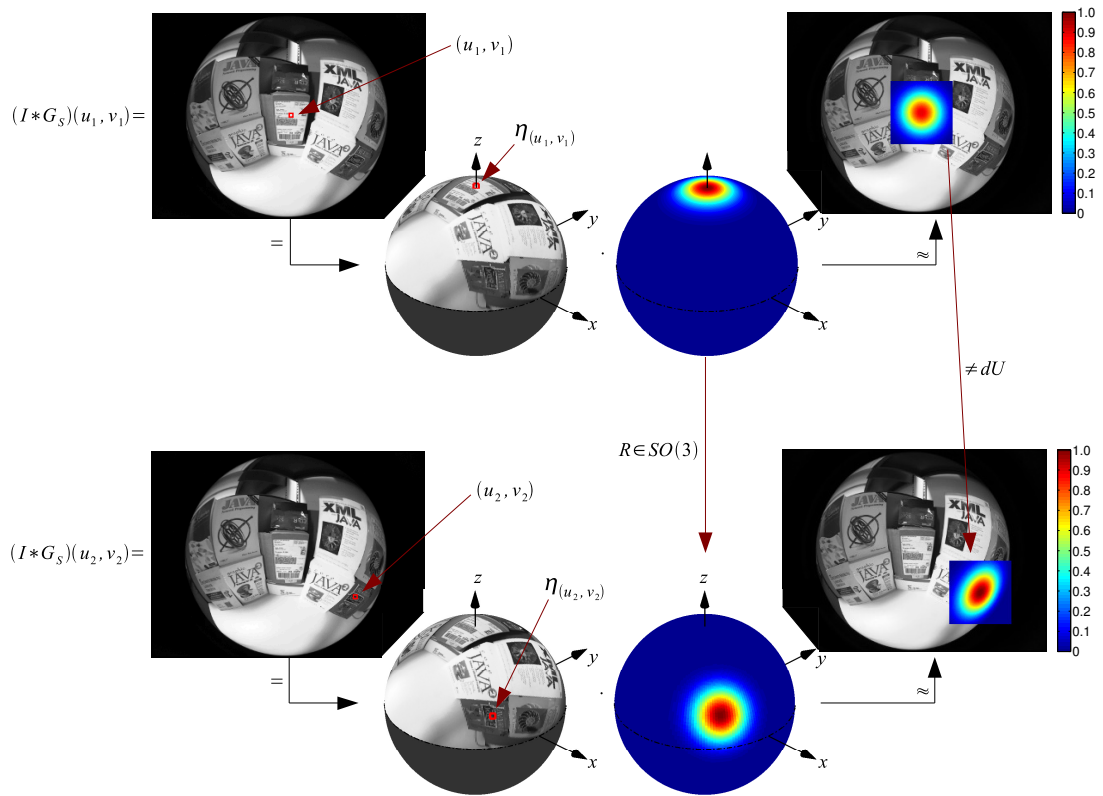
### 3.4.3 Proposed Approach to Wide-Baseline Matching with Wide-Angle Images

It is proposed that the general approach to image processing on the sphere suggested by Daniilidis et al [56] is ideal for wide-angle image processing. It accounts for the radial distortion in the image, image processing is shift-invariant with respect to rotations on the sphere, and it can be applied to any central projection camera assuming the camera intrinsic parameters can be calibrated. This general approach to image processing on the sphere has since been used for correspondenceless structure from motion in a series of works by Makadia et al [148, 147, 146].

Recall from previous discussions that methods of scale-invariant keypoint detection using the scale-space framework, for example SIFT, are ideal candidates for use with perspective images for vision-based localisation. It is logical to therefore consider reformulating these methods of image processing as operations on sphere. This is the approach taken in this work. The first step would be to construct a linear scale-space for an image mapped to the sphere, which requires a definition for scale-space on the sphere. Fortunately, Bülow derived the solution for the heat diffusion equation on the sphere in [31] and discussed its use for scale-space analysis of wide-angle images in [30]. The scale-space operator for functions on the sphere is the ‘spherical Gaussian’ [30], and the scale-space representation of a function on the sphere is the convolution of the function with the spherical Gaussian. The spherical Gaussian is an isotropic function on the sphere, and it is applied uniformly at all points on the sphere during convolution with a function on the sphere. These properties relate to scale-space axioms discussed in section 3.3.1 that were used to derive scale-space for two-dimensional signals, namely isotropy (the scale-space operator has no preferred direction in the domain of the function) and homogeneity (the scale-space operator is applied uniformly at all points in the domain of the function during convolution).



(a) Shift-invariant convolution in the image plane.



(b) Shift-invariant convolution on the unit sphere implemented on the image.

Figure 3.24: (a) Shift-invariant convolution in the image plane, and (b) shift-invariant convolution on the sphere implemented on the image. A pixel at position  $\mathbf{u} = (u, v)^T$  projects to a point  $\eta_{u,v}$  on the sphere. The functions  $G_S$  on the sphere in (b) are related by a rotation  $R \in SO(3)$ .

The work of Daniilidis et al [56] and Bülow [31, 30] has made it possible to explore suitable methods of wide-baseline keypoint detection and description with wide-angle images.

In the next chapter, two variants of SIFT are developed that are suited for scale-invariant keypoint detection and description with wide-angle images. Both reformulate SIFT as an image processing algorithm on the sphere, and they differ in their implementations. The keypoints found using these variants can be used to find correspondences in wide-angle images separated by a wide-baseline change in camera pose.

### 3.5 Conclusions

A review of some classical methods of keypoint detection, description, matching and registration were presented in this chapter that are used to find keypoint correspondences in different images separated by a small-baseline change in camera pose. The limitations of their use with images separated by a wide-baseline change in camera pose were discussed, in particular their inability to handle large projective changes between images. A review of methods suited for wide-baseline changes in camera pose was presented next. The key to these methods is the use of scale, and scale and affine invariant keypoint detection algorithms, including those using the scale-space framework and a number of alternatives. A summary of numerous comparative works was presented, where it was concluded that scale-invariant keypoint detection and description algorithms such as SIFT and SURF are ideal candidates for vision-based localisation applications. From the review of the comparative works, it was found that both perform similarly, however, in general SIFT is capable of finding more correspondences between images than SURF. Algorithms such as SIFT and SURF are designed for use with perspective images, and applying them directly to wide-angle images is not ideal as no account is made for the image distortion. The approach to wide-angle image processing proposed by Daniilidis et al [56] was identified as suitable means by which image processing algorithms can be designed for use with wide-angle images. They argue that to account for image distortion, image processing algorithms should be formulated as shift-invariant operations on the sphere. It was proposed that by using the general approach to wide-angle image processing developed by Daniilidis et al [56], and the foundations of scale-space analysis for functions on the sphere derived by Bülow [31, 30], SIFT could be reformulated as an image processing algorithm on the sphere. This would make it suitable for wide-baseline keypoint detection and de-

---

scription in wide-angle images. Chapter 4 develops two variants of SIFT based on this principle.



## Chapter 4

# Wide-Baseline Keypoint Detection, Description and Matching with Wide-Angle Images

*Two novel keypoint detectors named spherical SIFT (sSIFT) and parabolic SIFT (pSIFT) are developed in this chapter that are variants of SIFT designed for use with wide-angle images. Both define scale-space space for wide-angle images as the convolution of the image, mapped to the sphere, with the spherical Gaussian. sSIFT implements this convolution in the spherical Fourier domain, and pSIFT approximates this convolution using an efficient operation on the stereographic image plane. Both algorithms detect keypoints as local extrema in the difference of scale-space (difference of Gaussian) images and define the support region for a keypoint as a circle on the sphere, centred about the keypoint position on the sphere. The SIFT descriptor for a keypoint is evaluated from the image content within this support region. sSIFT and pSIFT are compared to SIFT in a number of experiments using real and synthetic wide-angle images.*

### 4.1 Introduction

As discussed in chapter 3, simply converting any wide-angle image to a single undistorted perspective image and applying existing image processing algorithms is problematic. Firstly, this approach is not suited for omnidirectional cameras with a field of view in excess of a full hemisphere — although this constraint can be avoided by

converting a wide-angle image to multiple perspective images, as discussed previously this work is focussed on single image processing. Secondly, the undistorted perspective image would contain artificially interpolated image values which, as argued by Daniilidis et al [56], have a negative effect during image processing. Applying blindly existing keypoint detection methods to wide-angle images without accounting for the radial distortion in the images is also problematic. The reason for this is that the appearance of the imaged region in the scene can appear very different depending on its position in the image as a result of the camera distortion. However, if the image obtained by any central projection camera is mapped to an image on the unit view sphere, the appearance of regions in the scene are unaffected by the camera distortion. It is only by reformulating image processing algorithms as (rotationally) shift invariant operations on the sphere that they are invariant to camera distortion and suited for use with central projection wide-angle images (i.e. images obtained with central projection wide-angle camera). This approach to wide-angle image processing was used by Daniilidis et al [56] for optical flow computation in catadioptric wide-angle images and is one of the inspirations for this work.

Two novel variants of the Scale-Invariant Feature Transform (SIFT) [142] are developed in this chapter, reformulated as image processing algorithms on the sphere, that are suited for scale-invariant keypoint detection and description in central projection wide-angle images. Both define scale-space for wide-angle images as the convolution of the image, mapped to the sphere, with the solution of the spherical heat diffusion equation (spherical Gaussian). This solution was derived by Bülow [31] whose work has made it possible to explore scale-space for wide-angle images. The first variant, termed spherical SIFT, implements this convolution in the spherical Fourier domain. The second variant, termed parabolic SIFT (pSIFT), approximates this convolution using an efficient operation on the stereographic image plane, and has similar computation expense to the standard SIFT algorithm. As will be discussed in section 4.4, this approximation, and the efficiency of its implementation, are related to an important property of stereographic projection; it is a conformal mapping which locally preserves shapes and distances when projecting functions from the sphere to the stereographic image plane. sSIFT and pSIFT both detect keypoints as local extrema in scale and space, using as a measure of saliency the difference of neighbouring scale-space (difference of Gaussian) images. For a given sSIFT or pSIFT keypoint, the keypoint support region is defined as a circle on the sphere that is centred at the position of the keypoint on the sphere. The SIFT descriptor is then evaluated from the local image content within this support region.

This remainder of this chapter is organised as follows. Section 4.2 discusses scale-



space for wide-angle images and the solution of the heat diffusion equation on the sphere. This includes a formal discussion regarding convolution of functions on the sphere and its implementation in the spatial and spherical Fourier domains. Section 4.3 formulates the sSIFT keypoint detector. A methodology used to estimate the bandwidth of a wide-angle image is presented. This bandwidth is used to select the minimum required sample rate that must be used when finding the spherical harmonic expansion of an image without aliasing. In the case where the required sample rate exceeds the maximum computationally feasible, a suitable anti-aliasing interpolation filter is designed that can be used to counterfeit aliasing. Experiments are presented which compare the relative performance of sSIFT to a ‘blind’ application of SIFT (operating directly on wide-angle images without accounting for the camera distortion) using synthetic wide-angle images. This performance is based on the percentage correlation of keypoints detected in wide-angle image pairs. The relative performance of pSIFT to both sSIFT and SIFT is then compared for the same experiments in section 4.4. The pSIFT keypoint detector is presented in section 4.4, where an approximation to the convolution of the spherical Gaussian and the image mapped to the sphere is developed that can be implemented efficiently on the stereographic image plane. The performance of pSIFT is then compared to sSIFT and SIFT using the same experimental procedure in section 4.3. Section 4.5 presents the method used to evaluate SIFT descriptors for sSIFT and pSIFT keypoints. Section 4.6 then compares the relative performance of sSIFT and pSIFT to a blind application of SIFT using three wide-angle image sequences — results for SIFT applied to rectified perspective views are also obtained. The performance metric used is recall versus 1-precision, which is a measure of the ability to recall the most correct correspondences between image pairs with the least number of outliers. Finally, conclusions are presented in section 4.7.

## 4.2 Scale-Space for Wide-Angle Images

As discussed in chapter 3, for a perspective image  $I \in \mathbb{R}^2$ , the scale-space representation  $L(\cdot; t)$  of the image at scale  $t$  is obtained as the solution of the heat diffusion equation

$$\partial_t L(u, v; t) = \frac{1}{2} \Delta L(u, v; t), \quad \Delta L(u, v; t) = \frac{1}{k} \partial_t L(u, v; t), \quad (4.1)$$

where  $\Delta$  is the Laplacian defined in  $\mathbb{R}^2$ . For initial condition  $L(\cdot; 0) = I$ , the scale-space representation  $L(\cdot; t)$  of an image at scale  $t$  is obtained as the convolution of the image

$I$  with the Gaussian  $G(\cdot; t)$ <sup>1</sup>.

Recall from chapter 2 that a point  $\eta$  on the unit view sphere  $\mathbb{S}^2$  is parameterised as  $\eta(\theta, \phi) = [\sin \theta \cos \phi, \sin \theta \sin \phi, \cos \theta]^T$ , where  $\theta \in [0, \pi)$  is an angle of colatitude and  $\phi \in [0, 2\pi)$  is an angle of longitude. Bülow [31] proposed that the *scale-space representation*  $L_{\mathbb{S}^2}(\cdot; t)$  of a continuous function  $f \in L^2(\mathbb{S}^2)$  is obtained as the solution of the spherical heat diffusion equation

$$\Delta_{\mathbb{S}^2} L_{\mathbb{S}^2}(\theta, \phi; t) = \frac{1}{k} \partial_t L_{\mathbb{S}^2}(\theta, \phi; t) \quad (4.2)$$

with initial condition  $L_{\mathbb{S}^2}(\cdot; 0) = f$ . The parameters  $k$  and  $t$  refer to thermal conductivity and time respectively. The parameter  $\Delta_{\mathbb{S}^2}$  is the Laplace operator restricted to the unit sphere  $\mathbb{S}^2$  [108]<sup>2</sup>:

$$\Delta_{\mathbb{S}^2} = \frac{1}{\sin \theta} \frac{\partial}{\partial \theta} \left( \sin \theta \frac{\partial f}{\partial \theta} \right) + \frac{1}{\sin^2 \theta} \frac{\partial^2 f}{\partial \phi^2}. \quad (4.3)$$

Bülow derived a solution to 4.2, where for initial condition  $L_{\mathbb{S}^2}(\cdot; 0) = f$ , the scale-space representation  $L_{\mathbb{S}^2}(\cdot; kt)$  of the function  $f$  is the convolution of  $f$  with the scale-space operator for functions on the sphere  $G_{\mathbb{S}^2}(\cdot; kt)$ . The parameter  $kt$  is defined as the ‘scale’. Before discussing this solution, including the definition of  $G_{\mathbb{S}^2}(\cdot; kt)$  (section 4.2.2) and convolution of functions on the sphere (section 4.2.3), it is necessary to provide a brief introduction to spherical harmonics.

## 4.2.1 Spherical Harmonics

A function  $f$  is harmonic if the solutions to the second partial derivatives of Laplace’s equation are continuous:

$$\Delta f = 0, \quad (4.4)$$

where  $\Delta$  is the Laplacian defined in the same domain as the function  $f$ . The eigenfunctions of the spherical Laplace operator  $\Delta_{\mathbb{S}^2}$  are the spherical harmonic functions

<sup>1</sup> $G(\cdot; t)$  is the Gaussian function sampled at discrete pixel positions, and whose scale is more frequently defined by  $\sigma = t^2$ , where  $\sigma$  is the standard deviation of the Gaussian.

<sup>2</sup>The Laplacian  $\Delta$  of a function  $u(\theta, \phi, r)$  defined in spherical polar coordinates is

$$\Delta u = \frac{1}{r^2} \frac{\partial}{\partial r} \left( r^2 \frac{\partial u}{\partial r} \right) + \frac{1}{r^2 \sin \theta} \frac{\partial}{\partial \theta} \left( \sin \theta \frac{\partial u}{\partial \theta} \right) + \frac{1}{r^2 \sin^2 \theta} \frac{\partial^2 u}{\partial \phi^2}.$$

The Laplacian  $\Delta_{\mathbb{S}^2}$  restricted to the unit sphere is found by setting  $r = 1$ ,  $\frac{\partial f}{\partial r} = 0$ .

$Y_l^m : \mathbb{S}^2 \mapsto \mathbb{C}$  of degree  $l$  and order  $m$  [92]:

$$\Delta_{\mathbb{S}^2} Y_l^m = -l(l+1)Y_l^m. \quad (4.5)$$

The spherical harmonic function  $Y_l^m(\eta)$  evaluated at some point  $\eta(\theta, \phi)$  on the unit sphere is

$$Y_l^m(\eta) = \sqrt{\frac{2l+1}{4\pi} \frac{(l-m)!}{(l+m)!}} P_l^m(\cos(\theta)) e^{im\phi}, \quad l \in \mathbb{N}, |m| \leq l, \quad (4.6)$$

where  $P_l^m$  are the associated Legendre polynomials

$$P_l^m(x) = \frac{(-1)^m (1-x^2)^{\frac{m}{2}}}{2^l l!} \frac{d^{l+m}}{dx^{l+m}} (x^2-1)^l. \quad (4.7)$$

Figure 4.1 displays the magnitude of the real components of the spherical harmonics functions sampled on an equiangular  $\theta, \phi$  grid up to degree  $l = 5$  and order  $0 \leq m \leq l$ . The values of the zonal harmonic functions ( $m = 0$ ) are dependent only on the angle of colatitude  $\theta$  and are shown in the top row. The sectoral harmonic functions ( $m = ||l||$ ) are those in the diagonal from top left to bottom right, where the angle of colatitude  $\theta$  contributes only to an overall scale factor. All other cases ( $m \neq 0, m \neq ||l||$ ) are the tesseral harmonic functions.

Any square integrable function  $f \in L^2(\mathbb{S}^2)$  on the unit sphere can be expanded as a linear summation of spherical harmonics:

$$f = \sum_{l \in \mathbb{N}} \sum_{|m| \leq l} \hat{f}_l^m Y_l^m, \quad (4.8)$$

where the coefficients  $\hat{f}_l^m$  are the spherical Fourier transform (spectrum) of  $f$  defined as

$$\hat{f}_l^m = \int_{\mathbb{S}^2} f(\eta) \overline{Y_l^m(\eta)} d\eta, \quad (4.9)$$

where  $\overline{Y_l^m}$  denotes the complex conjugate. The integral  $d\eta$  in 4.9 is defined to mean  $d\eta \triangleq \sin(\theta) d\theta d\phi$ . As the image  $I$  obtained with any central projection camera can be mapped to the the image  $I_{\mathbb{S}^2}$  on the sphere, it can be expanded into spherical harmonics as

$$I_{\mathbb{S}^2} = \sum_{l \in \mathbb{N}} \sum_{|m| \leq l} (\hat{I}_{\mathbb{S}^2})_l^m Y_l^m, \quad (\hat{I}_{\mathbb{S}^2})_l^m = \int_{\mathbb{S}^2} I_{\mathbb{S}^2}(\eta) \overline{Y_l^m(\eta)} d\eta, \quad (4.10)$$

where  $\hat{I}_{\mathbb{S}^2}$  is the spectrum of the image.

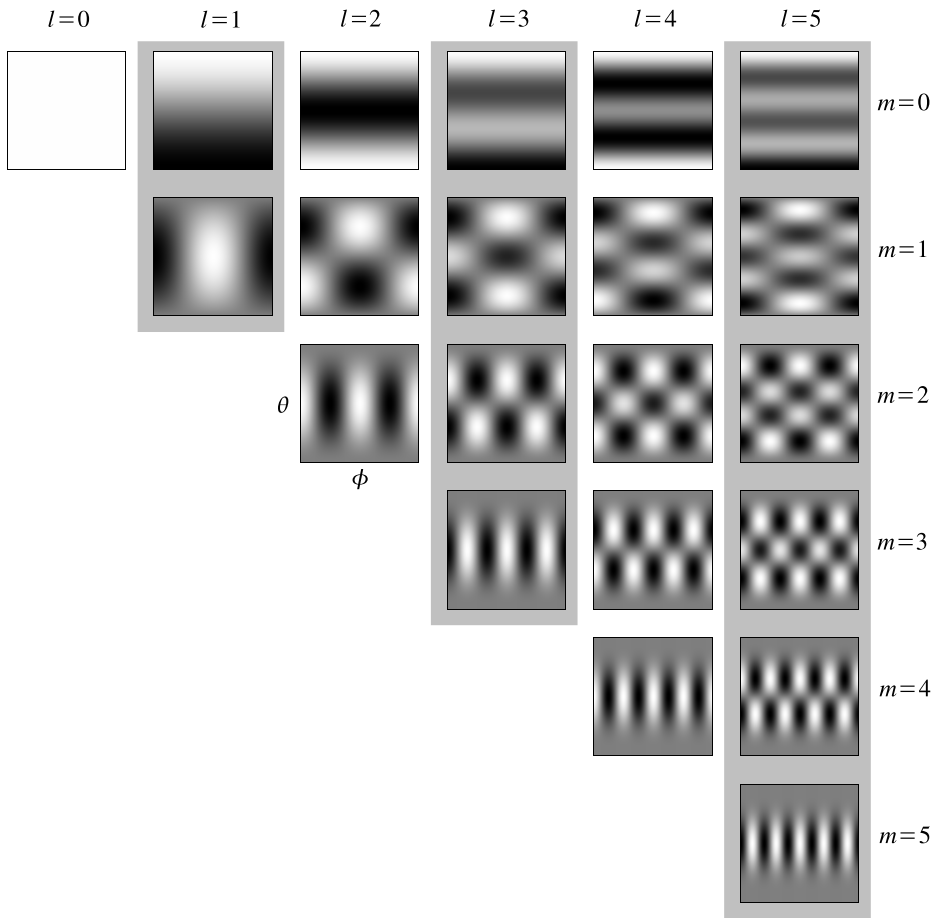


Figure 4.1: The magnitude of the real components of the spherical harmonic functions  $Y_l^m$  shown up to degree  $l = 5$  and order  $0 \leq m \leq l$ . The functions are shown on an equiangular  $\theta, \phi$  plane with angles  $\theta \in [0, \pi)$  and  $\phi \in [0, 2\pi)$ .

## 4.2.2 Spherical Gaussian Function

As just discussed, a solution to the heat diffusion equation with initial condition  $L_{\mathbb{S}^2}(\cdot; 0) = f$  was first solved by Bülow [31]. Although the same solution was obtained using an alternate derivation by Chung [40] who applied it to cortical data analysis [41], the solution of Bülow is described here.

Assuming that  $L_{\mathbb{S}^2}(\theta, \phi; kt)$  is separable, where  $kt$  is the scale, the spherical heat diffusion equation in 4.2 can be expressed in the spherical Fourier domain using equation 4.8 as

$$\Delta_{\mathbb{S}^2} Y_l^m (\hat{L}_{\mathbb{S}^2})_l^m(kt) = \frac{1}{k} \partial_t Y_l^m (\hat{L}_{\mathbb{S}^2})_l^m(kt). \quad (4.11)$$

Then, substituting 4.5 into 4.11 gives the first order ordinary differential equation

$$-l(l+1)k(\hat{L}_{\mathbb{S}^2})_l^m(t) = \partial_t (\hat{L}_{\mathbb{S}^2})_l^m(t), \quad (4.12)$$

whose unique solution is

$$(\hat{L}_{\mathbb{S}^2})_l^m(kt) = (\hat{L}_{\mathbb{S}^2})_l^m(0)e^{-l(l+1)kt}, \quad (4.13)$$

where  $(\hat{L}_{\mathbb{S}^2})_l^m(0)$  is the spectrum of the initial condition.

The Green's function of 4.13 is the scale-space operator  $G_{\mathbb{S}^2}(\cdot; kt)$  for functions on the sphere, and can be considered as the Gauss function on the sphere [31]. For convenience,  $G_{\mathbb{S}^2}(\cdot; kt)$  is referred to as the *spherical Gaussian* for the remainder of the thesis. Bülow obtains the solution for  $G_{\mathbb{S}^2}(\cdot; kt)$  with initial condition

$$G_{\mathbb{S}^2}(\theta, \phi; 0) = \delta_{\mathbb{S}^2}(\theta, \phi), \quad (4.14)$$

where  $\delta_{\mathbb{S}^2}$  is the spherical Dirac (unit impulse) function at the north pole  $\mathbf{n} = (0, 0, 1)^T$ . The spherical Dirac function  $\delta_{\mathbb{S}^2}$  is defined as

$$f(\mathbf{n}) = \int_{\eta \in \mathbb{S}^2} f(\eta) \delta_{\mathbb{S}^2}(\theta, \phi) d\eta, \quad f \in L^2(\mathbb{S}^2), \quad (4.15)$$

and can be written as the spherical harmonic expansion

$$\delta_{\mathbb{S}^2} = \sum_{l \in \mathbb{N}} \sqrt{\frac{2l+1}{4\pi}} Y_l^0, \quad (4.16)$$

where  $Y_l^0(\theta, \phi)$  are the zonal harmonic functions (refer to the top row of figure 4.1):

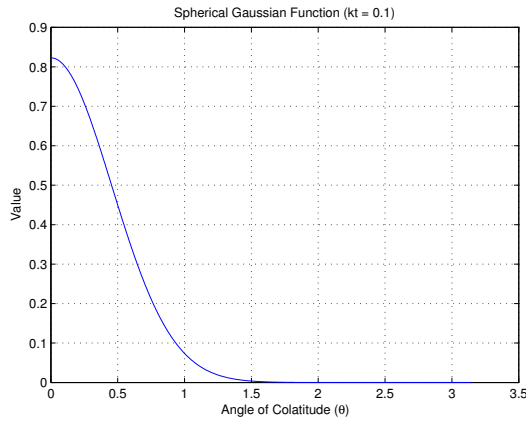
$$Y_l^0(\theta, \phi) = \sqrt{\frac{2l+1}{4\pi}} P_l(\cos \theta). \quad (4.17)$$

A derivation of the spherical harmonic expansion of  $\delta_{\mathbb{S}^2}$  is presented in appendix B.

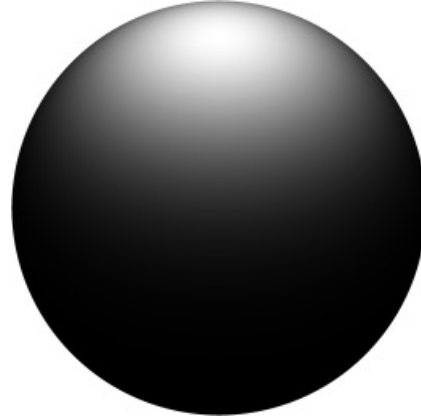
As equation 4.16 defines the spectrum of the initial condition  $G_{\mathbb{S}^2}(\cdot; 0)$ , and equation 4.13 describes how the spectrum of a function on the sphere diffuses over time  $t$  for some value of  $k$ , Bülow obtained as the solution for the spherical Gaussian

$$G_{\mathbb{S}^2}(\theta, \phi; kt) = \sum_{l \in \mathbb{N}} \sqrt{\frac{2l+1}{4\pi}} Y_l^0(\theta, \phi) e^{-l(l+1)kt}. \quad (4.18)$$

The solution is the sum of only the zonal harmonic functions as the function is rotationally symmetrical about the north pole  $\mathbf{n}$ . The spherical Gaussian, which is also known as the Gauss-Weierstrass kernel [80, 40], defines the evolution of a unit heat source at the north pole of a thin spherical vessel with constant thermal conductivity  $k$  over time  $t$ . The value of the heat profile at a given position  $\eta(\theta, \phi)$  can be obtained di-



(a) Spherical Gaussian function versus angle of colatitude  $\theta$



(b) Spherical Gaussian function illustrated on the unit radius sphere. By definition, the function is centred at the north pole.

Figure 4.2: The spherical Gaussian function versus angle of colatitude  $\theta$ .

rectly from 4.18. Again, as the spherical Gaussian is used in the context of scale-space analysis, the parameter  $kt$  is referred to simply as ‘scale’. Figure 4.2 shows value of the spherical Gaussian versus angle of colatitude  $\theta$  for scale  $kt = 0.1$ .

As the solution for  $G_{\mathbb{S}^2}(\cdot; kt)$  was obtained having as initial condition the spherical Dirac (unit impulse) at the north pole, one would expect the integral of  $G_{\mathbb{S}^2}(\cdot; kt)$  to be unity for all scales  $kt$ . Bülow [31] shows that this property holds as the integral of any square integrable function  $f \in L^2(\mathbb{S}^2)$  on the sphere is dependent only on the DC component of its spectrum  $\hat{f}_0^0$ :

$$\int_{\mathbb{S}^2} f(\eta) d\eta = \sqrt{4\pi} \hat{f}_0^0. \quad (4.19)$$

It follows from the definition of  $G_{\mathbb{S}^2}(\cdot; kt)$  in 4.18 that

$$\int_{\mathbb{S}^2} G_{\mathbb{S}^2}(\cdot; kt) = \sqrt{4\pi} \left( \sqrt{\frac{1}{4\pi}} e^0 \right) = 1, \quad \forall kt. \quad (4.20)$$

This property is important in later discussions regarding convolution of a function on the sphere with the spherical Gaussian.

### 4.2.3 Spherical Diffusion by Convolution

The scale-space representation  $L_{\mathbb{S}^2}(\cdot; kt)$  of a function  $f$  on the sphere with initial condition  $L_{\mathbb{S}^2}(\cdot; 0) = f$  is the convolution of  $f$  with the spherical Gaussian  $G_{\mathbb{S}^2}(\cdot; kt)$ . This

section discusses the convolution of functions on the sphere.

For all rotations  $R \in SO(3)$ , define the operator  $\Lambda(R)$  which rotates a point on the sphere to a new position  $\Lambda(R)f(\boldsymbol{\eta}) = f(R^{-1}\boldsymbol{\eta})$ . This operator could use, for example, Euler rotations or quaternions. The former is used here where  $R$  is parameterised as  $R = R_z(\gamma)R_y(\beta)R_z(\alpha)$ . Using this notation, Driscoll and Healy [63] define the convolution of any two (square integrable) functions  $f$  and  $h$  on the sphere as

$$(f * h)(\boldsymbol{\eta}) = \int_{R \in SO(3)} f(R\mathbf{n})h(R^{-1}\boldsymbol{\eta})dR, \quad \boldsymbol{\eta} \in \mathbb{S}^2, \quad (4.21)$$

where  $\mathbf{n}$  is the north pole. For an image  $I \mapsto I_{\mathbb{S}^2}$  mapped to the sphere, assuming that the initial condition  $L_{\mathbb{S}^2}(\cdot; 0)$  is  $I_{\mathbb{S}^2}$ , the scale-space representation  $L_{\mathbb{S}^2}(\cdot; kt)$  of the image at scale  $kt$  can be obtained using 4.21 as

$$L_{\mathbb{S}^2}(\boldsymbol{\eta}; kt) = \frac{1}{2\pi} (I_{\mathbb{S}^2} * G_{\mathbb{S}^2}(\cdot; kt))(\boldsymbol{\eta}) \quad (4.22)$$

$$= \frac{1}{2\pi} \int_{R \in SO(3)} I_{\mathbb{S}^2}(R\mathbf{n})G_{\mathbb{S}^2}(R^{-1}\boldsymbol{\eta}; kt)dR, \quad \boldsymbol{\eta} \in \mathbb{S}^2. \quad (4.23)$$

The additional  $\frac{1}{2\pi}$  factor is required as the convolution defined in 4.21 integrates over all rotations  $R \in SO(3)$ , whereas the integral of  $G_{\mathbb{S}^2}$  over  $\mathbb{S}^2$  is unity as shown previously in equation 4.20. Adding the additional  $\frac{1}{2\pi}$  ensures that the integral of  $G_{\mathbb{S}^2}$  over  $SO(3)$  is unity.

The convolution in equation 4.22 can be implemented in the spatial or frequency domains, and these implementations will be discussed in sections 4.2.3.1 and 4.2.3.2 respectively. However, before proceeding it is important to note that the scale-space representation  $L_{\mathbb{S}^2}(\cdot; kt)$  of an image  $I_{\mathbb{S}^2}$  is a function on the sphere. As each pixel  $\mathbf{u}$  maps to a unique point  $\boldsymbol{\eta}$  on the unit view sphere for a central projection camera, the scale-space image  $L_{\mathbb{S}^2}(\cdot; kt)$  can be projected back to the scale-space image  $\mathcal{L}_{\mathbb{S}^2}(\cdot; kt)$  on the original image plane,  $L_{\mathbb{S}^2}(\boldsymbol{\eta}; kt) \mapsto \mathcal{L}_{\mathbb{S}^2}(u, v; kt)$ .

#### 4.2.3.1 Spatial Domain

The definition of the convolution of two-functions on the sphere in 4.21 integrates over all rotations  $R \in SO(3)$ . Daniilidis et al [56] observed that if one of the functions is symmetrical about the north pole, the integration can be restricted to the subgroup of rotations  $R = R_z(\gamma)R_y(\beta)$  — the redundant rotation is the rotation about the symmetrical axis of the filter. They formulated a simplified form of the convolution defined in 4.22, restricted to an integration over  $\mathbb{S}^2$ , which can be used to rewrite the convolu-

tion defined in 4.22 as

$$L_{\mathbb{S}^2}(\beta, \gamma; kt) = (I_{\mathbb{S}^2} * G_{\mathbb{S}^2}(kt))(\beta, \gamma) = \int_{\eta \in \mathbb{S}^2} I_{\mathbb{S}^2}(\eta) G_{\mathbb{S}^2}(R^{-1}\eta; kt) d\eta, \quad (4.24)$$

where  $d\eta = \sin(\theta)d\theta d\phi$ . Here,  $L_{\mathbb{S}^2}(\beta, \gamma)$  is evaluated at the point  $\eta' = R\mathbf{n}$  on the sphere, where  $R = R_z(\gamma)R_y(\beta)$ . The  $\frac{1}{2\pi}$  factor is no longer required as the integral is taken over  $\mathbb{S}^2$ . Interestingly, the convolution in 4.28 is similar to the definition of correlation on the sphere [30, 247], where the correlation of  $I_{\mathbb{S}^2}$  and  $G_{\mathbb{S}^2}(\cdot; kt)$  would be

$$(I_{\mathbb{S}^2} \star G_{\mathbb{S}^2}(\cdot; kt))(\alpha, \beta, \gamma) = \int_{\eta \in \mathbb{S}^2} I_{\mathbb{S}^2}(\eta) G_{\mathbb{S}^2}(R^{-1}\eta; kt) d\eta, \quad R \in SO(3), \quad (4.25)$$

whose response is defined in  $SO(3)$ . Equation 4.25 is equivalent to the correlation of  $L_{\mathbb{S}^2}$  and  $G_{\mathbb{S}^2}$  defined for any fixed angle  $\alpha$ .

The convolution in equation 4.28 can be implemented on the image plane [56]. To simplify the discussions, a number of notations are used with reference to figure 4.3. The spherical Gaussian  $G_{\mathbb{S}^2}(\cdot; kt)$  is by definition centred at the north pole  $\mathbf{n}$ . For any central projection wide-angle camera, it can be projected to the function  $\mathcal{G}_{\mathbb{S}^2}(\cdot; kt)$  on the image, centred at the principal point  $\mathbf{u}_0$ . The spherical Gaussian centred at the point  $\eta' = R\mathbf{n}$ , where  $R$  is some rotation matrix, is denoted  $G_{\mathbb{S}^2(\eta')}(\cdot; kt)$  and projects to the function  $\mathcal{G}_{\mathbb{S}^2(\eta')}(\cdot; kt)$  on the image, centred at the point  $\mathbf{u}' \mapsto \eta'$ . The function  $\mathcal{G}_{\mathbb{S}^2(\eta')}(u, v; kt)$  has the values

$$\mathcal{G}_{\mathbb{S}^2(\eta')}(u, v; kt) = \sum_{l \in \mathbb{N}} \sqrt{\frac{2l+1}{4\pi}} Y_l^0(\theta_{\mathbf{u}'-\mathbf{u}}) e^{-l(l+1)kt}, \quad (4.26)$$

where  $\theta_{\mathbf{u}'-\mathbf{u}}$  is the angle on the sphere between the point  $\eta$  and  $\eta'$

$$\theta_{\mathbf{u}'-\mathbf{u}} = \cos^{-1}(\eta T \eta'), \quad (4.27)$$

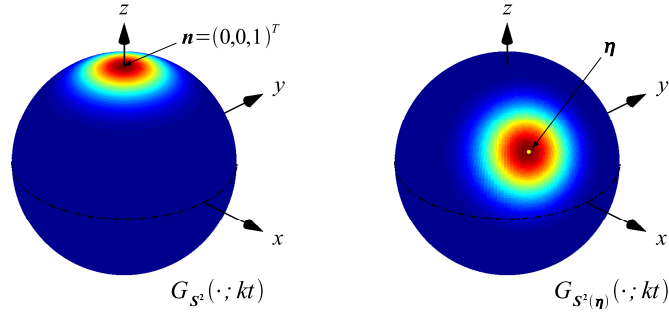
where  $\mathbf{u} \mapsto \eta$ . The values of  $\mathcal{G}_{\mathbb{S}^2}(u, v; kt)$  can be obtained in the same manner by substituting  $\mathbf{u}'$  with  $\mathbf{u}_0$ , where  $\eta' = \mathbf{n}$ . Using these notations, the definition of correlation in 4.28 can be rewritten as

$$L_{\mathbb{S}^2}(\eta'; kt) = (I_{\mathbb{S}^2} * G_{\mathbb{S}^2}(\cdot; kt))(\eta') \quad (4.28)$$

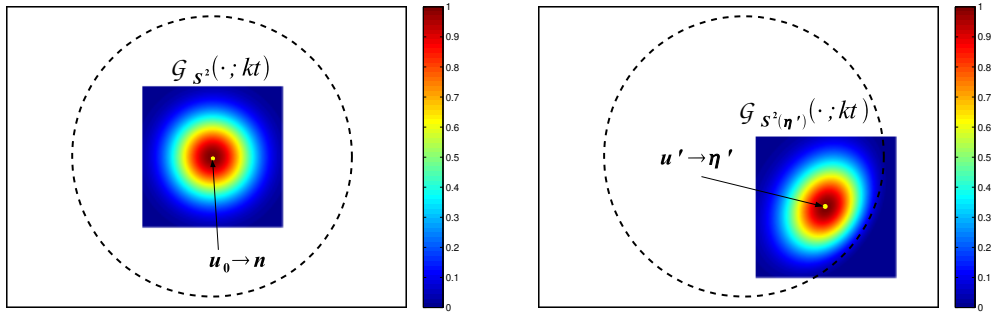
$$= \int_{\eta \in \mathbb{S}^2} I_{\mathbb{S}^2}(\eta) G_{\mathbb{S}^2(\eta')}(\eta; kt) d\eta, \quad (4.29)$$

where  $\eta' = R\mathbf{n}$  and  $R = R_z(\gamma)R_y(\beta)$ . This convolution can be implemented to find the





(a) The spherical Gaussian  $G_{\mathbb{S}^2}(\cdot; kt)$  defined at the north pole  $\mathbf{n}$  (left), and the spherical Gaussian  $G_{\mathbb{S}^2(\eta)}(\cdot; kt)$  centred at the point  $\eta'$  (right).



(b) The truncated spherical Gaussian kernels  $\mathcal{G}_{\mathbb{S}^2}(\cdot; kt)$  and  $\mathcal{G}_{\mathbb{S}^2(\eta')}(\cdot; kt)$  as they would appear on a typical wide-angle image. The circular dashed line represents the field of view of the wide-angle camera.

Figure 4.3: The spherical Gaussian  $G_{\mathbb{S}^2}(kt)$  is defined centred at the north pole  $\mathbf{n}$ . The spherical Gaussian rotated to a new point  $\eta' = R\mathbf{n}$  is denoted  $G_{\mathbb{S}^2(\eta')}(kt)$ .  $G_{\mathbb{S}^2}(\cdot; kt)$  projects for the given camera model to the kernel  $\mathcal{G}_{\mathbb{S}^2}(\cdot; kt)$  on the image, centred at the principal point  $\mathbf{u}_0 \mapsto \mathbf{n}$ .  $G_{\mathbb{S}^2(\eta')}(\cdot; kt)$  projects for the given camera model to the kernel  $\mathcal{G}_{\mathbb{S}^2(\eta')}(\cdot; kt)$  on the image, centred at the point  $\mathbf{u}' \mapsto \eta$ .

scale-space image value  $\mathcal{L}_{\mathbb{S}^2}(u', v'; kt) \mapsto \mathcal{L}_{\mathbb{S}^2}(\eta'; kt)$  at some pixel position  $\mathbf{u}'$  as

$$\mathcal{L}_{\mathbb{S}^2}(u', v'; kt) = \sum_{u,v} I(u, v) \mathcal{G}_{\mathbb{S}^2(\eta')}(u, v; kt). \quad (4.30)$$

Implementing the convolution on the image plane has the advantage that the original image values are used. However, this method is computationally expensive as a unique kernel  $\mathcal{G}_{\mathbb{S}^2(\eta')}(\cdot; kt)$  is required to evaluate  $\mathcal{L}_{\mathbb{S}^2}(u', v'; kt)$  at each pixel position  $\mathbf{u}'$ . The change in appearance of  $\mathcal{G}_{\mathbb{S}^2(\eta')}(\cdot; kt)$  at different positions in the image is evident in figure 4.3. If all kernels  $\mathcal{G}_{\mathbb{S}^2(\eta')}(\cdot; kt)$  at a given scale  $kt$  were precomputed offline with size  $n \times n$ , convolution with an  $m \times m$  sized image would require a total of  $2n^2m^2$  computations ( $O(n^2)$ ). In contrast, the convolution of an image with a two-dimensional Gaussian  $G(\cdot; \sigma)$  can be obtained efficiently as<sup>3</sup>

<sup>3</sup>The two-dimensional Gaussian  $G(\cdot; \sigma)$  is a rank 1 matrix and can be written as the outer product

$$I * G(\cdot; \sigma) = (I * G_x(\cdot; \sigma)) * G_y(\cdot; \sigma) \quad (4.31)$$

where  $G_x(\cdot; \sigma)$  and  $G_y(\cdot; \sigma)$  are one-dimensional Gaussians with equal scale  $\sigma$  and length  $n$  in the  $x$  and  $y$  directions respectively. This requires a total of  $4nm^2$  computations ( $O(n)$ ). Fortunately, an efficient approach to convolution in the frequency (spherical Fourier) domain can be used.

### 4.2.3.2 Spherical Fourier Domain

From the definition of convolution in equation 4.21, Driscoll and Healy [63] prove the following theorem for convolution of a square integrable function  $f$  and rotationally symmetrical filter  $h$  on the sphere in the spherical Fourier domain:

**Theorem 4.1** *For functions  $f, h \in L^2(\mathbb{S}^2)$ , the transform of the convolution is a point-wise product of the transforms*

$$(\widehat{f * h})_l^m = 2\pi \sqrt{\frac{4\pi}{2l+1}} \hat{f}_l^m \hat{h}_l^0 \quad (4.32)$$

where  $\hat{h}_l^0$  are the zonal harmonics coefficients of the symmetrical filter  $h$ , and  $(\widehat{f * h})_l^m$  is the spectrum of the convolution. This convolution theorem can be used to define scale-space for functions on the sphere as a response in the spherical Fourier domain by

$$(\hat{L}_{\mathbb{S}^2})_l^m(kt) = (L_{\mathbb{S}^2}(\cdot; 0) * \widehat{G}_{\mathbb{S}^2}(\cdot; kt))_l^m \quad (4.33)$$

$$= \sqrt{\frac{4\pi}{2l+1}} (\hat{L}_{\mathbb{S}^2})_l^m(0) (\hat{G}_{\mathbb{S}^2})_l^0(kt) \quad (4.34)$$

$$= (\hat{L}_{\mathbb{S}^2})_l^m(0) e^{-l(l+1)kt}, \quad (4.35)$$

which is the same as equation 4.13. As before, an additional  $\frac{1}{2\pi}$  factor has been added as the convolution in 4.1 is derived from the definition of convolution in 4.21 which integrates over all rotations  $R \in SO(3)$ . If the initial condition  $\hat{L}_{\mathbb{S}^2}(0) = \hat{I}_{\mathbb{S}^2}$  is the spectrum of the image  $I_{\mathbb{S}^2}$ , this convolution theorem can be used to find the scale-space representation  $\hat{L}_{\mathbb{S}^2}(kt)$  of an image  $I_{\mathbb{S}^2}$  as a response in the spherical Fourier domain as

$$(\hat{L}_{\mathbb{S}^2})_l^m(kt) = (\hat{I}_{\mathbb{S}^2})_l^m e^{-l(l+1)kt}. \quad (4.36)$$

of two one-dimensional Gaussians  $G_x(\sigma)$  and  $G_y(\sigma)$  in the  $x$  and  $y$  directions respectively,  $G(\cdot; \sigma) = G_y(\sigma)G_x(\sigma)$  (i.e..  $G(\cdot; \sigma)$  is separable). The convolution of an image with  $G(\cdot; \sigma)$  is therefore equal to the successive convolutions of the image with  $G_x(\sigma)$  and  $G_y(\sigma)$ .

As a final note, from the definition of convolution in equation 4.33, Bülow [31] shows that the scale-space for functions on the sphere satisfies the semi-group property:

$$\begin{aligned}
 ((\hat{L}_{\mathbb{S}^2})_l^m(kt))(ks) &= (\hat{L}_{\mathbb{S}^2})_l^m e^{-l(l+1)kt} e^{-l(l+1)ks} \\
 &= (\hat{L}_{\mathbb{S}^2})_l^m e^{-l(l+1)k(t+s)} \\
 &= (\hat{L}_{\mathbb{S}^2})_l^m(k(t+s)).
 \end{aligned} \tag{4.37}$$

### 4.3 Scale-Invariant Keypoint Detection: spherical SIFT (sSIFT)

The spherical SIFT (sSIFT) keypoint detector is developed in this section. sSIFT finds for a given image  $I_{\mathbb{S}^2}$  the set of scale-space images  $L_{\mathbb{S}^2}(\cdot; kt)$  by convolving  $I_{\mathbb{S}^2}$  with  $G_{\mathbb{S}^2}(\cdot; kt)$  in the spherical Fourier domain using the method outlined in section 4.2.3.2. This requires finding the spectrum  $\hat{I}_{\mathbb{S}^2}$  of the the image. The set of all scale-space images  $L_{\mathbb{S}^2}(\cdot; kt)$  are then mapped back to the set of scale-space images  $\mathcal{L}_{\mathbb{S}^2}(\cdot; kt)$  on the original image plane. Keypoints are detected as local extrema in the difference of scale-space (difference of Gaussian) images  $\mathcal{D}_{\mathbb{S}^2}(kt_i) = \mathcal{L}_{\mathbb{S}^2}(kt_i) - \mathcal{L}_{\mathbb{S}^2}(kt_{i-1})$ , where the same principles of edge removal and quadratic interpolation of keypoint location and scale used by SIFT are utilised. The support region for a keypoint is defined as a circle on the sphere, centred about the keypoint position on the sphere, whose size is set relative to the characteristic scale  $kt$  of the keypoint.

This section starts by describing the practical procedure used to find the spectrum  $\hat{I}_{\mathbb{S}^2}$  of a wide-angle image. A means for estimating the *bandwidth* of a wide-angle image from the camera intrinsic parameters is developed which is used to find the minimum sample rate required to obtain the spectrum of the image without aliasing artifacts. A suitable anti-aliasing interpolation filter is then designed, and can be used to minimise aliasing when the required sample rate exceeds the maximum computationally feasible. A method to select a suitable set of scales  $kt$  is then developed which is based on the camera intrinsic parameters and the scales  $\sigma$  used by SIFT. The specific details of keypoint detection are then discussed, where the definition of a key-points support region is introduced. Finally, the percentage correlation and outright number of keypoints detected in synthetic wide-angle images using SIFT and sSIFT are compared— SIFT operates directly on the wide-angle images without making any account for the cameras radial distortion.

### 4.3.1 Spherical Fourier Transform (spectrum) of a Wide-Angle Image

The discrete spherical Fourier transform (SFT) of a wide-angle image is used to obtain the spectrum  $\hat{I}_{S^2} = SFT(I_{S^2})$ . The s2kit<sup>4</sup> is used to find the SFT of the image, and requires sampling the image function  $I_{S^2}$  at a predefined set of sample points  $\eta(\theta, \phi)$ .

#### 4.3.1.1 Sample Scheme

The sample points  $\eta(\theta, \phi)$  used by s2kit are based on the sample scheme described by Driscoll and Healy [63]. Driscoll and Healy [63] show that for a bandlimited function  $f$  on the sphere (i.e. a function with a bandwidth of  $b_f$ ),  $\hat{f}_l^m = 0$  for  $l \geq b_f$ , where  $l$  is some integer value. The bandwidth of a function on the sphere can be visualised from inspection of figure 4.1, where it can be seen that for a given degree  $l$ , the maximum number of cycles of the spherical harmonic functions per  $2\pi$  radians is equal to  $l$ . As  $\hat{f}_l^m = 0$  for  $l \geq b_f$ , the bandwidth  $b_f$  of a function on the sphere is the number of cycles per  $2\pi$  radians.

To recover exactly the original signal, the frequency of sampling  $2b$  must be at least  $2b_f$  —  $b$  is defined as the *sample rate*. The reader is referred to Driscoll and Healy [63] for a discussion on the selection of the sampling scheme with respect to computational efficiency. For sample rate  $b$ , s2kit samples an image at the points  $\eta(\theta_i, \phi_j)$ , where<sup>5</sup>

$$\theta_i = \frac{\pi(2i+1)}{4b} \quad i \in \{0, 1, \dots, 2b-1\} \quad \text{and} \quad \phi_j = \frac{\pi j}{b} \quad j \in \{0, 1, \dots, 2b-1\}, \quad (4.38)$$

The sample points and are illustrated in figure 4.4 for sample rate  $b = 8$ .

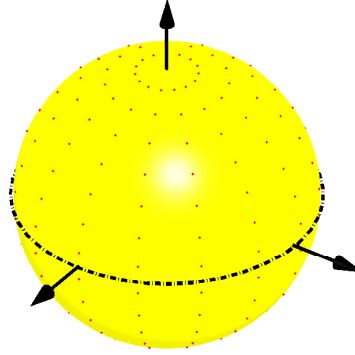
As the density of the sample points in the direction of longitude  $\phi$  decreases as one moves away from the poles, the discrete SFT of equation 4.9 used by Driscoll and Healy is [63]:

$$\hat{f}_l^m = \frac{\sqrt{2\pi}}{2b} \sum_{i=0}^{2b-1} \sum_{j=0}^{2b-1} a_i^{(b)} f(\theta_i, \phi_j) \overline{Y_l^m(\theta_i, \phi_j)}, \quad (4.39)$$

where  $a_i^{(b)}$  is used to weight the samples to account for the variable sample density in

<sup>4</sup>Available <http://www.cs.dartmouth.edu/~geelong/sphere/>

<sup>5</sup>The angles  $\theta$  used by Driscoll and Healy are  $\theta_i = (\pi i/2b)$  — s2kit adds a half sample offset.

Figure 4.4: The sampling scheme used by s2kit ( $b=8$ )

the direction of longitude  $\phi$  which varies with angle of colatitude  $\theta$ :

$$a_i^{(b)} = \frac{2\sqrt{2}}{n} \sin\left(\frac{\pi i}{n}\right) \sum_{l=0}^{\frac{n}{2}-1} \frac{1}{2l+1} \sin\left([2l+1]\frac{\pi i}{n}\right), \quad i \in \{0, 1, \dots, n-1\}. \quad (4.40)$$

In practice, this weighting is implemented automatically using s2kit. The input to s2kit is simply the image values sampled at the points  $\eta(\theta_i, \phi_j)$ , where  $\theta_i$  and  $\phi_j$  are given in equation 4.38.

The bandlimit (bandwidth) of a wide-angle image needs to be known to select an appropriate sample rate  $b$ . If  $b$  is less than the bandwidth of the image, there is the potential that the spectrum  $\hat{I}_{\mathbb{S}^2}$  will contain some form of aliasing. If the required sample rate  $b$  needed to prevent aliasing exceeds the maximum computationally feasible, which is  $b = 512$  for the hardware used in this work, some approach to minimise or prevent aliasing needs to be considered.

#### 4.3.1.2 Spherical Bandwidth of a Wide-Angle Image

Referring to equation 4.38, the function on the sphere is sampled at  $2b$  points over the range of angles of longitude  $\phi = [0, 2\pi)$ . The greatest angular separation between sample points  $\phi_j$  occurs at the equator  $\theta = \pi/2$ , where the change in angle  $d\psi$  along the great circle (equator) per change in sample position  $ds$  is

$$\frac{d\psi}{ds} = \frac{\pi}{b}. \quad (4.41)$$

Assume that there exists some function  $f$  on the sphere whose values are known only at a set of sample points  $s$ . If the change in angle  $d\psi$  between and two samples points is known (i.e.  $\frac{d\psi}{ds}$  is known), then the bandwidth  $b_f$  of the function can be locally

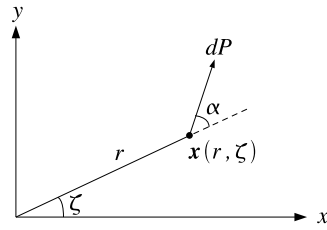


Figure 4.5: Coordinate system of image plane. The vector  $dP$  represents a small shift at angle  $\alpha$  from a point on the image at radius  $r$  from the image centre.

estimated as

$$b_f = \frac{\pi}{\frac{d\psi}{ds}}. \quad (4.42)$$

The term locally is used here as the distribution of the sample points on the sphere may be non-uniform, in which case  $\frac{d\psi}{ds}$ , and consequently  $b_f$  defined in equation 4.42, will be different at different positions on the sphere.

For a wide-angle image, the set of sample points can be defined as the pixels themselves, where a change in sample point  $ds$  is related to a change in pixel position  $dP$ . As a wide-angle image can be mapped to the sphere, the change in angle  $d\psi$  corresponding to a change in pixel position  $dP$  between two pixels can be found. By doing this, it is possible to locally estimate the spherical bandwidth  $b_I$  of a wide-angle image.

With reference to figure 4.5, let  $dP(r, \alpha)$  be the unit pixel shift from the pixel  $\mathbf{x}(r, \zeta) = \mathbf{u} - \mathbf{u}_0$  at radius  $r$  from the principal point  $\mathbf{u}_0$ , where  $\alpha$  is the angle from the line passing through  $\mathbf{u}$  and  $\mathbf{u}_0$ . If  $\mathbf{u}$  maps to the point  $\eta$  on the sphere, and  $\mathbf{u}'$  is the pixel position at shift  $dP(r, \alpha)$  from  $\mathbf{u}$  which maps to the point  $\eta'$  on the sphere, then

$$\frac{d\psi}{dP(r, \alpha)} = \cos^{-1}(\eta^T \eta'), \quad (4.43)$$

where  $d\psi$  is the angle along the great circle passing through  $\eta$  and  $\eta'$ . The bandwidth  $b_I$  of a wide-angle image at some radius  $r$  from the principal point can then be estimated in the sample direction  $\alpha$  as<sup>6</sup>

$$b_I(r, \alpha) = \frac{\pi}{\frac{d\psi}{dP(r, \alpha)}}. \quad (4.44)$$

An algebraic derivation of the image bandwidth can be obtained for some camera

<sup>6</sup>At a given pixel position  $\mathbf{x}$  relative to the principal point, the distance  $dP$  to neighbouring pixels is dependent on the position of the pixel  $\mathbf{x}$  and the sample direction  $\alpha$ — the distance  $dP$  can be as low as 1 pixel, and as high as  $\sqrt{2}$  pixels. As the bandwidth is estimated using a unit pixel shift  $dP = 1$ , the estimate obtained is the minimum possible bandwidth at a given radius  $r$  and sample direction  $\alpha$ .

models. The result for the unified image model is presented here, which is used to model the fisheye camera used throughout this work, and can model the entire class of central projection catadioptric cameras. The only assumption made is that the camera model  $C : M \mapsto \Omega$  defines a mapping between the manifolds  $M$  and  $\Omega$ , here the unit sphere  $\mathbb{S}^2$  and the image respectively. At any point on the sphere  $\eta(\theta, \phi) = (x, y, z)^T = (\sin \theta \cos \phi, \sin \theta \sin \phi, \cos \theta)^T$ , the Euclidean line element on the sphere is  $dl^2 = dx^2 + dy^2 + dz^2$ , where  $d\psi^2 \equiv dl^2$  for small angles  $d\psi$ . The change in angle  $d\psi^2$  can then be parameterised by a change in spherical polar coordinates as

$$d\psi^2 = d\theta^2 + \sin^2 \theta d\phi^2. \quad (4.45)$$

To avoid confusion with the parameters  $l$  and  $m$  which denote the degree and order of the spherical harmonic functions respectively, the unified image model parameters given in equation 2.9 (chapter 2, pg. 40) are denoted  $l_c$  and  $m_c$ . At a given point  $\mathbf{x}(r, \zeta)$  on the image plane defined with respect to the principal point, the variables  $d\theta^2$ ,  $\sin^2 \theta$  and  $d\phi^2$  can be derived for a change in the polar coordinates  $dr$  and  $d\zeta$ . Letting  $n_c = m_c + l_c$ , they are

$$d\phi^2 = d\zeta^2 \quad (4.46)$$

$$\sin^2 \theta = \left( \frac{l_c n_c + \sqrt{r^2(1-l_c^2) + n_c^2}}{r + \frac{n_c^2}{r}} \right)^2 \quad (4.47)$$

$$d\theta^2 = \left( \frac{\left( \frac{r^2(1-l_c^2)}{\sqrt{r^2(1-l_c^2) + n_c^2}} \right) - \left( l_c n_c + \sqrt{r^2(1-l_c^2) + n_c^2} \right) \left( \frac{r^2 - n_c^2}{r^2 + n_c^2} \right)}{(r^2 + n_c^2) \sqrt{1 - r^2 \left( \frac{l_c n_c + \sqrt{r^2(1-l_c^2) + n_c^2}}{r^2 + n_c^2} \right)^2}} \right)^2 dr^2, \quad (4.48)$$

and can then be substituted directly into 4.45 to obtain the expression for change in angle  $d\psi^2$  as a function of the change in polar coordinates  $r, \zeta$  on the image at a some radius  $r$  from the principal point. Referring to figure 4.5, as a small shift  $dP(r, \alpha)$  at radius  $r$  from the principal point at angle  $\alpha$  corresponds to the following changes in polar coordinates in the image plane:

$$dr^2 = \begin{cases} dP^2 & \text{if } r = 0 \\ dP^2 \cos^2 \alpha & \text{if } r > 0 \end{cases}, \quad (4.49)$$

$$d\zeta^2 = \begin{cases} 0 & \text{if } r = 0 \\ \frac{dP^2 \sin^2 \alpha}{r^2} & \text{if } r > 0 \end{cases}, \quad (4.50)$$

an expression for  $d\psi^2/dP^2(r, \alpha)$  can be obtained and the image bandwidth  $b_I(r, \alpha)$

estimated from equation 4.44. One should observe that when  $r = 0$ ,  $\sin^2 \theta = 0$ . Therefore, the equation for  $d\psi^2$  reduces to  $d\psi^2(r=0) = d\theta^2 + 0 = ((l_c + 1)/n_c)^2 dP^2$  which states that, as expected,  $d\psi^2(r=0)$  is independent of the sample direction  $\alpha$ .

The estimated bandwidths of the fisheye camera ( $1024 \times 768$  pixel images) calibrated in chapter 2 and a theoretical parabolic catadioptric camera are shown in figures 4.6a and 4.6b respectively. The camera intrinsic parameters for the parabolic catadioptric camera were selected so that a point on the equator of the view sphere projects to the same radius from the principal point in both the fisheye and parabolic catadioptric images. The angle of colatitude  $\theta$  versus radius  $r$  on the image plane is shown in figure 4.7 for each camera.

It is interesting to observe that the bandwidth for the parabolic catadioptric camera in figure 4.6b appears to be independent of the sample direction  $\alpha$ . Recall that when using the unified image model, a parabolic catadioptric camera is modelled using a point of projection  $l_c = 1$ . Substituting  $l_c = 1$  into equations 4.46, 4.47 and 4.48 simplifies the formulas considerably and gives

$$d\phi^2 = d\zeta^2, \quad (4.51)$$

$$\sin^2 \theta = \frac{4r^2 n_c^2}{(r^2 + n_c^2)^2}, \quad (4.52)$$

$$d\theta^2 = \frac{4n_c^2}{(r^2 + n_c^2)^2} dr^2, \quad (4.53)$$

where  $n_c = m_c + l_c = m_c + 1$ . Substituting equations 4.51, 4.52 and 4.53 into 4.45 gives

$$d\psi^2 = \frac{4n_c^2}{(r^2 + n_c^2)^2} (dr^2 + r^2 d\zeta^2). \quad (4.54)$$

Finally, substituting equations 4.49 and 4.50 into 4.54 gives

$$d\psi^2(r, \alpha)_{l=1} = \frac{4n_c^2}{(r^2 + n_c^2)^2} \left( dP^2 \cos^2 \alpha + r^2 \frac{dP^2 \sin^2 \alpha}{r^2} \right) \quad (4.55)$$

$$= \frac{4n_c^2}{(r^2 + n_c^2)^2} dP^2. \quad (4.56)$$

This result shows that the change in angle  $d\psi$  along any great circle on the sphere for a small shift  $dP(r, \alpha)$  is independent of the sample direction  $\alpha$ , and confirms that the bandwidth for a parabolic catadioptric camera is independent of the sample direction  $\alpha$ . As will be discussed in more detail in section 4.4, the reason for this is the fact that



parabolic catadioptric image formation is equivalent to the stereographic projection of an image on the sphere to the image plane. Stereographic projection is a conformal mapping, and because conformal mappings preserve shapes and distances locally, the bandwidth is independent of  $\alpha$ .

It can be observed in figure 4.6 that both cameras have a maximum image bandwidth in excess of the maximum computationally feasible sample rate of  $b = 512$ . Therefore the spectrum of the images  $\hat{I}_{\mathbb{S}^2}$  obtained using a sample rate  $b = 512$  may contain some form of aliasing.

### 4.3.1.3 Anti-aliasing

The simplest solution to prevent aliasing is to reduce the resolution of the image so that the maximum image bandwidth  $b_I$  is less than the maximum computationally feasible sample rate ( $b = 512$ ). However, for the examples shown in the previous section, this would require reducing the size of the images by a factor greater than 2. Reducing the image resolution is not ideal as it penalises regions in the image with a bandwidth below  $b_I = 512$ . It is proposed that an anti-aliasing interpolation filter can be used to sample the image points and minimise potential aliasing.

For a function  $f$  on the sphere with bandwidth  $b$ ,  $\hat{f}_l^m = 0, \forall l \geq b$ . Making use of the convolution theorem 4.1, the function  $f$  can be bandlimited to  $b$  by convolving it with a symmetrical filter  $h$  with zonal harmonic coefficients

$$\hat{h}_l^0 = \begin{cases} \frac{1}{2\pi} \sqrt{\frac{2l+1}{4\pi}} & \text{if } l < b \\ 0 & \text{if } l \geq b \end{cases}. \quad (4.57)$$

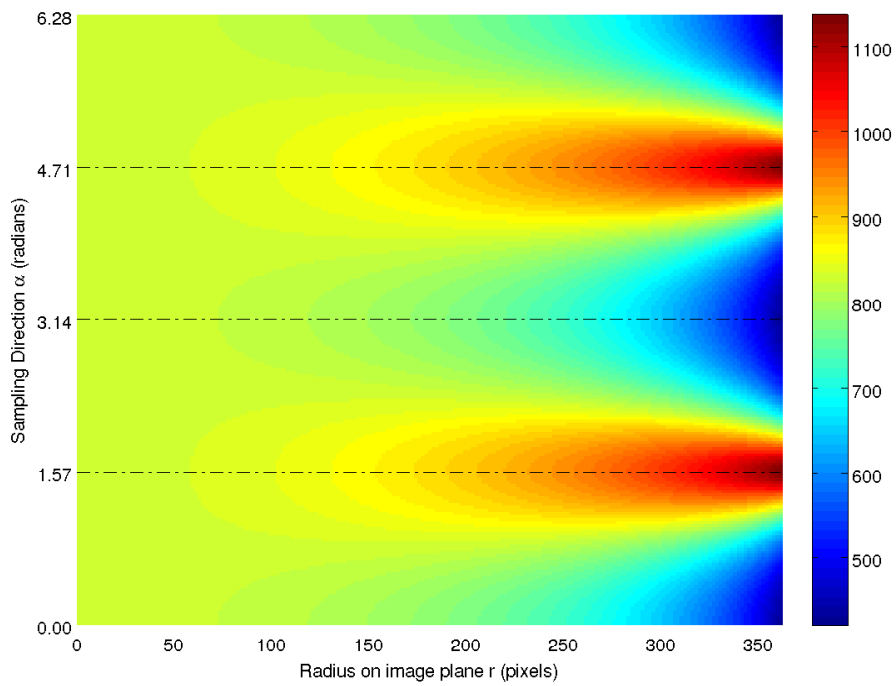
The ideal low-pass filter  $B_b$  obtained from equation 4.57 is

$$B_b(\theta, \phi) = \sum_{l \leq b} h_l^0 Y_l^0(\theta, \phi) \quad (4.58)$$

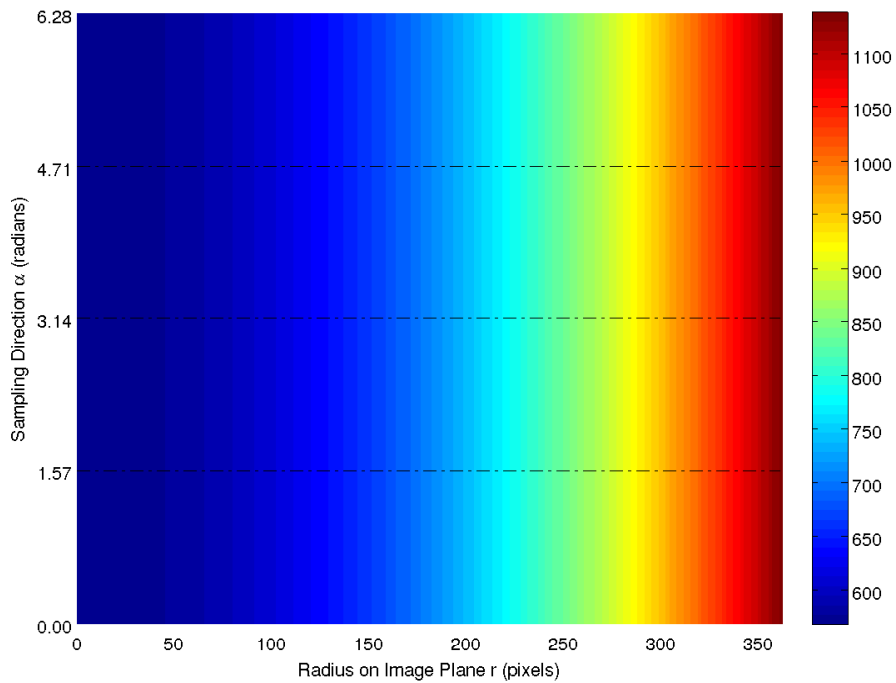
$$= \sum_{l \leq b} \sqrt{\frac{2l+1}{16\pi^2}} Y_l^0(\theta, \phi), \quad (4.59)$$

where  $b$  is the ‘stop band’ frequency. Figure 4.8 shows the filter values versus angle of colatitude  $\theta$  for  $b = 128, 256, 512$ . The filter is centred at the north pole  $\mathbf{n}$  and is similar in appearance to the one-dimensional sinc filter defined in  $\mathbb{R}^1$ .

The anti-aliasing interpolation filter can be used to obtain the set of sample points  $I_{\mathbb{S}^2}(\theta_i, \phi_j)$ , used to find the spectrum  $\hat{I}_{\mathbb{S}^2}$  of the image, where the sample angles  $\theta_i$  and



(a) Fisheye



(b) Parabolic

Figure 4.6: The estimated image bandwidths  $b_I(r, \alpha)$  of the fisheye and parabolic catadioptric cameras.

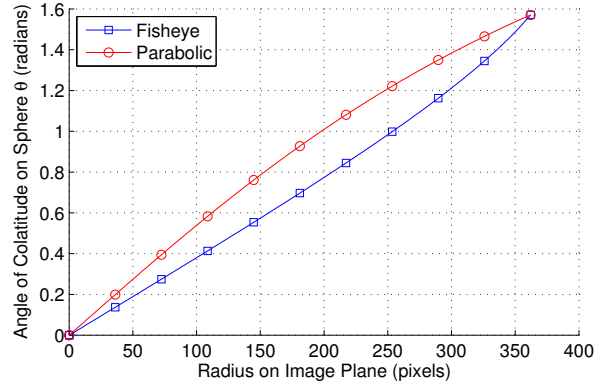
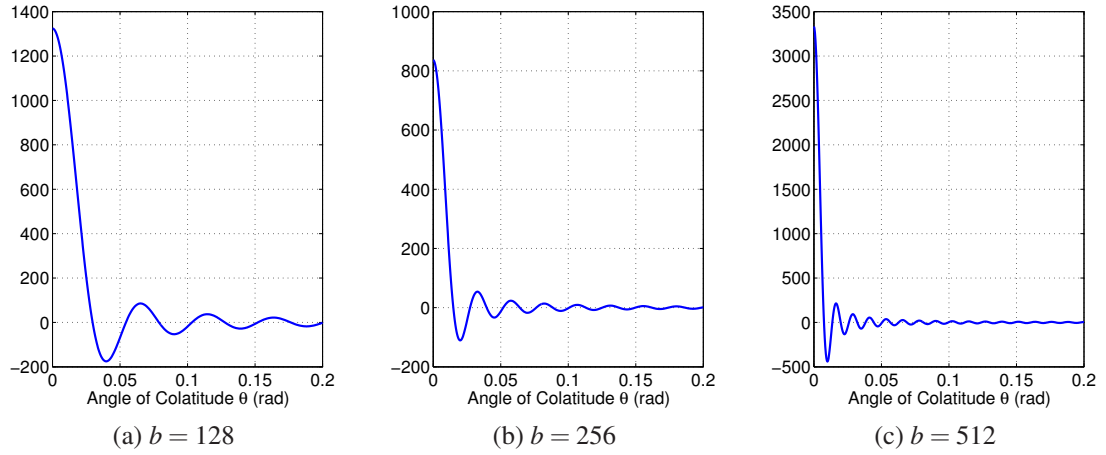


Figure 4.7: Camera model for the fisheye and parabolic catadioptric cameras

Figure 4.8: Values of the anti-aliasing filter  $B_b$  versus angle of colatitude  $\theta$  for stop band frequencies  $b = 128, 256, 512$ .

$\theta_j$  are given in equation 4.38 for a sample rate  $b$ . By definition  $B_b$  is a function on the sphere centred at the north pole  $\mathbf{n}$ . Using a similar analogy described in section 4.2.3.1, let  $B_b(\eta')$  be the filter centred at the point  $\eta'$ , which projects to the function  $\mathcal{B}_b(\eta')$  on the wide-angle image centred at the pixel  $\mathbf{u}' \mapsto \eta'$ . The sample measurement  $I_{\mathbb{S}^2}(\theta_i, \phi_j)$  defined at the point  $\eta'(\theta_i, \phi_j)$  can be obtained as

$$I_{\mathbb{S}^2}(\theta_i, \phi_j) = \sum_{u,v} I(u,v) \mathcal{B}_b(\eta')(u,v), \quad \sum_{u,v} \mathcal{B}_b(\eta')(u,v) = 1, \quad (4.60)$$

which is a convolution operation on the sphere that is implemented on the image. The integration (discrete summation) would need to be taken over all pixel values  $u, v$  to obtain ideal frequency response. As discussed previously in section 4.2.3.1, implementing this type of convolution is computational expensive as a unique kernel  $\mathcal{B}_b(\eta')$  is required for each sample point  $\eta'(\theta_i, \phi_j)$ . To reduce the size of the kernel

$\mathcal{B}'_{b(\eta')}$ , a Blackman window function

$$w(i) = 0.42 - 0.5 \cos\left(\frac{2\pi i}{N-1}\right) + 0.08 \cos\left(\frac{4\pi i}{N-1}\right) \quad (4.61)$$

is applied to the filter  $B_b$  to obtain the new windowed filter  $B'_b(i) = B_b(i)w(i)$ , where  $N$  is selected to include all points up to the fourth zero crossing of the  $B_b$ . The sample measurement  $I_{\mathbb{S}^2}(\theta_i, \phi_j)$  at the point  $\eta'(\theta_i, \phi_j)$  is in practice calculated as

$$I_{\mathbb{S}^2}(\theta_i, \phi_j) = \sum_{u,v \in \mathcal{B}'_{b(\eta')}} I(u, v) \mathcal{B}'_{b(\eta')}(u, v), \quad \sum_{u,v} \mathcal{B}'_{b(\eta')}(u, v) = 1, \quad (4.62)$$

where  $\mathcal{B}'_{b(\eta')}$  is the windowed filter  $B'_b(\eta')$ , centred at  $\eta'$ , projected to the image plane. The integral in 4.62 is only taken over the pixels  $u, v$  within the kernel  $\mathcal{B}'_{b(\eta')}$  and not all possible pixel positions. Figure 4.9 shows the comparison of the ideal low pass filter and the windowed filter  $B'_b$  for a bandwidth  $b = 256$ . The theoretical frequency response of the filter is shown in figure 4.10. Using the windowed filter  $B'_b$  gives improved computational efficiency at the expense of non-ideal frequency response.

The magnitude of the spectrum  $\hat{I}_{\mathbb{S}^2}$  of a fisheye image obtained with and without the anti-aliasing filter with is shown in figure 4.11. When the filter is not used, the image values are sampled using a simple linear interpolation in the image plane. The spectrums were obtained using a sample rate  $b = 512$ , and the ‘stop band’ frequency of the filter  $B'_b$  was set to 256. The bandwidth of the  $1024 \times 768$  pixel fisheye image was shown previously in figure 4.6a. The magnitude of the spectrum for each spherical harmonic degree  $l$  is

$$mag(l) = \sum_{|m| \leq l} \sqrt{\frac{4\pi}{2l+1}} (\hat{I}_{\mathbb{S}^2})_l^m \overline{(\hat{I}_{\mathbb{S}^2})_l^m}. \quad (4.63)$$

Although ideal frequency response cannot be achieved, there is a considerable reduction in the magnitude above the stop band. Later experiments will compare the relative performance of keypoint detection in wide-angle images using sSIFT with and without the use of the anti-aliasing filter.

### 4.3.2 Scale Selection

The scales  $\sigma$  used by SIFT define the standard deviation of the Gaussians  $G(\cdot; \sigma)$  used to obtain the set of scale-space images  $L(\cdot; \sigma)$ . As  $\sigma$  is a measure of the standard deviation in pixels, the scales  $\sigma$  used are appropriate for an image of arbitrary size. The

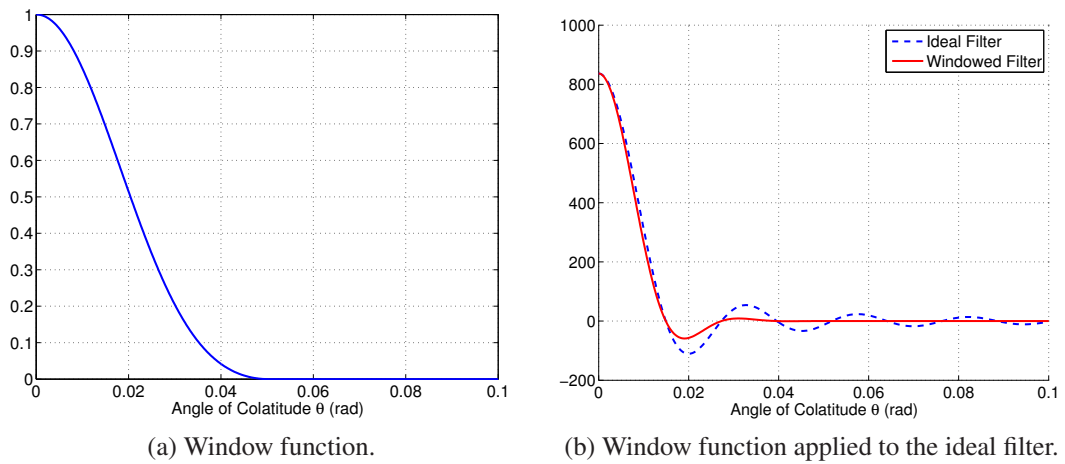


Figure 4.9: The (a) window function and (b) the ideal filter before and after the application of the window function.

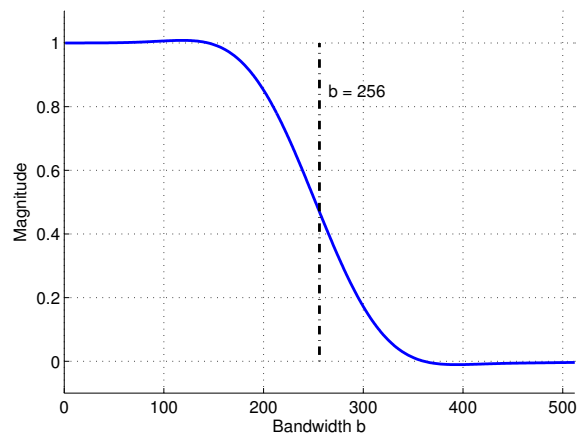


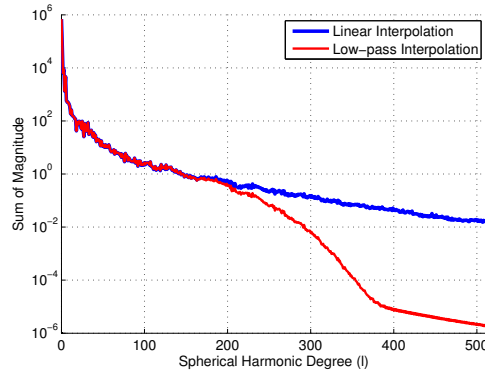
Figure 4.10: Theoretical zonal harmonic coefficients of the windowed anti-aliasing filter.

scale  $kt$  of the spherical Gaussian  $G_{\mathbb{S}^2}(\cdot; kt)$  can be interpreted as a value proportional to the variance of the spherical Gaussian measured in radians. Selecting some fixed set of scales  $kt$  for sSIFT is not suitable as no account is made for the size of the original image or field of view of the camera. The scales used by sSIFT are therefore selected as a function of the scales  $\sigma$  used by SIFT and the camera intrinsic parameters.

The scales  $\sigma$  used by SIFT are defined by four parameters; an assumed initial scale  $\sigma_{input}$  of the original image  $I$ , a starting scale  $\sigma_0$ , a scale multiplication factor (number of scales per octave  $n_{spo}$ ), and the number of octaves  $n_{oct}$  used. SIFT first doubles the original image size and assumes that this image has a starting scale  $\sigma_{input} = 0.5$ . The starting scale is set to  $\sigma_0 = 1.6$ , where the first scale-space image is obtained by



(a) Input fisheye image.



(b) Magnitude of the image spectrum obtained using a simple linear interpolation, and with the windowed anti-aliasing filter.

Figure 4.11: The effect of the anti-aliasing interpolation filter on the magnitude of the spectrum of the fisheye image in (a). The ‘stop band’ frequency  $b$  was set to  $b = 256$ .

pre-smoothing the double size image to this scale:

$$L(\cdot; \sigma_0) = G(\cdot; \sigma_0 - input) * I, \quad (4.64)$$

where  $\sigma_0 - input = \sqrt{\sigma_0^2 - \sigma_{input}^2}$ . SIFT selects  $n_{spo} = 3$  scales per octave and used the scales

$$\sigma_i = \sigma_0 2^{\frac{i}{n_{spo}}} \quad i \in \{0, 1, 2, \dots, n_{oct} n_{spo} + 2\}, \quad (4.65)$$

where  $n_{oct}$  is dependent on the original image size. The scales  $kt$  used by sSIFT are selected using a similar method.

#### 4.3.2.1 Input and Initial Scale

For the Gaussian function, the ratio of the amplitude  $G(x = \sigma, \sigma) / G(0, \sigma)$  is

$$\frac{G(x = \sigma, \sigma)}{G(0, \sigma)} = \frac{e^{-\left(\frac{\sigma^2}{2\sigma^2}\right)}}{e^{-\left(\frac{0}{2\sigma^2}\right)}} \quad (4.66)$$

$$= e^{-0.5} \quad \forall \sigma > 0. \quad (4.67)$$

It is of interest to then consider is a similar property holds for the spherical Gaussian for which

$$\frac{G_{\mathbb{S}^2}(\theta = f(kt); kt)}{G_{\mathbb{S}^2}(0; kt)} = e^{-0.5}, \quad \forall kt > 0, \quad (4.68)$$

where  $f(kt)$  is some function of the scale  $kt$ . If a point in a wide-angle image at a distance  $x_s = 1$  pixel from the principal point projects to a point on the sphere with

an angle of colatitude  $\theta_s$ , then finding a solution to equation 4.68 will provide some means for comparing the shapes of  $G$  and  $G_{\mathbb{S}^2}$  with respect to the values  $x_s$  and  $\theta_s$  respectively. This can then be used to select suitable scales  $kt$  based on the values used by SIFT.

Although a closed for solution to 4.68 has not been found, the relationship  $\theta = f(kt)$  can be obtained empirically using a non-linear optimisation. Referring to equation 4.68, it is more efficient to try and evaluate the function  $kt = f(\theta)$  for which

$$\frac{G_{\mathbb{S}^2}(\theta; kt = f(\theta))}{G_{\mathbb{S}^2}(0; kt = f(\theta))} = e^{-0.5}. \quad (4.69)$$

Recalling that the spherical Gaussian can be written as the spherical harmonic expansion

$$G_{\mathbb{S}^2}(\theta, \phi; kt) = \sum_{l \in \mathbb{N}} \sqrt{\frac{2l+1}{4\pi}} Y_l^0(\theta, \phi) e^{-l(l+1)kt}, \quad (4.70)$$

the condition in 4.69 can be written as

$$\frac{G_{\mathbb{S}^2}(\theta; kt = f(\theta))}{G_{\mathbb{S}^2}(0; kt = f(\theta))} = \frac{\sum_{l \in \mathbb{N}} \sqrt{\frac{2l+1}{4\pi}} Y_l^0(\theta, \phi) e^{-l(l+1)kt=f(\theta)}}{\sum_{l \in \mathbb{N}} \sqrt{\frac{2l+1}{4\pi}} Y_l^0(0, \phi) e^{-l(l+1)kt=f(\theta)}} = e^{-0.5}, \quad (4.71)$$

and can be simplified to

$$\frac{\sum_{l \in \mathbb{N}} \frac{2l+1}{4\pi} P_l^0(\cos(\theta)) e^{-l(l+1)kt=f(\theta)}}{\sum_{l \in \mathbb{N}} \frac{2l+1}{4\pi} e^{-l(l+1)kt=f(\theta)}} = e^{-0.5}. \quad (4.72)$$

For some angle  $\theta$ , the scale  $kt$  is found as

$$\operatorname{argmin}_{kt>0} f(kt) = \left( e^{-0.5} - \frac{\sum_{l \in \mathbb{N}} \frac{2l+1}{4\pi} P_l^0(\cos(\theta)) e^{-l(l+1)kt}}{\sum_{l \in \mathbb{N}} \frac{2l+1}{4\pi} e^{-l(l+1)kt}} \right)^2. \quad (4.73)$$

Figure 4.12 shows the resulting scale  $\sqrt{kt}$  found for a range of initial angles  $\theta$ . The results were found for the spherical Gaussian computed up to  $b = 2048$ . A straight lines passing through the origin was fitted using a least squares minimisation. The gradient of the line was found to be  $m = 0.707 \approx \frac{1}{\sqrt{2}}$ , where  $kt = \frac{\theta^2}{2}$ .

As SIFT first doubles the image size and sets  $\sigma_{input} = 1.0$  and  $\sigma_0 = 1.6$ , these scales are 0.5 and 0.8 with respect to the original sized image. The input scale  $kt_{input}$  and starting scale  $kt_0$  used by sSIFT are set as

$$kt_{input} = \frac{(\sigma_{input} \theta_s)^2}{2}, \quad \sigma_{input} = 0.5, \quad (4.74)$$

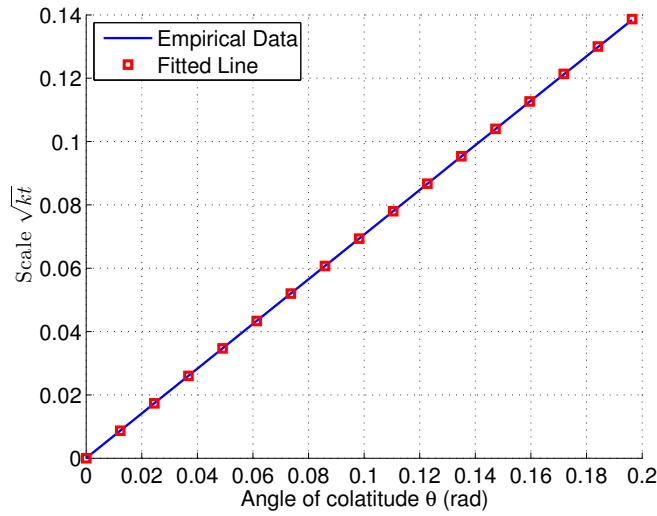


Figure 4.12: Fitted line obtained via least squares minimisation

and

$$kt_0 = \frac{(\sigma_0 \theta_s)^2}{2}, \quad \sigma_0 = 0.8, \quad (4.75)$$

where the angle of colatitude  $\theta_s$  corresponding to a point in the wide-angle image at a distance  $x_s = 1$  pixel from the principal point is dependent on the camera intrinsic parameters.

Figure 4.13 compares the appearance of the Gaussian  $G(\cdot; \sigma)$  and the spherical Gaussian  $G_{\mathbb{S}^2}(\cdot; kt)$  projected to the function  $G_{\mathbb{S}^2}(\cdot; kt)$  on the image plane for the fish-eye and parabolic catadioptric cameras in figure 4.7 (the angle  $\theta_s$  is different for each camera). The results are shown using two different scales of  $\sigma = 1$  and  $\sigma = 20$ . In both cases, the scales  $kt$  were obtained from equations 4.74 and 4.75 as  $kt = \frac{(\sigma \theta_s)^2}{2}$ . The similarity in the appearance of  $G(\cdot; \sigma)$  and  $G_{\mathbb{S}^2}(\cdot; kt)$  for each camera and scale  $\sigma$  suggests that this is an appropriate way to select the input scale  $kt_{input}$  and initial scale  $kt_0$  for sSIFT. It is interesting to note that the inverse stereographic projection of the Gaussian  $G(\cdot; \sigma)$  to the sphere has been used as a smoothing kernel on the sphere [56]. The inverse stereographic projection of the Gaussian and its derivatives to the sphere have also been used for wavelet based analysis of functions on the sphere [5, 247, 152].

#### 4.3.2.2 Scales per octave and number of octaves

sSIFT uses the same  $n_{oct} = 3$  number of scales per octave as SIFT. The set of scales  $kt$  used by sSIFT are

$$kt_i = \left( \sqrt{kt_0} 2^{\frac{i}{n_{spo}}} \right)^2, \quad i \in \{0, 1, \dots, n_{oct} n_{spo} + 2\}, \quad (4.76)$$



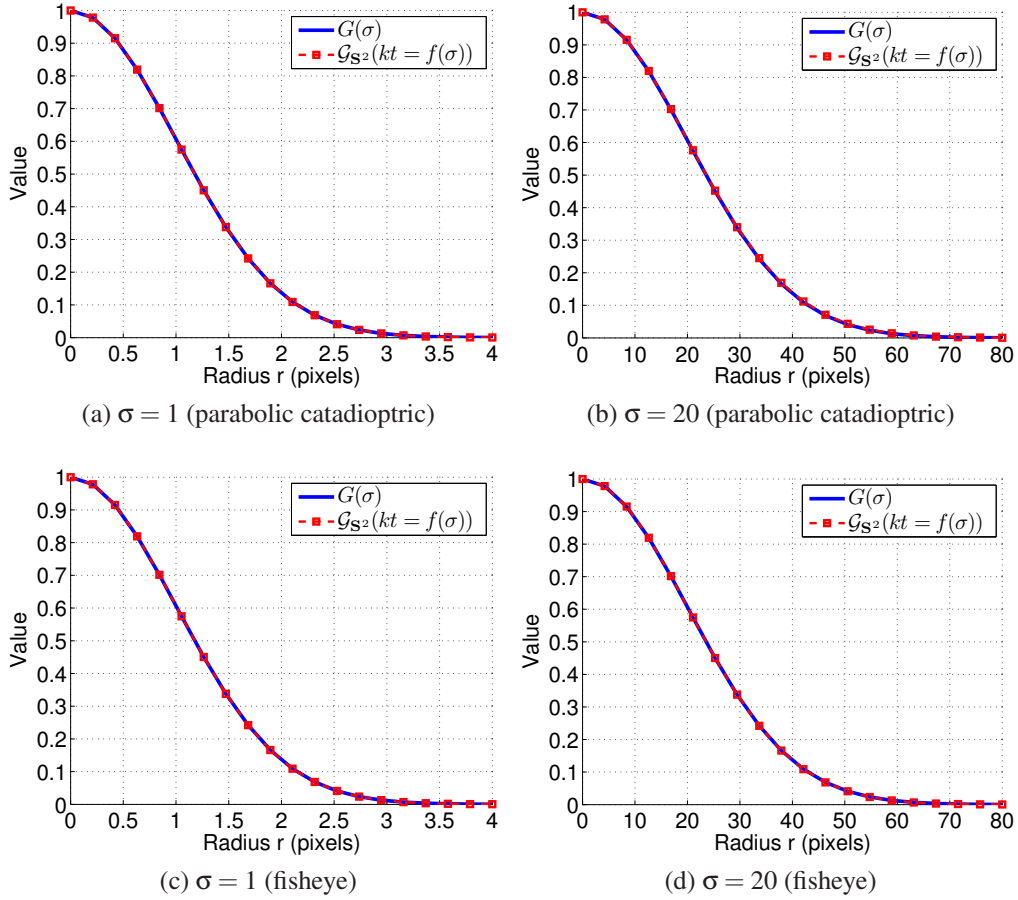


Figure 4.13: Comparison of  $G(\cdot; \sigma)$  and  $G_{\mathbb{S}^2}(\cdot; kt = (\sigma \theta_s)^2/2)$  for the parabolic catadioptric (top row) and fisheye (bottom row) cameras for two different scales  $\sigma$ .

where  $n_{oct}$  is the number of octaves used. Although the same octave based approach used by SIFT is not implemented (keypoints are detected using the set of scale-space images  $\mathcal{L}_{\mathbb{S}^2}(\cdot; kt)$  that are all the same size as the original wide-angle image  $I$ ), it is still convenient to use this term. In all proceeding experiments, sSIFT detects keypoints in the first  $n_{oct} = 5$  octaves of scale-space. This means that keypoints are detected in a total of  $n_{oct} \times n_{spo}$  difference of Gaussian images. As keypoints are local extrema in scale and space, this requires finding  $n = n_{oct} \times n_{spo} + 2$  difference of Gaussian images from a total of  $n = n_{oct} \times n_{spo} + 3$  scale-space images.

### 4.3.3 Obtaining the Scale-Space Images

The procedure used to find the set of scale-space images  $\mathcal{L}_{\mathbb{S}^2}(\cdot; kt)$  from a wide-angle image  $I$  is illustrated in figure 4.14 and includes the following steps:

1. Sample the image at points  $I_{\mathbb{S}^2}(\theta_i, \phi_j)$  defined in equation 4.38 for a sample rate

- b.* If the maximum image bandwidth  $b_I$  exceeds the maximum permissible sample rate, use the anti-aliasing interpolation filter  $B'_b$  to obtain the sample points.
2. Use s2kit to obtain the spectrum of the image  $\hat{I}_{\mathbb{S}^2} = SFT(I_{\mathbb{S}^2})$ .
  3. Set  $\hat{L}_{\mathbb{S}^2}(0) = \hat{I}_{\mathbb{S}^2}$  as the initial condition.
  4. For the set of all scales  $kt_{0,1,\dots,n_{oct}n_{spo}+2}$ , find the scale-space representation  $\hat{L}_{\mathbb{S}^2}(kt)$  of the image as a response in the spherical Fourier domain from equation 4.36.
  5. Find the inverse SFT of all  $\hat{L}_{\mathbb{S}^2}(kt)$  to obtain  $L_{\mathbb{S}^2}(\cdot;kt) = ISFT(\hat{L}_{\mathbb{S}^2}(kt))$ . The output is the set of scale-space images  $L_{\mathbb{S}^2}(\cdot;kt)$  on an equiangular  $\theta, \phi$  grid — the angles  $\theta, \phi$  are the same sample points in equation 4.38.
  6. Map each scale-space image  $L_{\mathbb{S}^2}(\cdot;kt)$  back to the original image plane to obtain  $\mathcal{L}_{\mathbb{S}^2}(\cdot;kt)$ . The scale-space images  $\mathcal{L}_{\mathbb{S}^2}(\cdot;kt)$  are the same size as the original wide-angle image  $I$  irrespective of the scale  $kt$ .

It is important to observe that sSIFT defines the initial condition  $\hat{L}_{\mathbb{S}^2}(0) = \hat{I}_{\mathbb{S}^2}$ , whereby  $(\hat{L}_{\mathbb{S}^2})_l^m(kt_0) = (\hat{I}_{\mathbb{S}^2})_l^m e^{-l(l+1)kt_0}$  — the input scale  $kt_{input}$  is set to zero. If the input scale defined in equation 4.74 were used, then  $(\hat{L}_{\mathbb{S}^2})_l^m(kt_0) = (\hat{I}_{\mathbb{S}^2})_l^m e^{-l(l+1)kt_0-input}$ , where  $kt_0-input = kt_0 - kt_{input}$ . The former is used as it equates to the convolution of the image with a spherical Gaussian of increased scale. This has the ability to suppress more aliasing artifacts which may exist in the high frequency components of the spectrum  $\hat{I}_{\mathbb{S}^2}$  — recall that the anti-aliasing filter  $B'_s$  is unable to achieve ideal frequency response.

#### 4.3.4 Keypoint Detection

Candidate keypoints are selected as local extrema in the difference of Gaussian (scale-space) images  $\mathcal{D}_{\mathbb{S}^2}(\cdot;kt_i) = \mathcal{L}_{\mathbb{S}^2}(\cdot;kt_{i+1}) - \mathcal{L}_{\mathbb{S}^2}(\cdot;kt_i)$  with absolute value  $\mathcal{D}_{\mathbb{S}^2}(u, v; kt)$  greater than 0.8 times the difference of Gaussian threshold. In the following experiments a range of thresholds are used. The 0.8 factor is used as the locations of the extrema and difference of Gaussian values at the extrema are interpolated later. A candidate keypoint is a local extrema if the magnitude of its difference of Gaussian value is greater than the magnitude its nearest 26 pixels in the current and adjacent difference of Gaussian images (see figure 3.8, pg.118).

Edge responses are removed by enforcing a minimum ratio  $r_{edge}$  between the magnitudes of the maximum and minimum principal curvatures of  $\mathcal{D}_{\mathbb{S}^2}(\cdot;kt)$  evaluated at

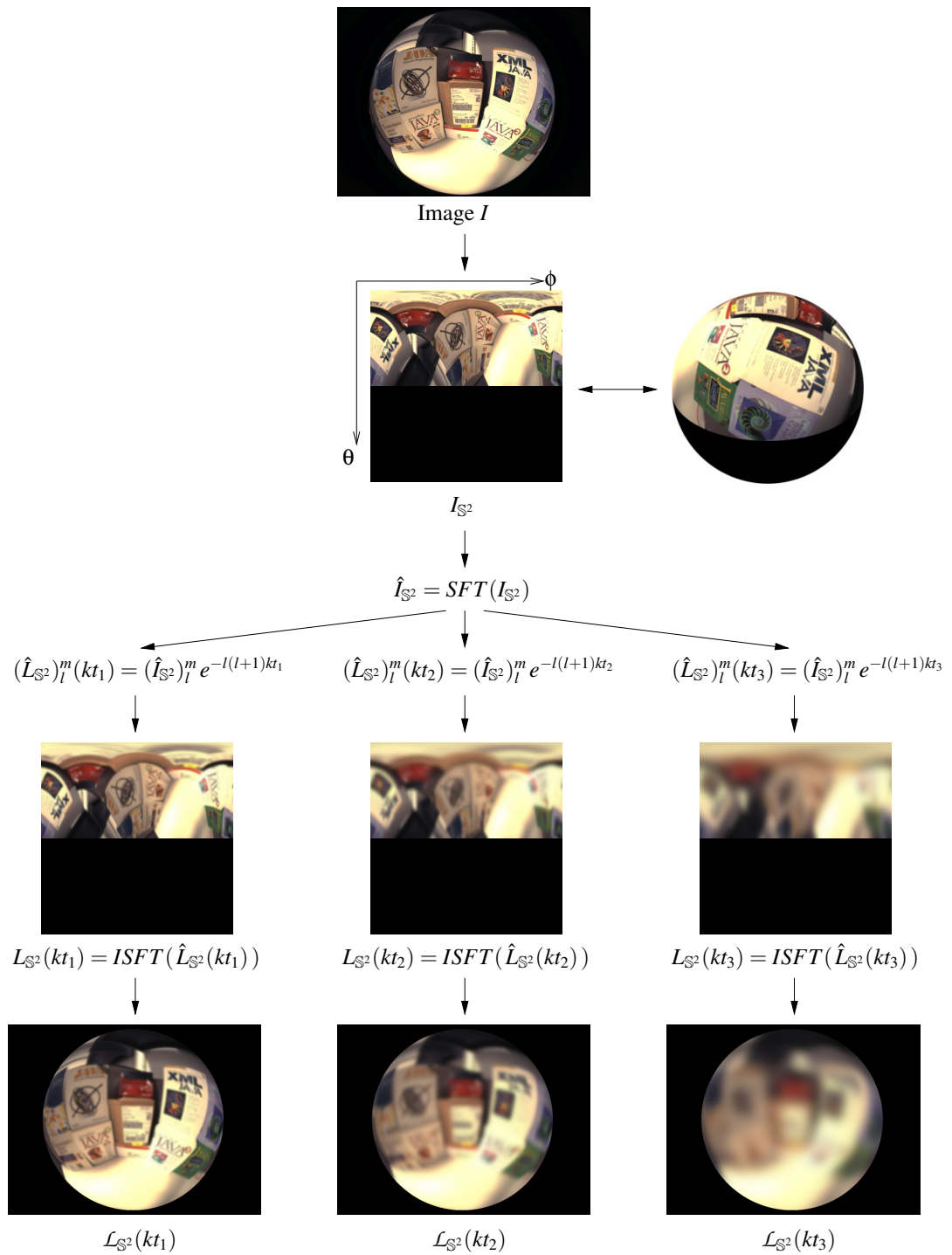


Figure 4.14: General procedure used by sSIFT to find the scale-space images  $L_{\mathbb{S}^2}(\cdot; kt)$ . SFT is a forward discrete spherical Fourier transform, and ISFT is an inverse discrete spherical Fourier transform.

the pixel positions  $\mathbf{u}$  of the candidate keypoints. The difference of Gaussian image  $\mathcal{D}_{\mathbb{S}^2}(\cdot; kt)$  is assumed to be locally perspective during edge removal, where the ratio of principal curvatures are obtained from the Hessian matrix  $\mathcal{H}$ :

$$\mathcal{H} = \begin{bmatrix} \mathcal{D}_{\mathbb{S}^2}(u, v; kt)_{uu} & \mathcal{D}_{\mathbb{S}^2}(u, v; kt)_{uv} \\ \mathcal{D}_{\mathbb{S}^2}(u, v; kt)_{uv} & \mathcal{D}_{\mathbb{S}^2}(u, v; kt)_{vv} \end{bmatrix} \quad (4.77)$$

where

$$\mathcal{D}_{\mathbb{S}^2}(u, v; kt)_{uu} = \mathcal{D}_{\mathbb{S}^2}(u + 1, v; kt) + \mathcal{D}_{\mathbb{S}^2}(u - 1, v; kt) - \mathcal{D}_{\mathbb{S}^2}(u, v; kt), \quad (4.78)$$

$$\mathcal{D}_{\mathbb{S}^2}(u, v; kt)_{vv} = \mathcal{D}_{\mathbb{S}^2}(u, v + 1; kt) + \mathcal{D}_{\mathbb{S}^2}(u, v - 1; kt) - \mathcal{D}_{\mathbb{S}^2}(u, v; kt), \quad (4.79)$$

and

$$\mathcal{D}_{\mathbb{S}^2}(u, v; kt)_{uv} = \frac{1}{4} [(\mathcal{D}_{\mathbb{S}^2}(u + 1, v + 1; kt) - \mathcal{D}_{\mathbb{S}^2}(u - 1, v + 1; kt)) - \quad (4.80)$$

$$(\mathcal{D}_{\mathbb{S}^2}(u + 1, v - 1; kt) - \mathcal{D}_{\mathbb{S}^2}(u - 1, v - 1; kt))]. \quad (4.81)$$

A candidate keypoint is deemed not to be an edge response, and is retained if

$$\frac{\text{trace}(\mathcal{H})^2}{\det(\mathcal{H})} < \frac{(r + 1)^2}{r_{edge}} \quad (4.82)$$

for the threshold  $r_{edge} = 10$ .

The accuracy of a keypoint's scale and location is improved using the quadratic interpolation scheme developed by Brown and Lowe [29], and is the same scheme used by SIFT. If the position and scale of the keypoint lies at the origin of the difference of Gaussian function  $\mathcal{D}_{\mathbb{S}^2}$ , the difference of Gaussian function is estimated at an offset  $\mathbf{x} = (u, v, i)^T$  from the origin, where  $i$  is the scale index, from the quadratic Taylor expansion

$$\mathcal{D}_{\mathbb{S}^2}(\mathbf{x}) = \mathcal{D}_{\mathbb{S}^2} + \frac{\partial \mathcal{D}_{\mathbb{S}^2}}{\partial \mathbf{x}}^T \mathbf{x} + \frac{1}{2} \mathbf{x}^T \frac{\partial^2 \mathcal{D}_{\mathbb{S}^2}}{\partial \mathbf{x}^2} \mathbf{x}, \quad (4.83)$$

where the pixel values used to compute both  $\frac{\partial \mathcal{D}_{\mathbb{S}^2}}{\partial \mathbf{x}}$  and  $\frac{\partial^2 \mathcal{D}_{\mathbb{S}^2}}{\partial \mathbf{x}^2}$  are given in appendix C. Note that the difference of Gaussian function  $\mathcal{D}_{\mathbb{S}^2}$  is assumed to be locally perspective when computing the derivatives. The estimated offset  $\mathbf{x}' = (u', v', i')^T$  at which the difference of Gaussian function  $\mathcal{D}_{\mathbb{S}^2}$  is an extrema is found by taking the derivative of equation 4.83,

$$\frac{\partial \mathcal{D}_{\mathbb{S}^2}(\mathbf{x}')}{\partial \mathbf{x}} = \frac{\partial \mathcal{D}_{\mathbb{S}^2}}{\partial \mathbf{x}} + \frac{\partial^2 \mathcal{D}_{\mathbb{S}^2}}{\partial \mathbf{x}^2} \mathbf{x}', \quad (4.84)$$

and setting the value to zero to obtain

$$\mathbf{x}' = -\frac{\partial^2 \mathcal{D}_{\mathbb{S}^2}}{\partial \mathbf{x}^2}^{-1} \frac{\partial \mathcal{D}_{\mathbb{S}^2}}{\partial \mathbf{x}}, \quad (4.85)$$

which is solved using Gaussian elimination. If either of the pixel offset values  $u'$  or  $v'$  exceed  $\pm 0.5$ , the location of the keypoint position  $\mathbf{u}$  is shifted accordingly and the process repeated up to a fixed number of trials. The estimate of the difference of Gaussian function at the offset  $\mathbf{x}'$  is

$$\mathcal{D}_{\mathbb{S}^2}(\mathbf{x}') = \mathcal{D}_{\mathbb{S}^2} + \frac{1}{2} \frac{\partial \mathcal{D}_{\mathbb{S}^2}}{\partial \mathbf{x}}^T \mathbf{x}'. \quad (4.86)$$

The keypoint is retained if the absolute value of  $\mathcal{D}_{\mathbb{S}^2}(\mathbf{x}')$  is above the initial set threshold. The final position of the keypoint is  $u + u', v + v'$ , and the characteristic scale  $kt$  is

$$kt = \left( \sqrt{kt_0 \frac{i+i'}{n_{spo}}} \right)^2, \quad (4.87)$$

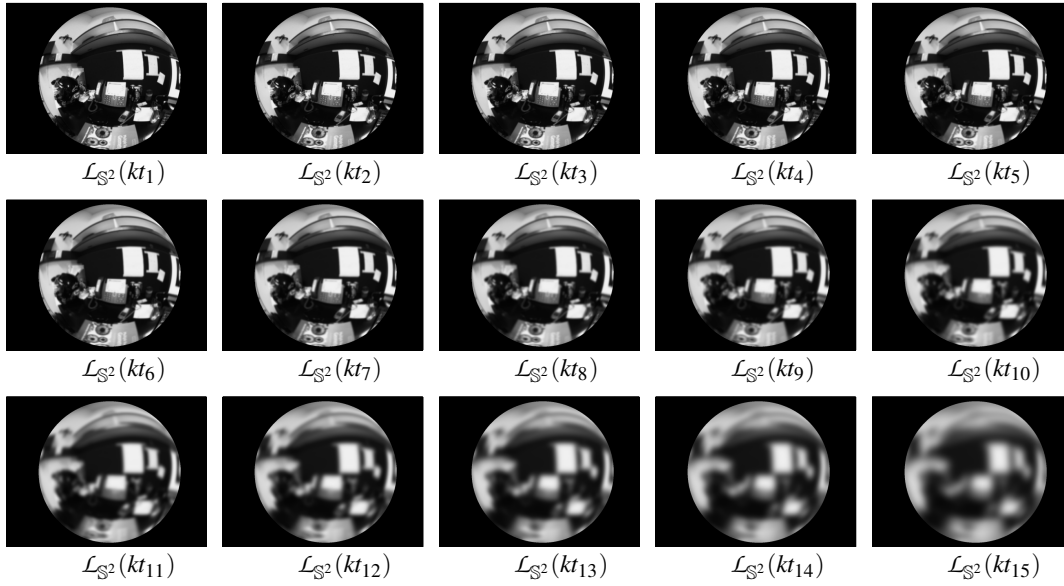
where  $kt_0$  is the initial scale, and  $i$  is the index for the difference of Gaussian image  $\mathcal{D}_{\mathbb{S}^2}(\cdot; kt_i)$  in which the keypoint was detected.

Figure 4.15a illustrates the first 15 scale-space images  $\mathcal{L}_{\mathbb{S}^2}(\cdot; kt)$  obtained for a fisheye image. The sSIFT keypoints detected in the difference of Gaussian images  $\mathcal{D}_{\mathbb{S}^2}(\cdot; kt)$  are shown in figure 4.15b. No keypoints can be detected in the first or last difference of Gaussian images as they need to be local extrema compared to the adjacent difference of Gaussian images. Although only the position of the keypoints is shown, each keypoint has some characteristic scale  $kt$ .

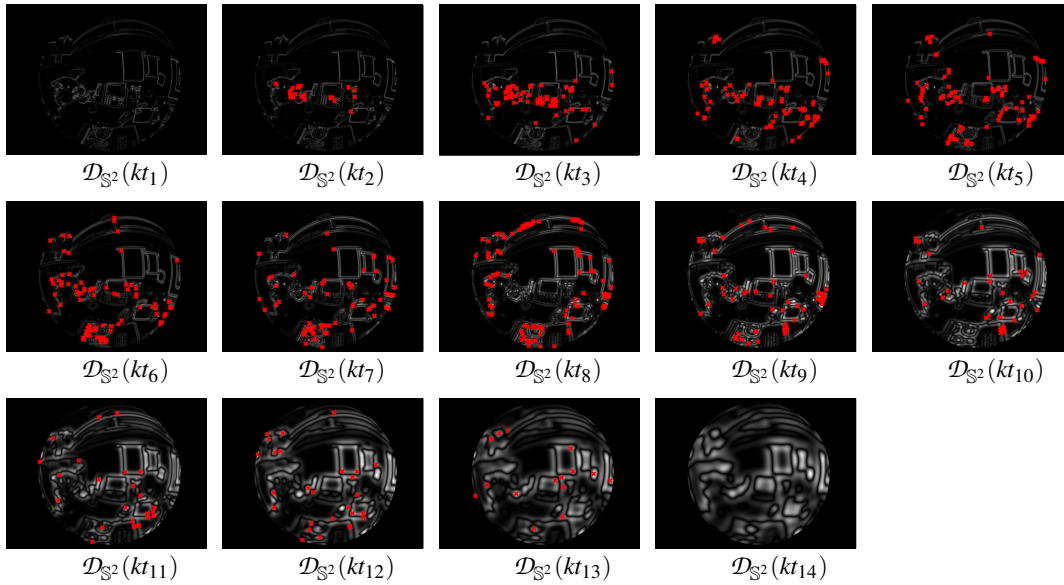
### 4.3.5 Keypoint Support Region

A descriptor for an sSIFT keypoint is evaluated from the local image content within the keypoint's support region. For a keypoint detected at a pixel position  $\mathbf{u}$ , which maps to the point  $\eta$  on the unit sphere, the boundary of this support region is defined by an angle  $\psi_s$ , as illustrated in figure 4.16. This is an angle from the axis passing through the centre of the sphere and the point  $\eta$ . A circular support region on the sphere is used as the spherical Gaussian is an isotropic function on the sphere. The angle  $\psi_s$  is set relative to a keypoint's characteristic scale  $kt$  as

$$\psi_s = c \sqrt{2kt}, \quad (4.88)$$



(a) The set of scale-space images  $\mathcal{L}_{S^2}(\theta, \phi; kt) \mapsto \mathcal{L}_{S^2}(u, v; kt)$  mapped to the original fisheye image plane.



(b) The difference of Gaussian (neighbouring scale-space) images  $\mathcal{D}_{S^2}(u, v; kt)$  used for keypoint detection. The red dots are the location of the detected keypoints.

Figure 4.15: Keypoint detection using sSIFT in a fisheye image. The set of difference of Gaussian images  $\mathcal{D}_{S^2}(\cdot; kt)$  are obtained from the set of scale-space images  $\mathcal{L}_{S^2}(\cdot; kt)$ , where  $\mathcal{D}_{S^2}(\cdot; kt_i) = \mathcal{L}_{S^2}(\cdot; kt_{i+1}) - \mathcal{L}_{S^2}(\cdot; kt_i)$ . The red dots show the position of the sSIFT keypoints — keypoints cannot be detected in the first and last difference of Gaussian images. The absolute values of the difference of Gaussian images  $\mathcal{D}_{S^2}(\cdot; kt)$  have been shown for visualisation purposes.

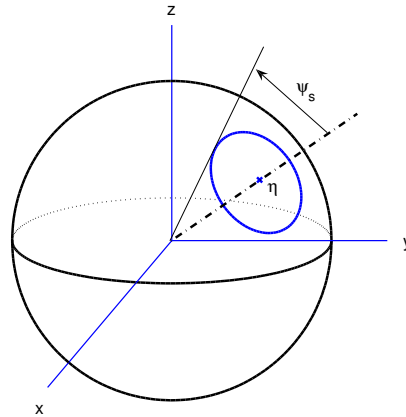


Figure 4.16: The boundary of the support region for an sSIFT is defined as a circle on the sphere, centred about the position  $\eta$  of the keypoint. The size of the circle is parameterised by the angle  $\psi_s = c\sqrt{2kt}$ , where  $kt$  is the characteristic scale of the keypoint and  $c$  is some constant.

where  $c$  is some constant. The factor  $\sqrt{2kt}$  is used as it was observed in section 4.3.2.1 that

$$\frac{G(x = \sigma, \sigma)}{G(0, \sigma)} \approx \frac{G_{S^2}(\theta = \sqrt{2kt}; kt)}{G_{S^2}(0; kt)}. \quad (4.89)$$

Therefore, for any scale-invariant keypoint detection algorithm which sets the support region as a circle on the image with radius  $r_s = c\sigma$ , where  $\sigma$  is the characteristic scale of the keypoint, substituting the same value for  $c$  in 4.88 gives some guide to selecting a suitable sized support region on the sphere.

Figure 4.17 illustrates the keypoint support regions for a subset of the sSIFT keypoints detected in the example in figure 4.15. The support regions are shown on the fisheye image in figure 4.17a, and on the unit sphere in figure 4.17b. The value of the constant used was  $c = 10$ .

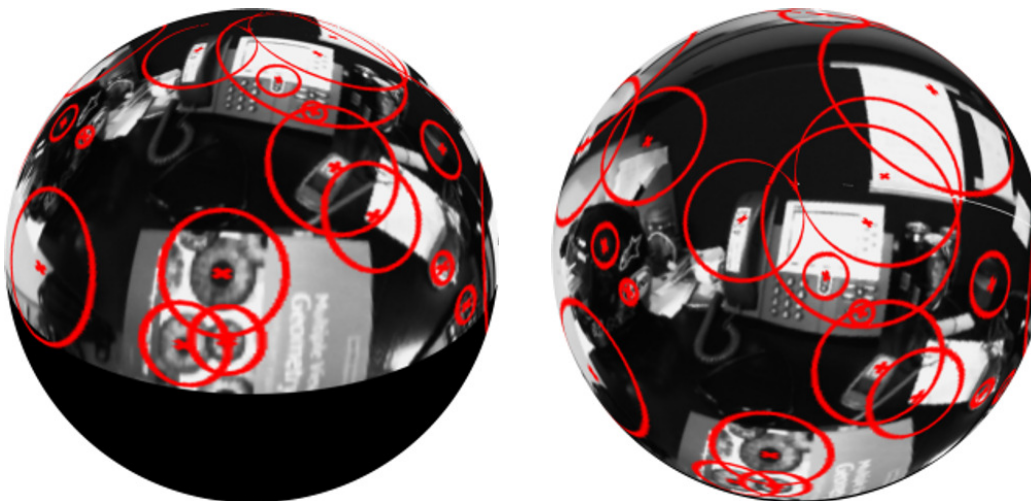
### 4.3.6 Experiments

The experiments in this section compare the percentage correlation and number of correct keypoint correspondences found between synthetically generated wide-angle images pairs using SIFT and sSIFT. This comparison serves as an initial guide to the relative performance of sSIFT versus SIFT without dependence on the type of keypoint descriptor used or the method of keypoint matching (i.e. the distance metric used to assess descriptor similarity).





(a) Keypoint support regions as they appear on the fisheye image.



(b) Keypoint support regions as they appear on the unit view sphere. The figure on the right is taken at a different viewpoint (change in rotation) to the figure on the left.

Figure 4.17: A subset of sSIFT keypoints detected in the example in figure 4.15 and their associated support regions. The support regions are shown on the fisheye image in (a), and on the view sphere in (b). The size of the support regions has been set to  $\psi_s = 10\sqrt{2kt}$ , where  $kt$  is the characteristic scale of a keypoint.  $\psi_s$  is an angle as indicated in figure 4.16.



### 4.3.6.1 Input Data

The images used in the experiments are wide-angle parabolic catadioptric and fisheye images generated synthetically from the set of 40 high resolution ( $2272 \times 1704$ pixel) reference images shown in figure 4.18. Synthetically generated images are used to ensure that correct keypoint correspondences can be identified reliably — the exact transformation (homography) between any two of the synthetically generated wide-angle images is known exactly. Furthermore, results can be found for any wide-angle camera.

The wide-angle images are obtained as follows. Each pixel  $\mathbf{u}$  on the wide-angle image is mapped to a point  $\eta$  on the sphere. This point is then rotated to a new position  $\eta' = R\eta$  by a rotation  $R = R_y(\beta)R_x(\alpha)$ , where  $R_y(\beta)$  and  $R_x(\alpha)$  are rotations about the  $y$  and  $x$  axes respectively:

$$R_y(\beta) = \begin{bmatrix} \cos\beta & 0 & \sin\beta \\ 0 & 1 & 0 \\ -\sin\beta & 0 & \cos\beta \end{bmatrix} \quad R_x(\alpha) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha \\ 0 & \sin\alpha & \cos\alpha \end{bmatrix}. \quad (4.90)$$

Each point  $\eta'$  is then projected to the point  $\mathbf{x}_p$  on the perspective plane by

$$\mathbf{x}_p = d \begin{bmatrix} \frac{\eta'_x}{\eta'_z} \\ \frac{\eta'_y}{\eta'_z} \end{bmatrix}, \quad (4.91)$$

where  $d$  is the distance of the perspective plane from the centre of the sphere. Each point  $\mathbf{x}_p$  corresponds to a pixel position  $\mathbf{u}_p$  on the reference image by  $\mathbf{u}_p = \mathbf{x}_p + (nc/2, nr/2)^T$ , where  $nc$  and  $nr$  are the number of columns and rows of pixels in the reference image respectively. This process is analogous to a wide-angle camera observing some planar region (reference image) in space. Rather than sample the value on the reference image at position  $\mathbf{u}_p$  using a linear interpolation, the mean value of all pixels on the reference image that project within the pixel  $\mathbf{u}$  on the wide-angle image is used. This technique is used to more closely simulate the acquisition of images using a digital camera.

For each reference image, wide-angle images are generated for five different distances  $d$ , and nine different rotations  $R$  at each distance (a total of 45 images are generated for each reference image). The distances  $d$  and rotation angles  $\alpha$  and  $\beta$  used at each distance are given in table 4.1.

The parabolic catadioptric and wide-angle images all have a resolution of  $1024 \times$



Figure 4.18: The data set consisting of 40 input images of size  $2272 \times 1704$  pixels.

Rotation $R$	distance $d$				
	550	1150	1750	2350	2950
$R_1(\alpha, \beta)$	$\alpha = 0$ $\beta = 0$	$\alpha = 0$ $\beta = 0$	$\alpha = 0$ $\beta = 0$	$\alpha = 0$ $\beta = 0$	$\alpha = 0$ $\beta = 0$
$R_2(-\alpha, \beta)$ $R_3(\alpha, \beta)$	$\alpha = 0.372$ $\beta = 0$	$\alpha = 0.726$ $\beta = 0$	$\alpha = 0.912$ $\beta = 0$	$\alpha = 1.023$ $\beta = 0$	$\alpha = 1.116$ $\beta = 0$
$R_4(\alpha, -\beta)$ $R_5(\alpha, \beta)$	$\alpha = 0$ $\beta = 0.488$	$\alpha = 0$ $\beta = 0.837$	$\alpha = 0$ $\beta = 1.023$	$\alpha = 0$ $\beta = 1.116$	$\alpha = 0$ $\beta = 1.209$
$R_6(-\alpha, -\beta)$ $R_7(-\alpha, \beta)$ $R_8(\alpha, -\beta)$ $R_9(-\alpha, \beta)$	$\alpha = 0.186$ $\beta = 0.244$	$\alpha = 0.363$ $\beta = 0.419$	$\alpha = 0.456$ $\beta = 0.512$	$\alpha = 0.512$ $\beta = 0.558$	$\alpha = 0.558$ $\beta = 0.605$

Table 4.1: Distances and rotation angles  $R = R_y(\beta)R_x(\alpha)$  used to generate the wide-angle images. The angles  $\alpha$  and  $\beta$  have units of radians.

768 pixels. The parabolic catadioptric and fisheye camera model functions that were used are shown in figure 4.7. The wide-angle images all have equal pixel scaling in the  $u$  and  $v$  coordinates, zero shear, and exhibit only radial distortion. Figures 4.19 and 4.20 show the set of synthetically generated parabolic catadioptric and fisheye images respectively for one of the reference images.

#### 4.3.6.2 Keypoints

For each reference image, SIFT and sSIFT keypoints are detected in each of the 45 parabolic catadioptric and 45 synthetic fisheye images using three different difference of Gaussian thresholds of 0.01, 0.02 and 0.03 (images have greyscale intensity values in the range 0-1). The SIFT and sSIFT keypoints are detected in the first  $n_{oct} = 5$  octaves of scale-space using  $n_{spo} = 3$  scales per octave. The same edge removal threshold  $r_{edge} = 10$  is used by both SIFT and sSIFT.

The SIFT keypoints are found by first doubling the size of the original wide-angle image and pre-smoothing to a starting scale of  $\sigma_0 = 1.6$  assuming the double sized image has an input scale of  $\sigma_{input} = 1$ . Although the original SIFT algorithm halves the image after each octave of scale-space for computational efficiency, in these experiments the image size is only halved after the first octave to ensure accurate interpolation of keypoint position and scale. The support region for a keypoint is defined as a circle on the wide-angle image centred about the keypoint position with radius equal to the characteristic scale  $\sigma$  of the keypoint. No account for the camera distortion is made during SIFT keypoint detection.

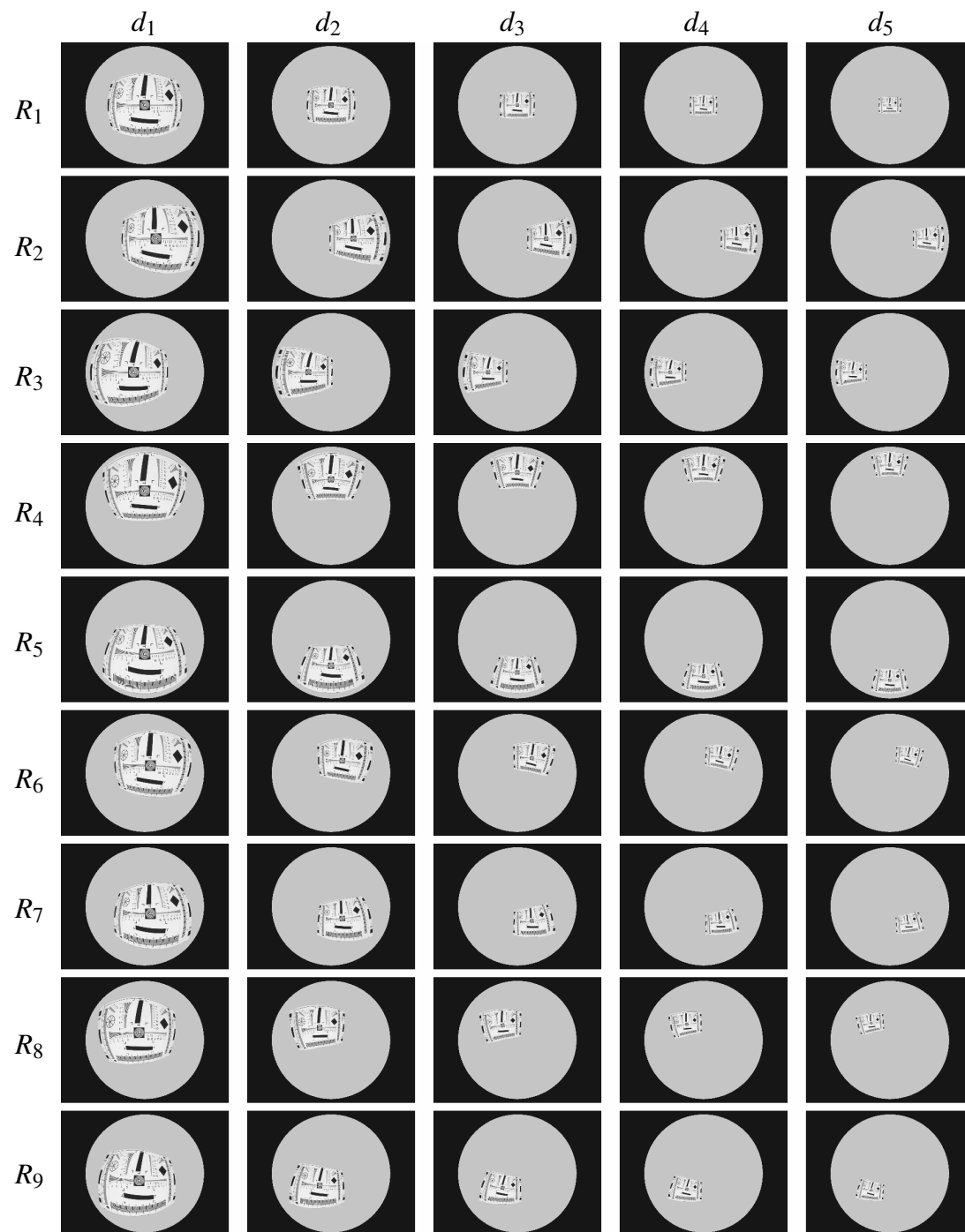


Figure 4.19: The set of synthetically generated wide-angle parabolic catadioptric images for one of the reference images. The parameters  $R$  and  $d$  refer to rotation and depth respectively (see table 4.1).

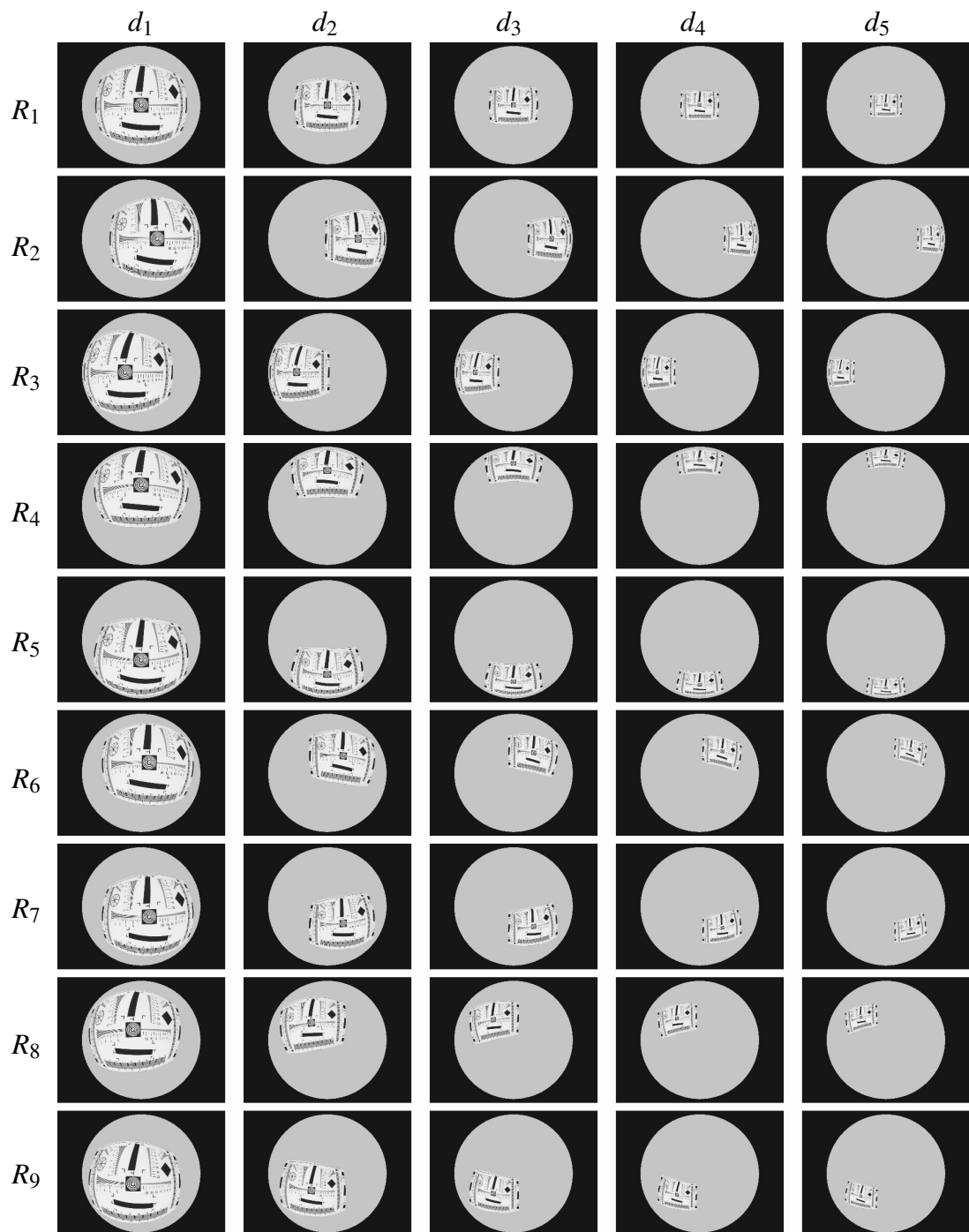


Figure 4.20: The set of synthetically generated wide-angle fisheye images for one of the reference images. The parameters  $R$  and  $d$  refer to rotation and depth respectively (see table 4.1).



The sSIFT keypoints are detected using the method described previously. The sample measurements  $\theta_s$  for the parabolic catadioptric and fisheye camera models are  $\theta_s = 0.0038$  and  $\theta_s = 0.0056$  radians respectively. These sample measurements are used to find the initial scale  $kt_0$  as described in section 4.3.2.1, and the set of remaining scales as described in section . The sSIFT keypoints were found using four different methods to obtain the image spectrum  $\hat{I}_{\mathbb{S}^2} = SFT(I_{\mathbb{S}^2})$ :

1. Sample rate  $b = 256$ , linear interpolation - sSIFT(256)
2. Sample rate  $b = 256$ , anti-aliasing interpolation filter  $B'_{b=256}$  - sSIFT(256\*)
3. Sample rate  $b = 512$ , linear interpolation - sSIFT(512)
4. Sample rate  $b = 512$ , anti-aliasing interpolation filter  $B'_{b=512}$  - sSIFT(512\*)

In all cases, the set of scale-space images  $\mathcal{L}_{\mathbb{S}^2}(\cdot; kt)$  were all the same size as the original wide-angle images ( $1024 \times 768$  pixels). The support region for each keypoint was defined as circle on the sphere parameterised by the angle  $\psi_s = \sqrt{2kt}$ , where  $kt$  is the characteristic scale of the keypoint.

### 4.3.6.3 Performance Metrics

The performance metrics used to compare SIFT and sSIFT are the outright number and the percentage correlation of keypoint correspondences found between wide-angle image pairs. If  $n_1$  and  $n_2$  are the number of keypoints detected in two different wide-angle images, the percentage correlation is

$$\% \text{correlation} = \frac{n_{1,2}}{\frac{1}{2}(n_1 + n_2)} \times 100\%, \quad (4.92)$$

where  $n_{1,2}$  is the number of correspondences. For all discussions regarding these experiments, the term correspondences refers to correct correspondences.

The correspondences are identified based on the position and support regions for each keypoint using the methodology of Mikolajczyk et al [161]. Let  $k'(\mathbf{u}', \mu')$  be the set of keypoints detected in a wide-angle image, where  $\mathbf{u}'$  and  $\mu'$  are the coordinates and support region contours of the keypoints respectively mapped to the reference image — this is possible as the exact transform between the wide-angle image and the reference image is known. Similarly, let  $k''(\mathbf{u}'', \mu'')$  be the set of keypoints detected in another wide-angle image, where  $\mathbf{u}''$  and  $\mu''$  are the coordinates and support region contours of the keypoints respectively mapped on the reference image. A corresponding

pair of keypoints  $k'_i(\mathbf{u}', \mu') \leftrightarrow k''_j(\mathbf{u}'', \mu'')$  is found if a number of criteria are satisfied. Firstly, the Euclidean distance  $d(\mathbf{u}'_i, \mathbf{u}''_j)$  must be less than 5 pixels (reference image size is  $2272 \times 1704$  pixels) and be the smallest of all Euclidean distances  $d(\mathbf{u}'_i, \mathbf{u}''_j)$  and  $d(\mathbf{u}', \mathbf{u}''_j)$ . Secondly, the error  $\varepsilon$  between the support region contours  $\mu'_i$  and  $\mu''_j$  (on the reference image) must be below some threshold. This error  $\varepsilon$  is [161]:

$$\varepsilon = 1 - \frac{n(\mu'_i \cap \mu''_j)}{n(\mu'_i \cup \mu''_j)} \quad (4.93)$$

where  $n(\mu'_i \cap \mu''_j)$  and  $n(\mu'_i \cup \mu''_j)$  are the number of pixels in the intersection and union of the regions enclosed by the support region contours respectively. A threshold of 0.2 is used in these experiments.

It is important to note here again that the support regions for SIFT keypoints are defined as circles on the wide-angle images, and the support regions for sSIFT keypoints are defined as circles on the sphere. The support region contours for the SIFT keypoints must be mapped from the wide-angle image to the sphere, rotated by  $R$ , and then mapped to the reference image to find the contour  $\mu$ . The support region contours for the sSIFT keypoints are simply rotated by  $R$ , and then mapped to the reference image to find the contour  $\mu$ .

The set of sSIFT(512\*) keypoint correspondences found in a pair of wide-angle fisheye images used in these experiments is shown in figure 4.21. The images in the left column are the fisheye images, and the images in the right column show the set of all keypoints  $k'(\mathbf{u}', \mu')$  and  $k''(\mathbf{u}'', \mu'')$  on the reference image. The corresponding keypoints are joined with lines.

#### 4.3.6.4 Results

For each reference image, the number of correspondences and the percentage correlation of correspondences were found for all image pairs subject to change in rotation ( $R$ ), and change in both scale (distance  $d$ ) and rotation ( $R$ ) for which there are 180 and 810 unique image pairs respectively. The results for all 40 reference image sets were combined and the mean and median values were found. In total, there is a total of  $40 \times 180 = 72000$  image pairs subject to change in rotation, and  $40 \times 810 = 32400$  image pairs subject to change in both scale and rotation. The mean and median results are presented in table 4.2 for the parabolic catadioptric camera, and table 4.3 for the fisheye camera. The median results are displayed as a bar graph for for change in rotation in 4.22, and for change in both scale and rotation in figure 4.23.

Image Transform	Keypoint Detector	DoG <sub>1</sub> = 0.01		DoG <sub>2</sub> = 0.02		DoG <sub>3</sub> = 0.03	
		% Correlation	# correspondences	% Correlation	# correspondences	% Correlation	# correspondences
Rotation	SIFT	62.78	(458.14)	66.84	(289.57)	70.56	(166.83)
	sSIFT (256)	60.61	(440.40)	65.41	(229.12)	70.00	(117.04)
	sSIFT (256*)	76.53	(410.00)	79.73	(209.32)	82.83	(102.78)
	sSIFT (512)	70.71	(393.10)	74.72	(240.70)	78.60	(133.90)
	sSIFT (512*)	71.49	(403.68)	75.00	(249.91)	79.00	(138.72)
Rotation & Scale	SIFT	30.63	(181.04)	34.11	(122.35)	37.36	(73.08)
	sSIFT (256)	10.40	(61.62)	14.89	(44.50)	19.59	(28.32)
	sSIFT (256*)	14.02	(65.74)	18.80	(43.96)	23.41	(26.11)
	sSIFT (512)	25.84	(123.17)	29.49	(83.27)	33.50	(49.54)
	sSIFT (512*)	26.31	(126.63)	29.88	(86.60)	34.26	(51.49)

Table 4.2: Median percentage correlation and number of correspondences for the parabolic catadioptric camera using SIFT and sSIFT (mean values shown in brackets). DoG is the difference of Gaussian threshold.



Image Transform	Keypoint Detector	DoG <sub>1</sub> = 0.01		DoG <sub>2</sub> = 0.02		DoG <sub>3</sub> = 0.03							
		% Correlation	# correspondences	% Correlation	# correspondences	% Correlation	# correspondences						
Rotation	SIFT	33.65	(34.73)	211.00	(400.39)	35.15	(35.87)	129.00	(241.26)	36.48	(37.05)	72.00	(134.67)
	sSIFT (256)	56.43	(52.76)	274.00	(470.42)	60.43	(56.50)	145.00	(238.67)	65.03	(61.03)	76.00	(121.61)
	sSIFT (256*)	74.28	(72.26)	267.00	(417.45)	77.10	(75.40)	139.00	(209.73)	80.12	(78.43)	68.00	(101.43)
	sSIFT (512)	67.99	(66.67)	480.00	(768.99)	72.91	(71.60)	262.00	(404.80)	77.60	(76.34)	133.00	(201.62)
	sSIFT (512*)	73.17	(72.29)	542.00	(864.16)	76.56	(75.87)	287.00	(439.53)	80.08	(79.31)	139.00	(211.40)
Rotation & Scale	SIFT	11.22	(14.78)	78.00	(116.36)	11.85	(16.10)	51.00	(76.49)	12.48	(17.32)	29.00	(44.70)
	sSIFT (256)	8.32	(9.25)	41.00	(56.63)	12.28	(13.61)	31.00	(41.28)	16.44	(18.03)	21.00	(26.68)
	sSIFT (256*)	13.05	(15.15)	48.00	(63.62)	17.69	(19.99)	34.00	(42.29)	21.95	(24.32)	20.00	(24.87)
	sSIFT (512)	20.67	(22.70)	152.00	(196.44)	25.29	(27.39)	96.00	(120.13)	29.83	(31.91)	52.00	(66.29)
	sSIFT (512*)	19.80	(22.50)	157.00	(200.88)	24.37	(27.19)	98.00	(121.78)	29.25	(31.71)	53.00	(66.26)

Table 4.3: Median percentage correlation and number of correspondences for the fish-eye camera using SIFT and sSIFT (mean values shown in brackets). DoG is the difference of Gaussian threshold.

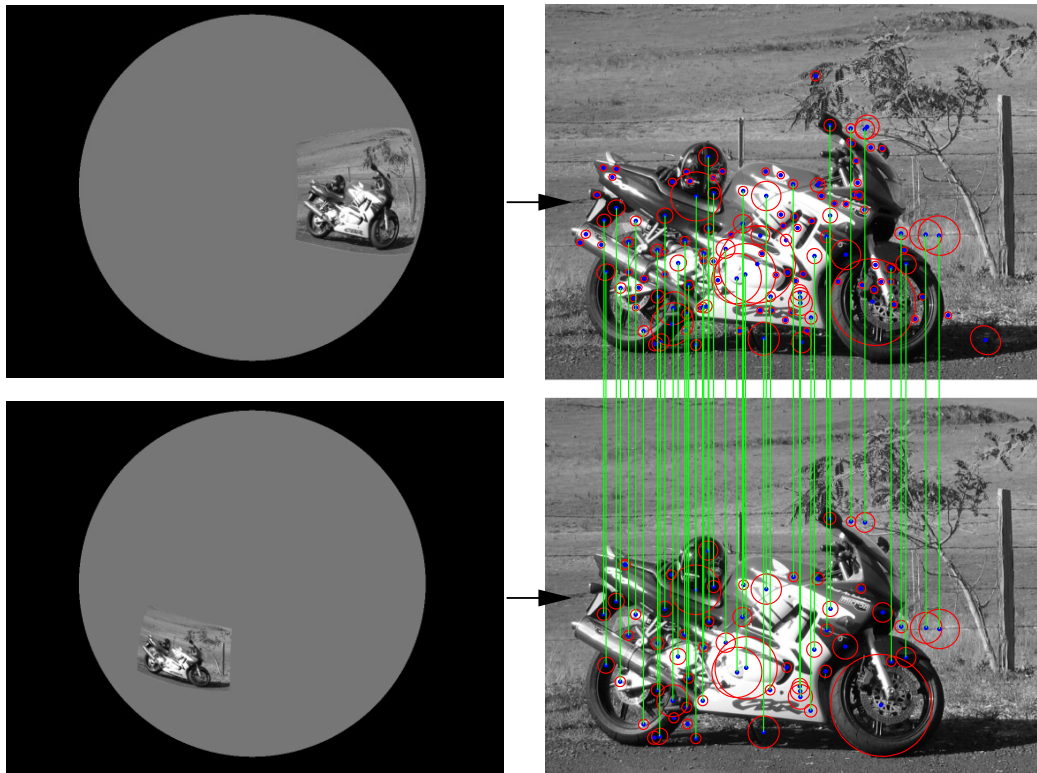
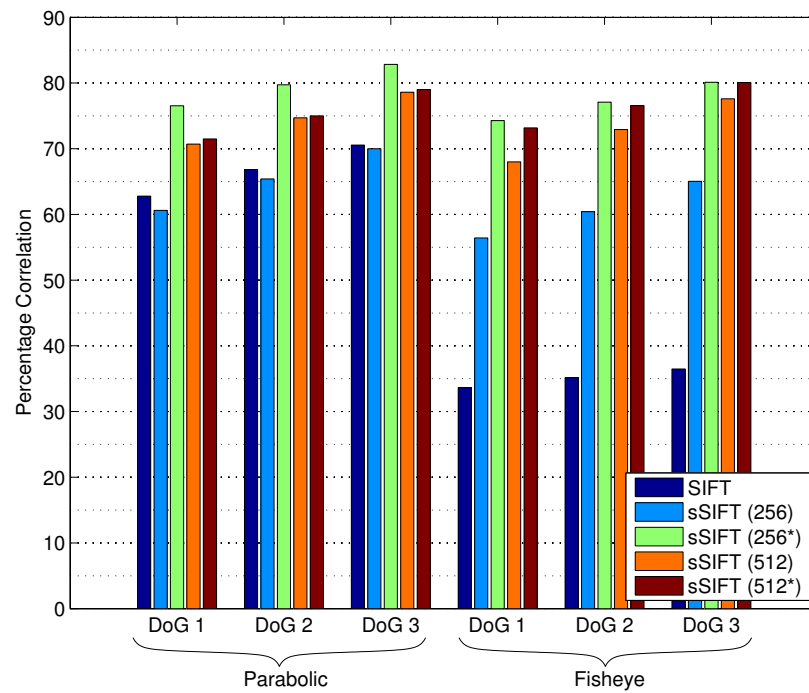


Figure 4.21: Keypoint correspondences found between two synthetic wide-angle fish-eye images. The images on the left are the synthetic images. The keypoint positions and support regions are shown mapped to the reference image in the right column. The lines indicate the correspondences.

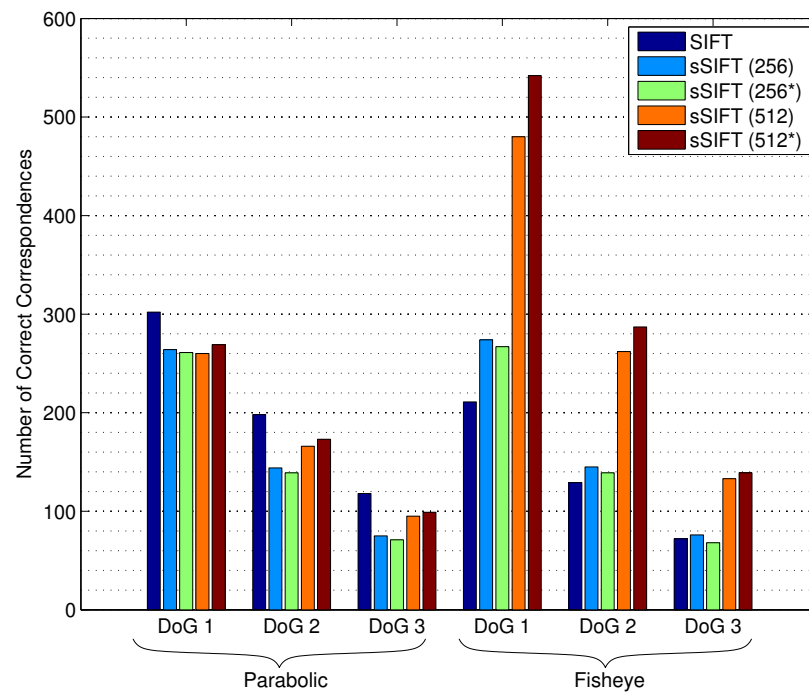
#### 4.3.6.5 Discussion

Several aspects of the results are discussed in this section, and include: the effect of the DoG threshold on the performance of SIFT and sSIFT, the effect of the anti-aliasing filter on the performance of sSIFT, the effect of sample rate selection on the performance of sSIFT, and the relative performance of SIFT and sSIFT.

**DoG Selection:** The results in figures 4.22 and 4.23 indicate that for each keypoint detector, camera and image transformation (change in rotation, and change in both scale and rotation), increasing the DoG threshold increases the percentage correlation and decreases the number of correspondences. The decrease in the number of correspondences was expected as only a limited number of keypoints can be detected using a large DoG threshold. The increase in the percentage correlation can be attributed to the fact that the ability to reliably detect and interpolate keypoint position and scale is affected by image noise — image noise will produce noise in the difference of Gaussian images. As the magnitude of the difference of Gaussian function increases, the

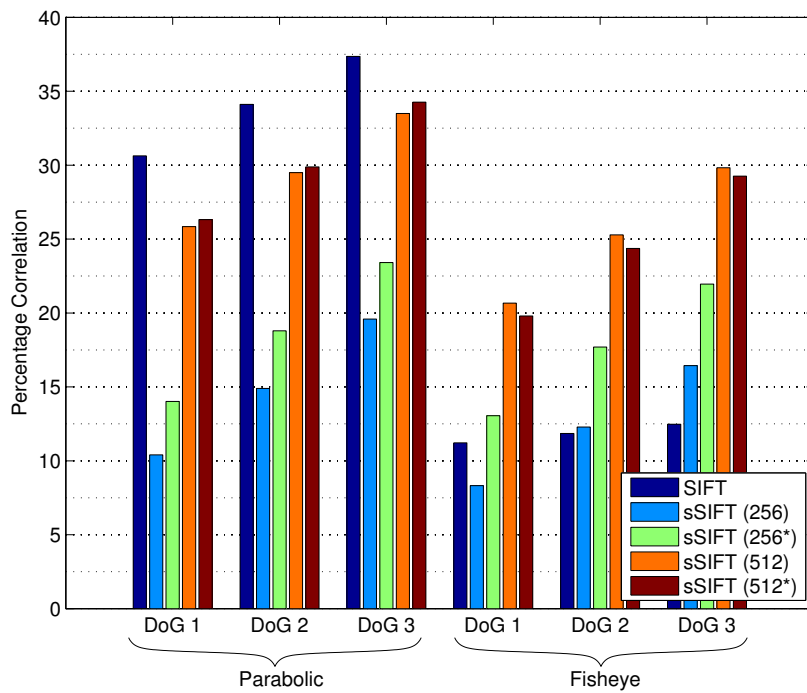


(a) Percentage Correlation.

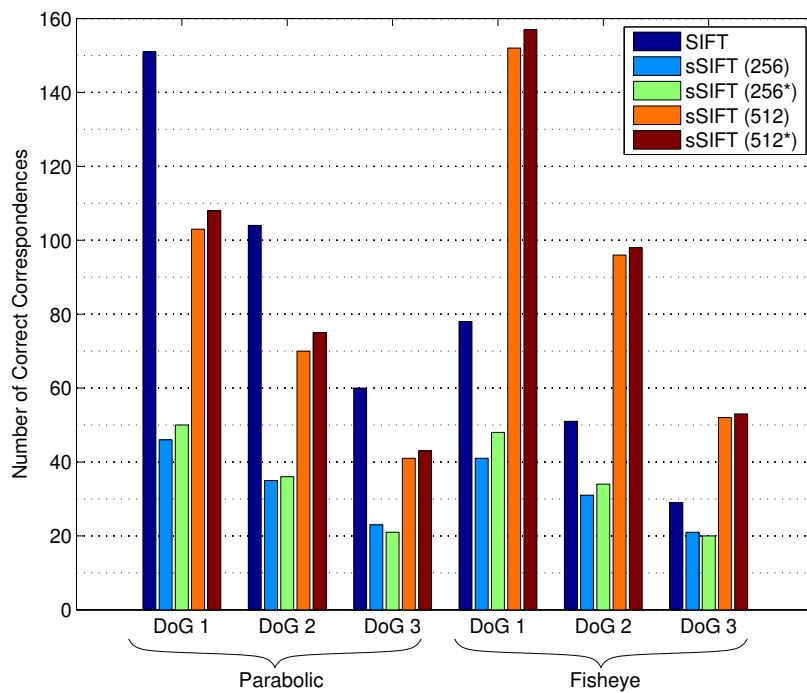


(b) Number of Correspondences.

Figure 4.22: Median percentage correlation and number of correspondences for images subject to change in rotation for SIFT and sSIFT. The difference of Gaussian thresholds are  $DoG_1 = 0.01$ ,  $DoG_2 = 0.02$  and  $DoG_3 = 0.03$ .

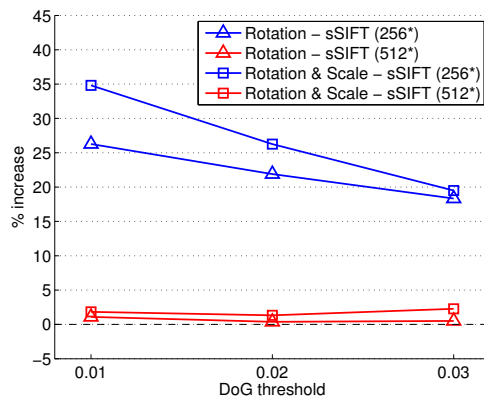


(a) Percentage Correlation.

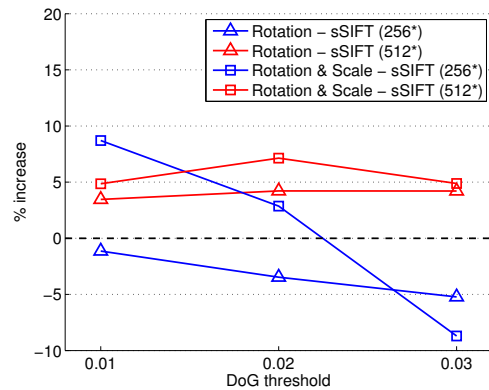


(b) Number of Correspondences.

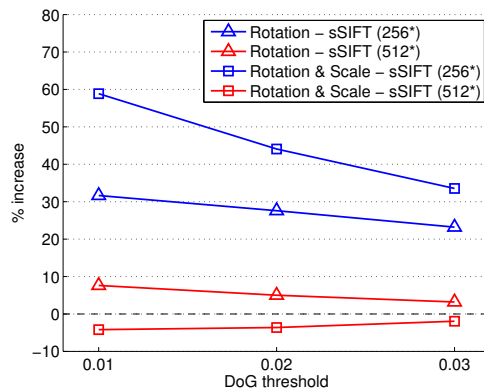
Figure 4.23: Median percentage correlation and number of correspondences for images subject to change in both rotation and scale for SIFT and sSIFT. The difference of Gaussian thresholds are  $DoG_1 = 0.01$ ,  $DoG_2 = 0.02$  and  $DoG_3 = 0.03$ .



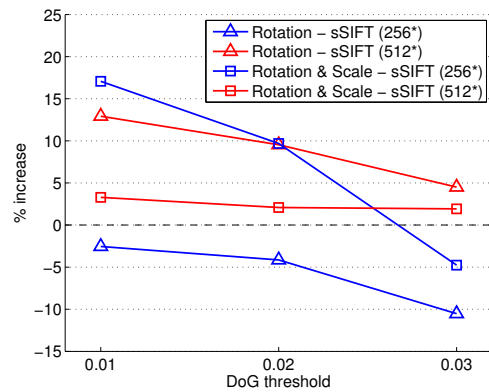
(a) Increase in percentage correlation for the parabolic catadioptric camera.



(b) Increase in number of correspondences for the parabolic catadioptric camera.



(c) Increase in percentage correlation for the fisheye camera.



(d) Increase in number of correspondences for the fisheye camera.

Figure 4.24: Percentage increase in the percentage correlation of keypoints and the number of correspondences using the anti-aliasing filter for each sample rate  $b = 256$  and  $b = 512$ .

signal to noise ratio increases, hence the ability to reliably detect and interpolate keypoints with a large absolute DoG value is less sensitive to image noise than keypoints with a small absolute DoG value.

**Anti-aliasing:** As the maximum bandwidth  $b_l$  of both cameras exceeds the sample rates  $b = 256$  and  $b = 512$  used to find the sSIFT keypoints, it was anticipated that the anti-aliasing interpolation filter would improve the performance of sSIFT. Figure 4.24 shows the percentage increase in performance of sSIFT(256\*) over sSIFT(256), and sSIFT(512\*) over sSIFT(512).

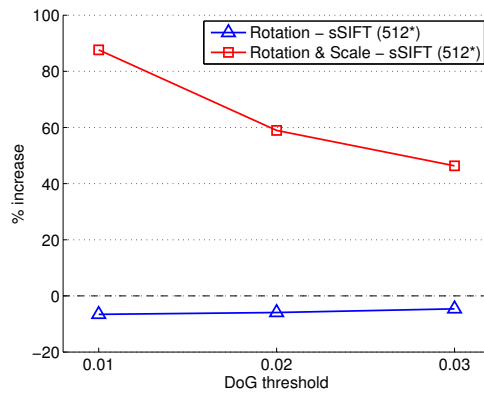
For the sample rate  $b = 256$ , the results in figures 4.24a and 4.24c indicate significant improvements in the percentage correlation of keypoints for the parabolic cata-

dioptric and fisheye camera respectively for both image transformations and each DoG threshold. These improvements reduce as the DoG threshold is increased. This occurs as the sensitivity of keypoint detection to image noise (i.e. aliasing artifacts) increases as the DoG threshold is reduced, as discussed previously. Referring again to the results for sample rate  $b = 256$ , figures 4.24b and 4.24d show a small decrease in the number of correspondences for each camera and DoG threshold for change in rotation. Except for the highest DoG threshold ( $DoG = 0.03$ ), there is a small improvement in the number of correspondences for each camera subject to change in both rotation and scale. Overall, although there are some small reductions in the number of correspondences, the large improvements in the percentage correlation observed from the result suggest that for these experiments, the anti-aliasing filter improves the performance of sSIFT keypoint detection for a sample rate  $b = 256$  (i.e. sSIFT(256\*) gives improved results over sSIFT(256)). This is particularly true for the smallest difference of Gaussian threshold  $DoG = 0.01$ .

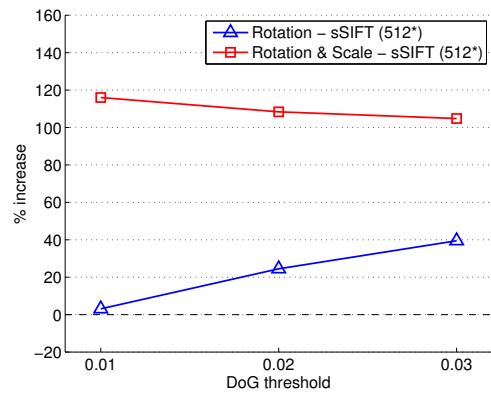
The results in figures 4.24a and 4.24c indicate, for each camera, much smaller overall improvement in the percentage correlation using the anti-aliasing filter for sample rate  $b = 512$ . The only reduction in performance is for a change in both scale and rotation for the fisheye camera. This is due to the fact that the sample rate  $b = 512$  is much closer to the maximum camera bandwidths than the sample rate  $b = 256$ . However, in contrast to the results for the sample rate  $b = 256$ , figures 4.24b and 4.24d indicate an increase in the number of correspondences for both image transformations and each DoG threshold for the parabolic catadioptric camera and fisheye camera respectively. Although there was a very small decrease in the percentage correlation for the fisheye camera subject to change in both scale and rotation, the increase in the number of correct correspondences suggests that, for these experiments, the use of the anti-aliasing filter is still beneficial for the sample rate  $b = 512$  (i.e. sSIFT(512\*) gives improved results over sSIFT(512)).

**Sample Rate Selection:** As it was concluded previously that the anti-aliasing filter in general improves the performance of sSIFT, the effect of sample rate selection on the performance of sSIFT is made between sSIFT(256\*) and sSIFT(512\*). The percentage improvement in performance of sSIFT(512\*) over sSIFT(256\*) is shown in figure 4.25.

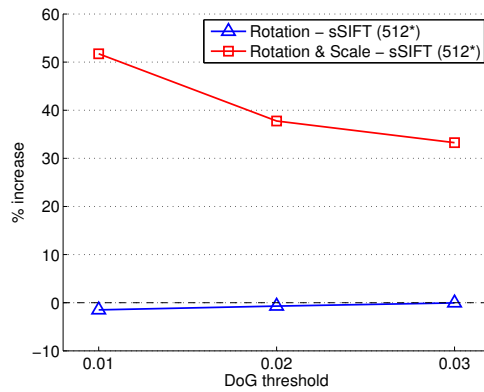
A large increase in the percentage correlation for change in both rotation and scale is observed for each DoG threshold in figures 4.25a and 4.25c for the parabolic catadioptric and fisheye camera respectively. In contrast, a small reduction in the percent-



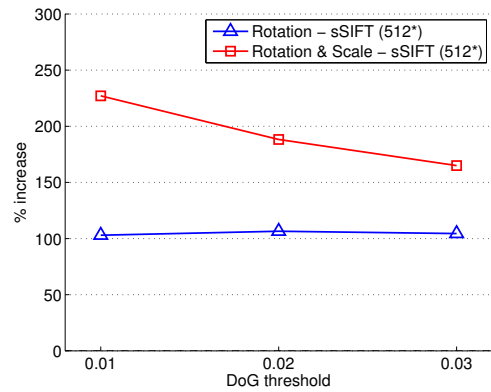
(a) Increase in percentage correlation for the parabolic catadioptric camera.



(b) Increase in number of correspondences for the parabolic catadioptric camera.



(c) Increase in percentage correlation for the fisheye camera.



(d) Increase in number of correspondences for the fisheye camera.

Figure 4.25: Percentage increase in the percentage correlation of keypoints and the overall number of keypoint correspondences of sSIFT(512\*) over sSIFT( $b = 256^*$ ).

age correlation is observed for change in rotation for each DoG threshold for both cameras. Figures 4.25b and 4.25d show an increase in the number of correspondences for each image transformation and DoG threshold for both cameras. Except for a change in rotation with the parabolic camera, this increase in the overall number of correspondences is significantly large, with at least twice as many correspondences found. These results give strong evidence that, in these experiments, the performance of sSIFT improves as the sample rate increases.

It is of interest to discuss here a potential explanation for the large increase in the percentage correlation for change in both scale and rotation, and only a small negative increase for a change in rotation. Decreasing the sample rate  $b$  is in effect a filtering operation, and suppresses the fine scale structures in the image (i.e. the high frequency components). This limits the ability to detect keypoints at small scales  $kt$ , and is the reason why there is a significant increases in the number of correct correspondences

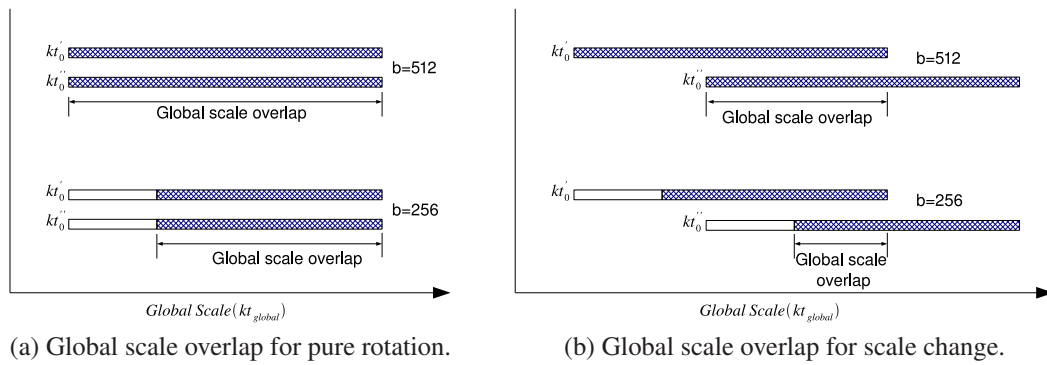


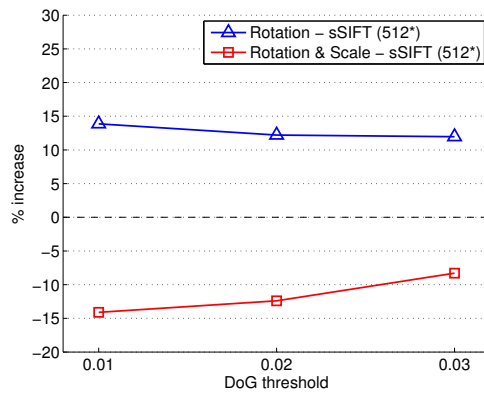
Figure 4.26: Variations in the global scale overlap for different sample rates and image transformations. The hatched region shows the range of ‘useful’ scales in which keypoint can be detected — as the sample rate is reduced, the number of keypoints detected at the smallest scales is limited.

using the higher sample rate  $b = 512$  — in general, the majority of keypoints detected using sSIFT(512\*) are found at the smallest scales  $kt$ . Consider then the set of keypoints detected in two different wide-angle images separated by a large scale change. For a large scale change, there is a limited *global* overlap in the set of scales  $kt$  used in each image. A keypoint in the first image for example, detected in the difference of Gaussian image  $\mathcal{D}_{\mathbb{S}^2}(\cdot; kt_8)$ , may correspond to the same keypoint detected in the difference of Gaussian image  $\mathcal{D}_{\mathbb{S}^2}(\cdot; kt_1)$  in the other image. By reducing the sample rate, the range of ‘useful’ scales in which keypoints can be detected is reduced. This reduces the *global* scale overall between the images and the percentage correlation of keypoints that can be found between the images, as illustrated in figure 4.26b. For images separated only by a change in rotation, there is a perfect global overlap in the range of scales  $kt$  between the two images. Assuming that the range of ‘useful’ scales in the two images remains the same as the sample rate decreases, there will still be a perfect global overlap of these ‘useful’ scales, as illustrated in figure 4.26a. This is why there are only small variations in the percentage correlation results between sSIFT(256\*) and sSIFT(512\*) for change in rotation.

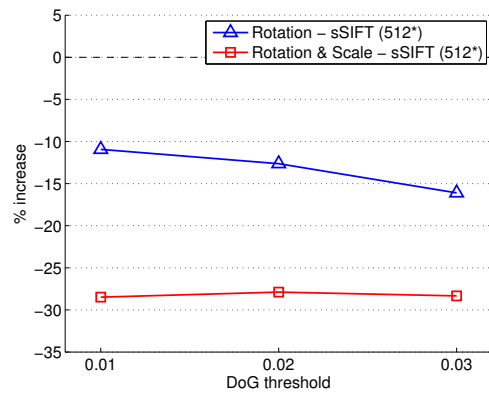
**SIFT vs sSIFT:** To date, the results suggest that overall sSIFT(512\*) gives the best performance compared to the other sSIFT keypoints. Therefore, the relative performance of SIFT and sSIFT(512\*) is compared here. The percentage improvement in performance of sSIFT(512\*) over SIFT is shown in figure 4.27.

The results in figures 4.27c and 4.27d show, for the fisheye camera, a significant increase in the percentage correlation and number of correspondences respectively for both image transformations and all DoG thresholds. It is concluded from these results

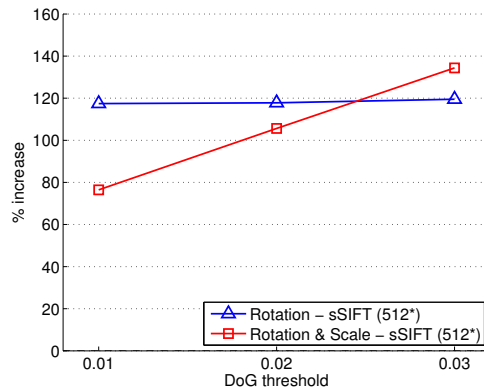




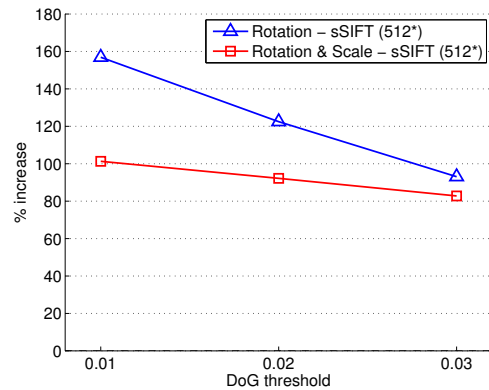
(a) Increase in percentage correlation for parabolic catadioptric camera.



(b) Increase in number of correspondences for the parabolic catadioptric camera.



(c) Increase in percentage correlation for the fisheye camera.



(d) Increase in number of correspondences for the fisheye camera.

Figure 4.27: Percentage increase in the percentage correlation of keypoints and the overall number of keypoint correspondences for spherical SIFT with bandwidth  $b = 512^*$  relative to SIFT.

that for the experiments conducted, sSIFT( $512^*$ ) is more suited for keypoint detection in the fisheye images than SIFT.

For the parabolic catadioptric camera, the results in figure 4.27a show an increase in the percentage correlation for change in rotation, and a decrease in percentage correlation for change in both rotation and scale change for all DoG thresholds. Figure 4.27b also indicates a decrease in the number of correspondences for change in rotation, and change in both rotation and scale for all DoG thresholds. These results suggest that in these experiments, SIFT was more suited for keypoint detection in the parabolic catadioptric images than sSIFT( $512^*$ ).

The explanation for the overall decrease in performance of sSIFT ( $512^*$ ) compared to SIFT is quite detailed. Recall from chapter 2 that image formation with a parabolic catadioptric camera can be modelled as the perspective projection of a scene point to

the sphere, followed by a stereographic projection of the point to the image plane. As stereographic projection is a conformal mapping which locally preserves angles, an isotropic kernel at any position on the image, such as the Gaussian, maps via inverse stereographic projection to an approximately isotropic function on the sphere. As the Gaussian and the stereographic projection of the spherical Gaussian to the the function  $\mathcal{G}_{\mathbb{S}^2}(\cdot; kt)$  are similar in shape (see figure 4.13, pg. 179), convolution of a parabolic catadioptric image with the Gaussian is approximately equal to the convolution of the image, mapped to the sphere, with the spherical Gaussian  $G_{\mathbb{S}^2}(\cdot; kt)$  with variable scale  $kt$  — the scale  $kt$  of  $G_{\mathbb{S}^2}(\cdot; kt)$  changes depending on its position on the sphere. Furthermore, as SIFT defines a keypoint support region as a circle on the image, for the parabolic catadioptric camera it maps via inverse stereographic projection to an approximately circular region on the sphere. In a nutshell, applying SIFT to a parabolic catadioptric image is approximately invariant to the camera distortion and not limited by the same sample rate issues as sSIFT. This approximation forms the basis of the pSIFT keypoint detector developed in section 4.4, where all the concepts discussed here will be described in greater detail.

To recap, the steps used to detect SIFT keypoints in a parabolic catadioptric image are approximately invariant to the camera distortion. As SIFT keypoints are found using the original image values without the need to obtain the spectrum  $\hat{I}_{\mathbb{S}^2}$  of the image, which as discussed can introduce aliasing artifacts, in some respects it is suited for keypoint detection in parabolic catadioptric images. The decrease in the number of correct correspondences using sSIFT(512\*) compared to SIFT is also due to the fact that sSIFT maps each scale-space image  $L_{\mathbb{S}^2}(\cdot; kt)$  back to the the set of scale-space images  $\mathcal{L}_{\mathbb{S}^2}(\cdot; kt)$  on the original image plane. This requires a linear interpolation of the values  $L_{\mathbb{S}^2}(\cdot; kt)$  which introduces an additional smoothing operating. This can reduce the overall number of keypoints detected in an image, and consequently reduce the number of correspondences between images.

### 4.3.7 Conclusions

The spherical SIFT (sSIFT) keypoint detector was developed in this section, which is a variant of the SIFT keypoint detector designed for wide-angle image. Scale-space images are obtained by convolving the image, mapped to the sphere, with the spherical Gaussian. This convolution is implemented in the spherical Fourier domain and requires finding the spectrum of the wide-angle image using a discrete spherical Fourier transform with sample rate  $b$ . A method to estimate the bandwidth of a wide-angle image was formulated which defines the minimum sample rate  $b$  required to prevent

aliasing in the image spectrum. In the case where this required sample rate exceeds the maximum computationally feasible value, a practical approach to minimise aliasing was presented in the form of an anti-aliasing interpolation filter used to sample the image values for the discrete spherical Fourier transform. A methodology for selecting a suitable set of scales  $kt$  was then presented based on the scales used by SIFT and the camera intrinsic parameters. The practical procedure used to find the set of scale-space images  $L_{\mathbb{S}^2}(\cdot; kt)$  was then outlined. Each scale-space image  $L_{\mathbb{S}^2}(\cdot; kt)$ , which is a function on the sphere, is mapped back to the scale-space image  $\mathcal{L}_{\mathbb{S}^2}(\cdot; kt)$  defined on the original image plane. Candidate keypoints are then detected in the difference of Gaussian (scale-space) images  $\mathcal{D}_{\mathbb{S}^2}(\cdot; kt)$ . After removing edge responses, the accuracy of the position and scale of the remaining keypoints are improved using the same interpolation scheme as SIFT. The support region for a keypoint is then defined as a circle on the sphere, centred at the position of the keypoint on the sphere, whose size is set relative to the keypoint scale  $kt$ .

Experiments were conducted to compare the percentage correlation and number of correspondences of keypoints detected in synthetic wide-angle images using SIFT and sSIFT. Overall, the results gave evidence that the performance of sSIFT improves as the sample rate  $b$  increase, and that the anti-aliasing filter improves performance when the required sample rate exceeds the maximum computationally feasible value. It was concluded that overall, the best performance of sSIFT in the experiments was obtained for a sample rate  $b = 512$  with the anti-aliasing interpolation filter (i.e. sSIFT(512\*)). For the fisheye camera, the results indicated that sSIFT(512\*) gave significantly better results than SIFT. However, for the parabolic catadioptric camera, SIFT outperformed sSIFT(512\*). This result for the parabolic catadioptric camera was attributed to the fact that applying SIFT to stereographic images (the images obtained by a parabolic catadioptric camera) is in some respects invariant to camera distortion. In particular, the convolution of a stereographic image with a Gaussian is approximately equal to the convolution of the image on the sphere with the spherical Gaussian with changing (non-uniform) scale. SIFT also has the advantage that the original image intensity values are used. sSIFT in contrast requires sampling the intensity values to obtain the spectrum  $\hat{I}_{\mathbb{S}^2}$  of the image. The sample rate that needs to be used by sSIFT to prevent aliasing when finding the spectrum of an image using a discrete spherical Fourier transform can exceed the capabilities of the hardware used. This was the case when finding sSIFT keypoints in the fisheye and parabolic catadioptric images in the experiments in this section.

## 4.4 Scale-Invariant Keypoint Detection: parabolic SIFT (pSIFT)

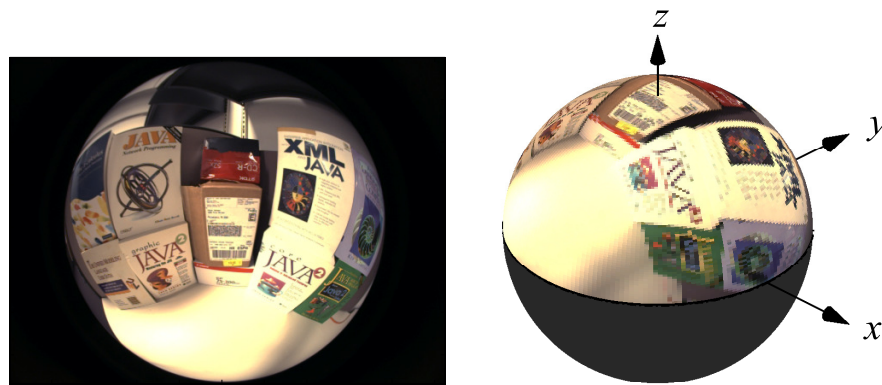
The second variant of SIFT developed in this chapter is parabolic SIFT (pSIFT). As was the case for sSIFT, pSIFT defines scale-space for wide-angle images as the convolution of  $I_{\mathbb{S}^2}$  with the spherical Gaussian  $G_{\mathbb{S}^2}(\cdot; kt)$ . An approximation to this convolution is used that is implemented efficiently in the spatial domain with stereographic images  $I_p$ . It was observed in the previous experiments that SIFT performed relatively well for the parabolic catadioptric camera. As was recently discussed in section 4.3.6.5, the reason for this is probably due to the fact that image formation for a parabolic catadioptric camera is described by the stereographic projection of an image, on the sphere, to the stereographic image plane. This projection is a conformal mapping which locally preserves shapes, and as a result an isotropic Gaussian on the stereographic image back projects to an approximately isotropic function on the sphere. Therefore, if the spherical Gaussian on the sphere is projected to the image plane, using stereographic projection, the resulting function in the image plane will be approximately isotropic. This is the fundamental property that pSIFT uses to derive an approximate spherical diffusion process which can be implemented efficiently in the spatial domain. Performing the convolution in the spatial domain removes the need to obtain the spectrum of the image and the sample rate problems encountered by sSIFT.

### 4.4.1 Conversion to a Stereographic Image

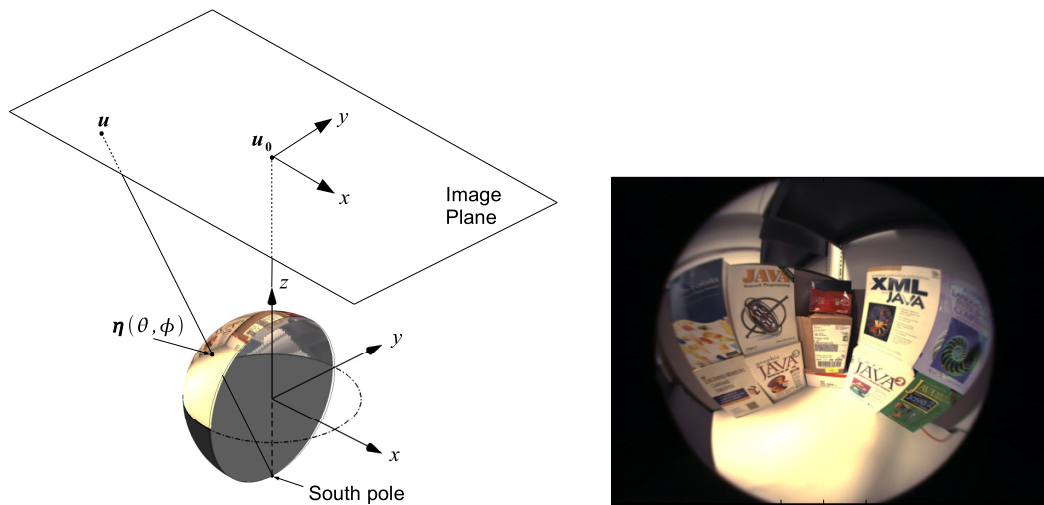
A *stereographic image* is defined to be the image that would be obtained if image formation were described by a perspective projection of scene points to the sphere, followed by a stereographic projection from the south pole to the image plane. This is the same image that would be obtained by a parabolic catadioptric camera, hence the name parabolic SIFT. As the image  $I$  obtained with any central projection wide-angle camera can be mapped to the image  $I_{\mathbb{S}^2}$  on the sphere, it can be converted to a stereographic image  $I_p$ , as illustrated in figure 4.28. The projection of a point  $\eta$  on the sphere to a pixel position  $\mathbf{u}$  on the stereographic image is

$$\mathbf{u} = \left( \frac{1 + m_p}{1 + \eta_z} \right) \begin{bmatrix} \eta_x \\ \eta_y \end{bmatrix} + \mathbf{u}_0, \quad (4.94)$$

where  $m_p$  is the distance of the stereographic image plane from the centre of the sphere. The polar coordinates of a point  $\mathbf{x}(r, \zeta) = \mathbf{u} - \mathbf{u}_0$  on the stereographic image can also



(a) A fisheye image (left) mapped to the image  $I_{S^2}$  on the unit sphere (right).



(b) Given any image  $I_{S^2}$  on the sphere, the stereographic image  $I_p$  (right) is produced as the stereographic projection of  $I_{S^2}$  from the south pole to the stereographic image plane.

Figure 4.28: An image (a) obtained with a calibrated central projection fisheye camera can be mapped to the image  $I_{S^2}$  on the unit sphere. (b) shows the process for obtaining the stereographic image  $I_p$ . The image  $I_{S^2}$  is mapped by stereographic projection from the south pole to the stereographic image plane.

be obtained from the spherical polar coordinates of the point  $\eta(\theta, \phi)$  as

$$r = (m_p + 1) \tan\left(\frac{\theta}{2}\right) \quad \zeta = \phi, \quad (4.95)$$

where the inverse is

$$\theta = 2 \tan^{-1}\left(\frac{r}{m+1}\right), \quad \phi = \zeta. \quad (4.96)$$

In later experiments, all wide-angle images are converted to stereographic image using the same approach. The size of the stereographic image is made equal to the size of the original wide-angle image. A  $1024 \times 768$  pixel fisheye image for example is converted to a  $1024 \times 768$  pixel stereographic image. The position of the principal

point  $\mathbf{u}_0$  is also set at the exact same position as in the original wide-angle camera. Finally, if a point on the equator of the sphere projects to a point at a radius  $r_{\pi/2}$  from the principal point in the original wide-angle image, the distance  $m_p$  is set so that a point on the equator of the sphere projects to a point on the stereographic image at the same radius  $r_{\pi/2}$  from the principal point:

$$m_p = r_{\pi/2} - 1. \quad (4.97)$$

A simple linear interpolation of the original wide-angle image values is used when converting it to a stereographic image.

As was the case when using sSIFT, pSIFT requires resampling (i.e. interpolating) the original wide-angle images values. For sSIFT, this sampling is needed when finding the SFT of an image (i.e. mapping the image to an equiangular  $\theta, \phi$  grid). For pSIFT, this resampling is needed when converting the original wide-angle image to a stereographic image. However, it is evident from figures 4.14 (pg. 181) and 4.28 that the degree by which the appearance of the original wide-angle changes is less for pSIFT than sSIFT, which means that the magnitude of interpolation artifacts will be less. Referring to figure 3.23 (pg. 147), it can also be seen that the degree by which the appearance of the original wide-angle changes is less for pSIFT than if the wide-angle image were converted to a wide-angle of view perspective image.

#### 4.4.2 Approximate Spherical Diffusion using Stereographic Projection

Recall from previous discussions that the spherical Gaussian  $G_{\mathbb{S}^2}(\cdot; kt)$  is by definition centred at the north pole  $\mathbf{n}$  and projects for the given camera model to the kernel  $\mathcal{G}_{\mathbb{S}^2}(\cdot; kt)$  on the image, centred at the principal point  $\mathbf{u}_0$ . The spherical Gaussian rotated by some rotation matrix  $R$  to be centred at the point  $\eta' = R\mathbf{n}$  was defined as  $G_{\mathbb{S}^2(\eta')}(\cdot; kt)$ .  $G_{\mathbb{S}^2(\eta')}(\cdot; kt)$  projects for the given camera model to the kernel  $\mathcal{G}_{\mathbb{S}^2(\eta')}(\cdot; kt)$  on the image, centred at the pixel coordinate  $\mathbf{u}' \mapsto \eta'$ . Figure 4.29 illustrates the appearance of the kernels  $\mathcal{G}_{\mathbb{S}^2}(\cdot; kt)$  and  $\mathcal{G}_{\mathbb{S}^2(\eta')}(\cdot; kt)$  as they appear on a stereographic image.  $G_{\mathbb{S}^2(\eta')}(\cdot; kt)$  is centred at the same point  $\eta'$  used in the previous example in figure 4.3.

pSIFT approximates spherical diffusion as a convolution operation on the stereographic image plane. This approximation is based on two key assumptions which are not mutually exclusive:

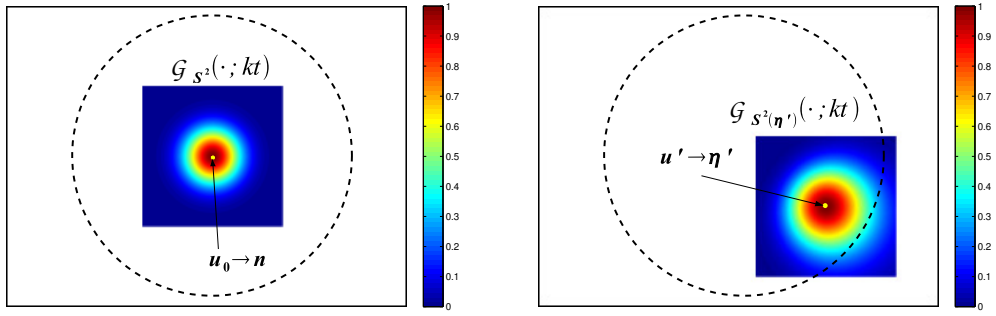


Figure 4.29: Appearance of the spherical Gaussian kernels  $\mathcal{G}_{\mathbb{S}^2}(\cdot; kt)$  and  $\mathcal{G}_{\mathbb{S}^2(\eta')}(\cdot; kt)$  as they would appear on a stereographic image. The dashed circle represents the field of view of the camera.

1. As stereographic is a conformal mapping which locally preserves angles, the spherical Gaussian  $\mathcal{G}_{\mathbb{S}^2(\eta')}(\cdot; kt)$  is assumed to be isotropic for all  $\eta'$  and scales  $kt$ .
2. If the kernel  $\mathcal{G}_{\mathbb{S}^2}(\cdot; kt)$  were shifted so that it was centred at the point  $\mathbf{u}' \mapsto \eta'$ , then there exists some scale  $kt(r)$ , which is a function of the scale  $kt$  and radius  $r$  of the point  $\mathbf{u}'$  from the principal point, for which  $\mathcal{G}_{\mathbb{S}^2}(\cdot; kt)$  is equal to  $\mathcal{G}_{\mathbb{S}^2(\eta')}(\cdot; kt(r))$ .

Using both these assumptions, which will be discussed in detail in this section, pSIFT first finds an isotropic version of the kernel  $\mathcal{G}_{\mathbb{S}^2}(\cdot; kt)$  ( $\mathcal{G}_{\mathbb{S}^2}(\cdot; kt)$  would only be circular symmetrical if the principal point were defined at an integer pixel position). This  $w \times w$  kernel is denoted  $\tilde{\mathcal{G}}_{\mathbb{S}^2}(\cdot; kt)$  and has values

$$\tilde{\mathcal{G}}_{\mathbb{S}^2}(x, y; kt) = \sum_{l \in \mathbb{N}} \sqrt{\frac{2l+1}{4\pi}} Y_l^0(\theta_{x,y}) e^{-l(l+1)kt}, \quad x, y \in \{-w, -w+1, \dots, w-1, w\}, \quad (4.98)$$

where

$$\theta_{x,y} = 2 \arctan \left( \frac{\sqrt{x^2 + y^2}}{m_p + 1} \right). \quad (4.99)$$

Assuming that  $\mathcal{L}_{\mathbb{S}^2}(\cdot; 0) = f_p$  is the initial condition, where  $f_p$  is a function on the stereographic image plane, pSIFT obtains for a given scale  $kt$  the scale-space image  $\mathcal{L}_{\mathbb{S}^2}(\cdot; kt(r))$  by convolving  $f_p$  with  $\tilde{\mathcal{G}}_{\mathbb{S}^2}(\cdot; kt)$ :

$$\mathcal{L}_{\mathbb{S}^2}(\cdot; kt(r)) = f_p * \tilde{\mathcal{G}}_{\mathbb{S}^2}(\cdot; kt), \quad (4.100)$$

where  $\mathcal{L}_{\mathbb{S}^2}(\cdot; kt(r))$  is a function on the stereographic image plane. As will be shown,  $kt(r)$  is some function of the scale  $kt$  and the radius  $r$  of the pixel from the principal



point. This means that the scale-space image  $\mathcal{L}_{\mathbb{S}^2}(\cdot; kt(r))$  has a non-uniform scale  $kt$ . It is important to note that the convolution in 4.100 differs from the definition in 4.28. The convolution in 4.100 is implemented on the stereographic image plane, where  $\mathcal{L}_{\mathbb{S}^2}(\cdot; kt(r))$  evaluated at a pixel position  $u', v'$  is

$$\mathcal{L}_{\mathbb{S}^2}(u', v'; kt(r)) = \sum_{x=-w}^{x=w} \sum_{y=-w}^{y=w} f_p(u' + x, v' + y) \tilde{\mathcal{G}}_{\mathbb{S}^2}(x, y, kt), \quad (4.101)$$

where  $x, y$  are integer values and  $w$  is the size of the kernel  $\tilde{\mathcal{G}}_{\mathbb{S}^2}(\cdot; kt)$ .

As discussed, it is assumed that if the kernel  $\mathcal{G}_{\mathbb{S}^2}(\cdot; kt)$  were shifted so that it was centred at a point  $\mathbf{u}' \mapsto \eta'$ , then there is some scale  $kt(r)$  for which  $\mathcal{G}_{\mathbb{S}^2(\eta')}(\cdot; kt(r))$  is equal to  $\mathcal{G}_{\mathbb{S}^2}(\cdot; kt(r))$ . Therefore, it is assumed that there is some scale  $kt(r)$  for which

$$\tilde{\mathcal{G}}_{\mathbb{S}^2}(x, y; kt) = \mathcal{G}_{\mathbb{S}^2(\eta')}(u' + x, v' + y; kt(r)), \quad \forall x, y. \quad (4.102)$$

To find a scale-space image  $\mathcal{L}_{\mathbb{S}^2}(\cdot; kt)$  with a uniform diffusion scale  $kt$ ,  $\mathcal{L}_{\mathbb{S}^2}(u', v'; kt)$  would be evaluated as

$$\mathcal{L}_{\mathbb{S}^2}(u', v'; kt) = \sum_u \sum_v f_p(u, v) \mathcal{G}_{\mathbb{S}^2(\eta')}(u, v, kt). \quad (4.103)$$

pSIFT assumes that the value of  $\mathcal{L}_{\mathbb{S}^2}(u', v'; kt(r))$  obtained from 4.101 is approximately equal to

$$\begin{aligned} \mathcal{L}_{\mathbb{S}^2}(u', v'; kt(r)) &= \sum_{x=-w}^{x=w} \sum_{y=-w}^{y=w} f_p(u' + x, v' + y) \tilde{\mathcal{G}}_{\mathbb{S}^2}(x, y, kt) \\ &\approx \sum_u \sum_v f_p(u, v) \mathcal{G}_{\mathbb{S}^2(\eta')}(u, v, kt(r)), \end{aligned} \quad (4.104)$$

which is similar to 4.103 except that the scale  $kt(r)$  of the spherical Gaussian varies depending on the distance  $r$  of the point  $\mathbf{u}'$  from the principal point. Despite the fact that the non-uniform diffusion scale would not make the method suitable for uniform diffusion, for the purposes of scale-invariant keypoint detection this is not a limiting factor as the image is analysed across a wide range of scales.

#### 4.4.2.1 Scale correction

For reasons which will become clear in later discussions, it is necessary to find the scale  $kt(r)$  for which  $\tilde{\mathcal{G}}_{\mathbb{S}^2}(\cdot; kt)$ , centred at the point  $\mathbf{u}'$ , and  $\mathcal{G}_{\mathbb{S}^2(\eta')}(\cdot; kt(r))$  are most similar. This similarity could for example be measured by the Frobenius norm  $\|\tilde{\mathcal{G}}_{\mathbb{S}^2}(\cdot; kt) -$



$\mathcal{G}_{\mathbb{S}^2(\eta')}(\cdot; kt(r))\|_F$ . Unfortunately, a closed for solution to find the scale  $kt(r)$  which minimises this Frobenius norm has not been found. However, it is proposed that a suitable estimate of the scale  $kt(r)$  can be obtained by comparing the local sample rates  $\frac{d\psi}{dP}(r=0, \alpha)$  and  $\frac{d\psi}{dP}(r, \alpha)$  evaluated at the principal point and at the point  $\mathbf{u}'$  respectively, where  $r = \sqrt{(u' - u_0)^2 + (v' - v_0)^2}$  — see section 4.3.1.2, pg. 167 for a definition of the sample rate  $\frac{d\psi}{dP}(r, \alpha)$ .

The local sample rate  $\frac{d\psi}{dP}(r, \alpha)$  was derived for the unified camera model in section 4.3.1.2, and the solution for a parabolic catadioptric camera (i.e. a stereographic projection) with  $l_c = 1$  was given in equation 4.56 (pg. 170)<sup>7</sup>, where  $n_c = m_c + 1$ . This equation can be rewritten using  $m_c = m_p$  as

$$d\psi^2(r, \alpha) = \frac{4(m_p + 1)^2}{(r^2 + (m_p + 1)^2)^2} dP^2. \quad (4.105)$$

The ratio of the sample rate at a distance  $r$  from the principal point to that at the principal point itself is

$$\frac{\frac{d\psi^2}{dP^2}(r)}{\frac{d\psi^2}{dP^2}(0)} = \frac{(m_p + 1)^4}{((m_p + 1)^2 + r^2)^2}. \quad (4.106)$$

Since it was shown that  $kt \propto d\psi^2$  in section 4.3.2 (pg. 174), the estimate of the ‘corrected’ scale  $kt(r)$  is

$$kt(r) = \frac{kt(m_p + 1)^4}{((m_p + 1)^2 + r^2)^2}, \quad (4.107)$$

and is shown in figure 4.30 as a function of  $r$  for a range of scales  $kt$ .

Using this scale correction, the assumption made in equation 4.102 becomes

$$\tilde{\mathcal{G}}_{\mathbb{S}^2}(x, y; kt) = \mathcal{G}_{\mathbb{S}^2(\eta')} \left( u' + x, v' + y; \frac{kt(m_p + 1)^4}{((m_p + 1)^2 + r^2)^2} \right), \quad \forall x, y. \quad (4.108)$$

where  $\mathbf{u}' \mapsto \eta$  and  $r = \sqrt{(u' - u_0)^2 + (v' - v_0)^2}$ . Therefore, referring to equation 4.104, pSIFT assumes that the value of the scale-space image  $\mathcal{L}_{\mathbb{S}^2}(u', v'; kt(r))$  is approximately equal to

$$\mathcal{L}_{\mathbb{S}^2}(u', v'; kt(r)) \approx \mathcal{L}_{\mathbb{S}^2} \left( u', v'; \frac{kt(m_p + 1)^4}{((m_p + 1)^2 + r^2)^2} \right). \quad (4.109)$$

<sup>7</sup>The same result can be obtained by deriving the parameters  $d\phi, \sin^2(\theta)$  and  $d\theta^2$  in equations 4.51, 4.52 and 4.53 algebraically from equation 4.96, and then substituting into equation 4.45.

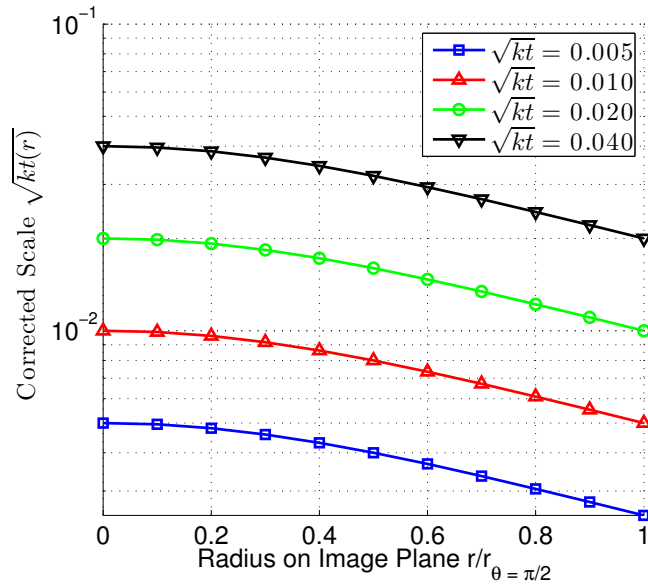


Figure 4.30: Corrected scale at a given radius on the image plane. The value  $r_{\theta=\pi/2}$  is the radius on the image plane which projects via inverse stereographic projection to a point  $\eta$  on the equator of the sphere.

#### 4.4.2.2 Approximation Error

The difference between  $\mathcal{G}_{\mathbb{S}^2(\eta')}(\cdot; kt(r))$  and  $\tilde{\mathcal{G}}_{\mathbb{S}^2}(\cdot; kt)$  centred at  $\mathbf{u}' \mapsto \eta'$  can be considered as the pSIFT approximation error  $\varepsilon$ . For a given scale  $kt$ , this error can be measured at any pixel position  $\mathbf{u}'$  from equation 4.108 as

$$\varepsilon = \left[ \sum_{x=-w}^{x=w} \sum_{y=-w}^{y=w} \left[ \tilde{\mathcal{G}}_{\mathbb{S}^2}(x, y; kt) - \mathcal{G}_{\mathbb{S}^2(\eta')} \left( u' + x, v' + y; \frac{kt(m_p + 1)^4}{((m_p + 1)^2 + r^2)^2} \right) \right]^2 \right]^{1/2}, \quad (4.110)$$

where  $r$  is the distance of the point  $\mathbf{u}'$  from the principal point. Figure 4.31a shows the error  $\varepsilon$  as a function of the radius  $r$  for a range of scales  $kt$  using the parabolic catadioptric camera model in figure 4.7. The same error  $\varepsilon$  found without scale correction ( $kt(r) = kt, \forall r$ ) is shown in figure 4.31b. The errors in both figures were obtained using the kernels  $\mathcal{G}_{\mathbb{S}^2(\eta')}(\cdot; kt(r))$  and  $\tilde{\mathcal{G}}_{\mathbb{S}^2}(\cdot; kt)$  computed up to  $b = 2048$  and normalised to have unit volume. Figure 4.32 illustrates the difference between  $\mathcal{G}_{\mathbb{S}^2(\eta')}(\cdot; kt(r))$  and  $\tilde{\mathcal{G}}_{\mathbb{S}^2}(\cdot; kt)$  with and without the scale correction for the largest scale  $\sqrt{kt} = 0.04$  and radius  $r/r_{\theta=\pi/2} = 1$  (the radius on the image plane which corresponds to a point on the equator of the sphere).

As expected there is a significant reduction in the approximation error using the scale correction factor. The results using the scale correction in figure 4.31a indicate that the error increases as the radius  $r$  increases and remains approximately equal for

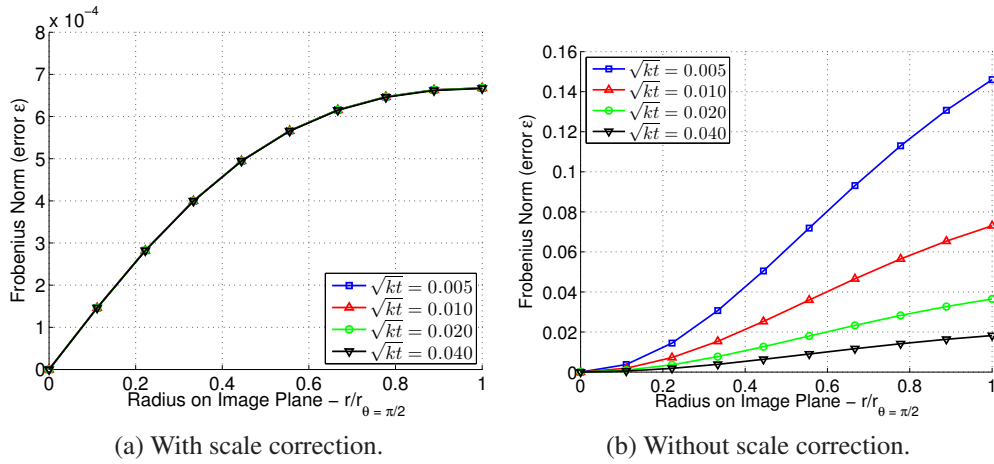


Figure 4.31: pSIFT approximation error versus radius  $r$  from the principal point for a range of scales  $kt$ . (a) shows the results with scale correction (equation 4.107), and (b) shows the results without scale correction. The value  $r_{\theta=\pi/2}$  is the radius on the image plane corresponding to a point on the equator of the sphere.

all scales  $kt$ . The largest error is less than 0.1% of the volume of  $\tilde{\mathcal{G}}_{\mathbb{S}^2}(\cdot; kt)$ . It is concluded from this result that the scale correction factor given in equation 4.107 is suitable. This scale correction factor is used later to correct the characteristic scale of pSIFT keypoints.

### 4.4.3 Scale-selection

The input scale  $kt_{input}$  and set of remaining scales  $kt$  used by pSIFT are selected using the method described in section 4.3.2 for  $n_{spo} = 3$  scales per octave. The original wide-angle camera model is used find the angle of colatitude  $\theta_s$  of a point on the sphere that projects to a point in the image at a radius of 1 pixel from the principal point. In all remaining experiments, pSIFT keypoints are detected in the first  $n_{oct} = 5$  octaves of scale-space.

### 4.4.4 Efficient Computation of Scale-Space Images

This section explores a number of the techniques, used by SIFT, that can be used by pSIFT to compute a set of scale-space images  $\mathcal{L}_{\mathbb{S}^2}(\cdot; kt(r))$  efficiently. These include separable convolution, cascade filtering, and an octave based approach to image processing which halves the image size after each octave of scale-space.

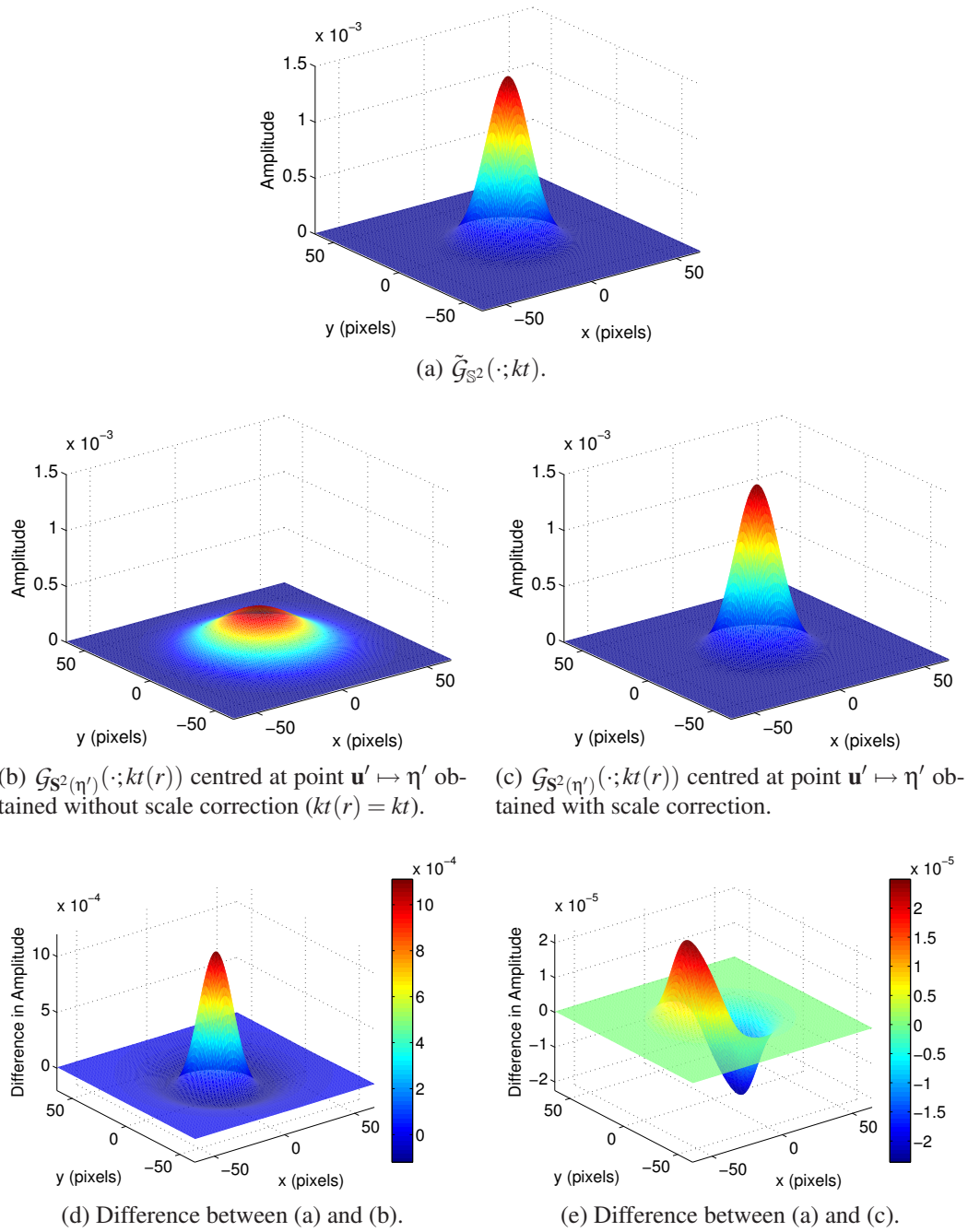


Figure 4.32: The kernel (a)  $\tilde{\mathcal{G}}_{\mathbb{S}^2}(\cdot; kt)$ , and the kernel  $\mathcal{G}_{\mathbb{S}^2(\eta')}(\cdot; kt(r))$  obtained (b) without scale correction, and (c) with scale correction— all kernels have been normalised to have unit volume. (d) and (e) illustrate the difference  $\tilde{\mathcal{G}}_{\mathbb{S}^2}(\cdot; kt) - \mathcal{G}_{\mathbb{S}^2(\eta')}(\cdot; kt(r))$  without and with scale correction respectively. All coordinates  $x, y$  are defined relative to the pixel at the centre of the kernel. The results are shown for a scale  $\sqrt{kt} = 0.04$  and a radius  $r_{\theta=\pi/2}$  (the radius on the image plane which corresponds to a point  $\eta'$  on the equator of the sphere).

#### 4.4.4.1 Separable Convolution

The two-dimensional discrete Gaussian  $G(\cdot; \sigma)$  is the only circular symmetrical kernel that is separable. It is a rank 1 matrix that can be written as the outer product of two one-dimensional Gaussian of the the same standard deviation  $\sigma$ :

$$G(\sigma) = G_y(\sigma)G_x(\sigma), \quad (4.111)$$

where  $G_x(\sigma)$  a row vector and  $G_y(\sigma)$  is a column vector. This property enables the convolution of an image with the Gaussian to be implemented efficiently by successive convolutions with  $G_x(\sigma)$  and  $G_y(\sigma)$ :

$$I * G(\cdot; \sigma) = G_x(\cdot; \sigma) * (I * G_y(\cdot; \sigma)). \quad (4.112)$$

This is referred to here as a ‘separable convolution’.

It was observed previously in figures 4.13a and 4.13b that the kernel  $\mathcal{G}_{\mathbb{S}^2}(\cdot; kt)$  on the stereographic image plane is similar in shape the the Gaussian  $G(\cdot; \sigma)$ . To the best of the author’s knowledge these two functions are not equivalent. This suggests that the square symmetric kernel  $\tilde{\mathcal{G}}_{\mathbb{S}^2}(\cdot; kt)$  is close to being a rank 1 matrix. This is confirmed with reference to figure 4.33a which shows the ratio of the second to first largest eigenvalues  $\lambda$  of  $\tilde{\mathcal{G}}_{\mathbb{S}^2}(\cdot; kt)$  computed up to  $b = 2048$  on a square  $241 \times 241$  pixel region for a range of scales  $kt$ . These scales correspond to the first 15 scales used by pSIFT for the fisheye camera calibrated in chapter 2. It is proposed that  $\tilde{\mathcal{G}}_{\mathbb{S}^2}(\cdot; kt)$  can be approximated as the outer product of a row and column vector. The convolution of  $I_p$  with  $\tilde{\mathcal{G}}_{\mathbb{S}^2}(\cdot; kt)$  can then be approximated by successive convolutions with these vectors, that is, using a separable convolution.

There are three ways that the row and column vectors could be selected whose outer product approximates  $\tilde{\mathcal{G}}_{\mathbb{S}^2}(\cdot; kt)$ :

1. Find  $\tilde{\mathcal{G}}'_{\mathbb{S}^2}(\cdot; kt)$ , the best rank 1 estimate of  $\tilde{\mathcal{G}}_{\mathbb{S}^2}(\cdot; kt)$  which minimises the Frobenius norm  $\|\tilde{\mathcal{G}}_{\mathbb{S}^2}(\cdot; kt) - \tilde{\mathcal{G}}'_{\mathbb{S}^2}(\cdot; kt)\|_F$ . The matrix  $\tilde{\mathcal{G}}'_{\mathbb{S}^2}(\cdot; kt)$  can be found from the eigen-decomposition of the square symmetric  $w \times w$  matrix  $\tilde{\mathcal{G}}_{\mathbb{S}^2}(\cdot; kt)$ . If  $\Lambda$  is the  $w \times w$  matrix whose diagonal values  $\{\lambda_1, \lambda_2, \dots, \lambda_w\}$  are the eigenvalues of  $\tilde{\mathcal{G}}_{\mathbb{S}^2}(\cdot; kt)$  in decreasing order of magnitude, and  $U$  is the  $w \times w$  matrix whose columns are the corresponding eigenvectors, then  $\tilde{\mathcal{G}}_{\mathbb{S}^2}(\cdot; kt) = U \Lambda U^T$ . The best rank 1 estimate of  $\tilde{\mathcal{G}}_{\mathbb{S}^2}(\cdot; kt)$  is  $U_1 \lambda_1 U_1^T$ , where  $U_1$  is the first column of  $U$ . Then  $I_p * \tilde{\mathcal{G}}_{\mathbb{S}^2}(\cdot; kt) \approx \lambda_1 U_1 * (I_p * U_1^T)$ .

2. Simply select the middle row  $\tilde{G}_{\mathbb{S}^2}(u_{mid}; kt)$  and column  $\tilde{G}_{\mathbb{S}^2}(v_{mid}; kt)$  of the kernel  $\tilde{G}_{\mathbb{S}^2}(\cdot; kt)$ . Then  $\tilde{G}_{\mathbb{S}^2}(\cdot; kt) \approx \tilde{G}_{\mathbb{S}^2}(v_{mid}; kt) \tilde{G}_{\mathbb{S}^2}(u_{mid}; kt)$ , and the error of this approximation can be measured as the Forbenius norm

$$\|\tilde{G}_{\mathbb{S}^2}(\cdot; kt) - \tilde{G}_{\mathbb{S}^2}(v_{mid}; kt) \tilde{G}_{\mathbb{S}^2}(u_{mid}; kt)\|_F.$$

Convolution would then be approximated as  $I_p * \tilde{G}_{\mathbb{S}^2}(\cdot; kt) \approx \tilde{G}_{\mathbb{S}^2}(u_{mid}; kt) * (I_p * \tilde{G}_{\mathbb{S}^2}(v_{mid}; kt))$ .

3. Out of interest, the best estimate of the Gaussian  $G(\cdot; \sigma)$  which approximates  $\tilde{G}_{\mathbb{S}^2}(\cdot; kt)$  could also be found, where best could be defined to mean the one which minimises the Frobenius norm  $\|\tilde{G}_{\mathbb{S}^2}(\cdot; kt) - G(\cdot; \sigma)\|_F$ . The convolution with  $G(\sigma)$  would then be approximated from equation 4.112 as  $I_p * \tilde{G}_{\mathbb{S}^2}(\cdot; kt) \approx G_x(\sigma) * (I_p * G_y(\sigma))$ .

In all cases, the vectors used for the separable convolution would need to be normalised to have unit volume.

Figure 4.33b shows, for each of the kernels  $\tilde{G}_{\mathbb{S}^2}(\cdot; kt)$  used in figure 4.33a (normalised to have unit volume), the Frobenius norm for each of the three methods discussed. The estimate of the Gaussian  $G(\cdot; \sigma)$  for method 3 was obtained using a non-linear optimisation. The results suggest that all methods provide an accurate estimate of  $\tilde{G}_{\mathbb{S}^2}(\cdot; kt)$ . Although the first method gives the most accurate results, the second method is used by pSIFT. The reason for selecting the second method over the first is the decreased computational cost as only the middle row or column needs to be found and not the whole kernel  $\tilde{G}_{\mathbb{S}^2}(\cdot; kt)$  — the values of the middle row and column vectors are identical as  $\tilde{G}_{\mathbb{S}^2}(\cdot; kt)$  is symmetrical.

#### 4.4.4.2 Cascade Filtering

Many scale-space keypoint detection algorithms such as SIFT use a cascade filtering approach to find scale-space images as

$$L(\cdot; \sigma_2) = G(\cdot; \sigma_2) * I \quad (4.113)$$

$$= G(\cdot; \sigma_{2-1}) * (G(\cdot; \sigma_1) * I) \quad (4.114)$$

$$= G(\cdot; \sigma_{2-1}) * L(\cdot; \sigma_1), \quad (4.115)$$

where  $\sigma_{2-1} = \sqrt{\sigma_2^2 - \sigma_1^2}$ . The same approach can be used to obtain scale-space images for wide-angle images as convolution with the spherical Gaussian satisfies the semi-

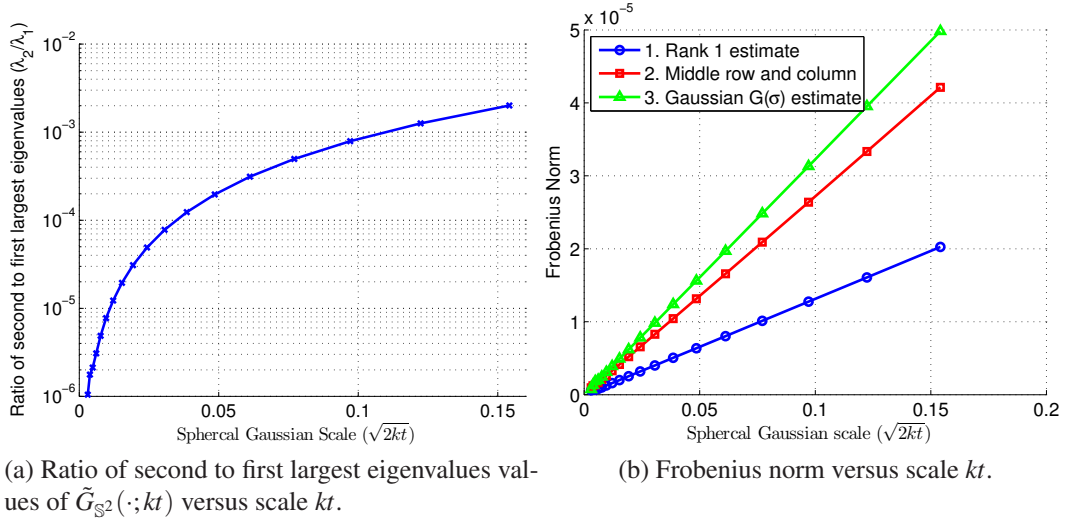


Figure 4.33: Error in the approximation of  $\tilde{G}_{\mathbb{S}^2}(\cdot; kt)$  as the outer product of a two vectors.

group property, as shown in equation 4.37:

$$L_{\mathbb{S}^2}(\cdot; kt_2(r)) = G_{\mathbb{S}^2}(\cdot; kt_2) * f_p \quad (4.116)$$

$$= G_{\mathbb{S}^2}(\cdot; kt_{2-1}) * (G_{\mathbb{S}^2}(\cdot; kt_1) * f_p) \quad (4.117)$$

$$= G_{\mathbb{S}^2}(\cdot; kt_{2-1}) * L_{\mathbb{S}^2}(\cdot; kt_1(r)), \quad (4.118)$$

where  $kt_{2-1} = kt_2 - kt_1$ . pSIFT can therefore use a cascade filtering approach to obtain scale-space images by

$$L_{\mathbb{S}^2}(\cdot; kt_2(r)) = \tilde{G}_{\mathbb{S}^2}(\cdot; kt_{2-1}) * L_{\mathbb{S}^2}(\cdot; kt_1(r)). \quad (4.119)$$

The advantage of this approach is that the size of the kernels used to find the scale-space images  $L_{\mathbb{S}^2}(\cdot; kt(r))$  remain as small as possible.

#### 4.4.4.3 Octave-based Approach

The final method that pSIFT uses to increase the speed of keypoint detection is the octave based approach used by SIFT, where the image size is halved after each octave. pSIFT doubles the size of the original stereographic image and finds the first scale-space image  $L_{\mathbb{S}^2}(\cdot; kt_0(r))$  by pre-smoothing the image to the starting scale  $kt_0$  with the kernel  $\tilde{G}_{\mathbb{S}^2}(\cdot; kt_{0-input})$  (pSIFT does not assume that  $L_{\mathbb{S}^2}(\cdot; 0) = I_{\mathbb{S}^2}$ ). The next  $n_{spo} + 2 = 5$  scale-space images are then found to obtain the set of six scale-space images  $L_{\mathbb{S}^2}(kt_{0,1,\dots,5})$  for the octave. The five difference of Gaussian images



$\mathcal{D}_{\mathbb{S}^2}(kt_{0,1,\dots,4})$  are obtained from these scale-space images from which the keypoint are detected in the middle three. Keypoints can only be detected in the middle three as they must be local extrema in scale and space. The fourth image in the stack,  $\mathcal{L}_{\mathbb{S}^2}(kt_3)$ , is then halved in size. This scale-space image is twice the scale of the first scale-space image in the stack ( $\sqrt{kt_3} = 2\sqrt{kt_0}$ ). This halved image becomes the first scale-space image in the next octave of scale-space. The process is then repeated for  $n_{oct}$  octaves of scale-space. This octave based approach, coupled with both the approximate separable convolution and cascade filtering, enables keypoints to be detected efficiently with the same computational order as SIFT.

Before proceeding, it is important to note that the position of the camera's principal point  $\mathbf{u}_0$  and the distance  $m_p$  of the stereographic image plane from the centre of the view sphere need to be updated each octave. This ensures that the correct kernels  $\tilde{\mathcal{G}}_{\mathbb{S}^2}(\cdot; kt)$  are used, and that the correct scale-correction factor derived in section 4.4.2.1 can be calculated.

#### 4.4.5 Keypoint Detection and Support Region

The keypoint detection stage of pSIFT is the same as sSIFT (see section 4.3). For each octave of scale-space, candidate keypoints are selected as local extrema in the difference of Gaussian images  $\mathcal{D}_{\mathbb{S}^2}(\cdot; kt(r))$ . Edge responses are then removed using the same edge removal threshold of  $r_{edge} = 10$ . Finally, the same quadratic interpolation is used to refine the position and scale of the keypoints. During keypoint detection, the non-linear diffusion scale  $kt(r)$  is effectively ignored. The initial characteristic scale of the keypoint is interpolated assuming that the difference of Gaussian images have a uniform diffusion scale — the difference of Gaussian images  $\mathcal{D}_{\mathbb{S}^2}(\cdot; kt(r))$  are treated as  $\mathcal{D}_{\mathbb{S}^2}(\cdot; kt)$ . It is only after the characteristic scale has been found, assuming that the difference of Gaussian images have a uniform diffusion scale  $kt$ , that the scale-correction factor described in section 4.4.2.1 is applied to find the corrected scale  $kt(r)$ . This value is then used to update the characteristic scale  $kt = kt(r)$  of the keypoint. The support region for a keypoint is defined as a circle on the sphere, centred at the keypoint position on the sphere, whose size is parameterised by the angle  $\psi_s = c\sqrt{2kt}$  (see figure 4.16, page 185).

One may argue that since stereographic projection is conformal, it would be suitable to define the support region for a pSIFT keypoint as a circle on the stereographic image. The problem with doing this is the fact the stereographic projection only *locally* preserves angles. To illustrate, figure 4.34 shows two views of the same scene

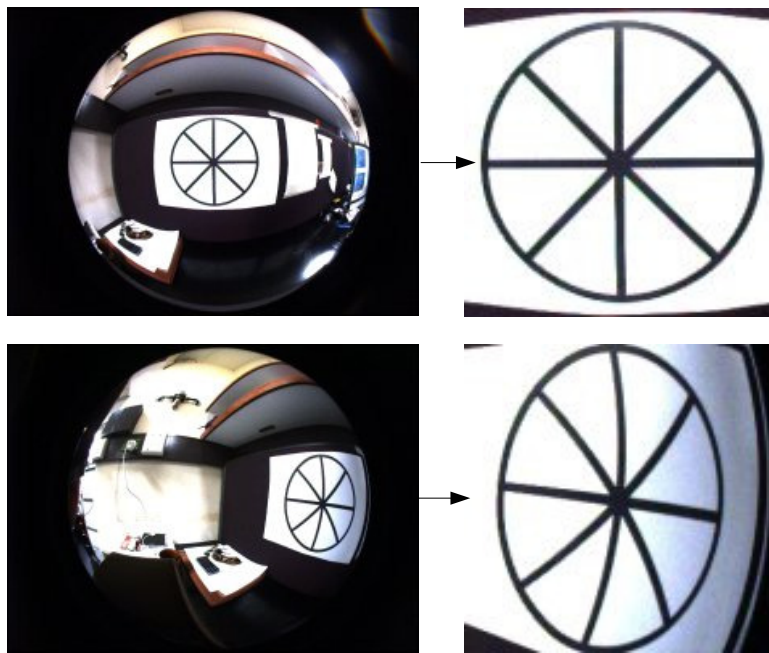


at different camera orientations. Figure 4.34a shows the original fisheye images, and figure 4.34b shows the fisheye images converted to stereographic images. With reference to the circular pattern, although the appearance remains more similar in the stereographic image compared to the fisheye image (up to a scale factor), there are still noticeable variations in the appearance. However, the appearance of the pattern on the view sphere would be identical up to a change in rotation. It is for this reason that the keypoint support region is set as a circle on the sphere.

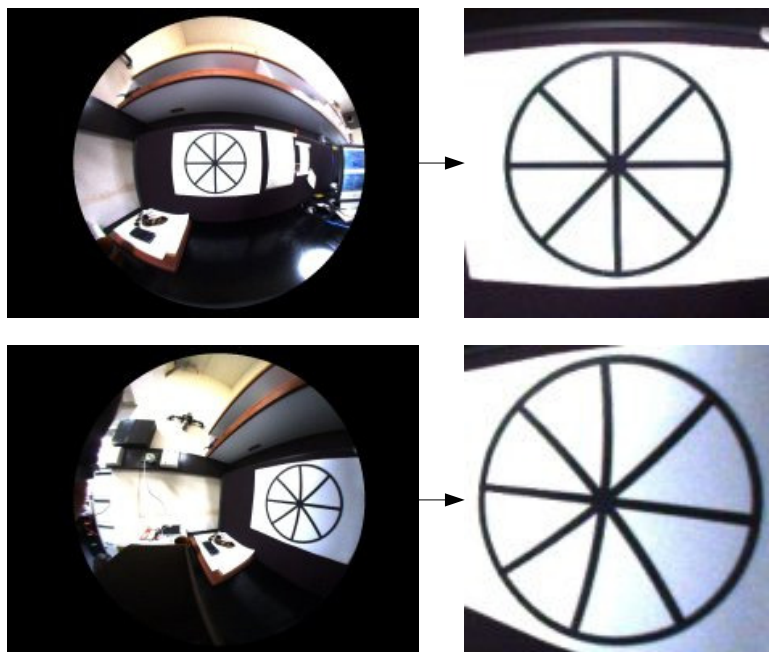
#### 4.4.6 Implementation

The details of pSIFT keypoint detection are summarised here. The original stereographic image's principal point and distance of the stereographic image from the view sphere are denoted  $\mathbf{u}_0$  and  $m_p$  respectively. The corrected values at a given octave are denoted  $\mathbf{u}'_0$  and  $m'_p$ .

1. Use the original wide-angle camera model to obtain the sample measurement  $\theta_s$  from which the input scale  $kt_{input}$  and set of remaining scales  $kt_{0,1,\dots,n_{oct}n_{spo}+2}$  are found, where  $n_{oct} = 5$  and  $n_{spo} = 3$  (see section 4.3.2).
2. Use the method described in section 4.4.1 to convert the wide-angle image  $I$  to an equal sized stereographic image  $I_p$ . The stereographic camera intrinsic parameters are the position  $\mathbf{u}_0$  of the principal point, and the distance  $m_p$  of the stereographic image from the centre of the view sphere.
3. Double the size of the stereographic image and find the new values for  $\mathbf{u}'_0$  and  $m'_p$ .
4. Find the pre-smoothing kernel  $\tilde{G}_{\mathbb{S}^2}(\cdot; kt_{0-input})$  and convolve the double sized stereographic image with this kernel to obtain the first scale-space image  $\mathcal{L}_{\mathbb{S}^2}(\cdot; kt_0(r))$  in the octave. The pre-smoothing kernel must be found using the correct distance  $m'_p$  for this octave. The starting scale for this octave is  $kt_{\kappa}$ , where  $\kappa = 0$ .
5. For  $i = 0 : n_{oct} + 2$ , compute the kernel  $\tilde{G}_{\mathbb{S}^2}(\cdot; kt_{i+1+\kappa} - kt_{i+\kappa})$  and convolve with  $\mathcal{L}_{\mathbb{S}^2}(\cdot; kt_{i+\kappa}(r))$  to find the scale-space image  $\mathcal{L}_{\mathbb{S}^2}(\cdot; kt_{i+1+\kappa}(r))$ . Use the correct value of  $m'_p$  for this octave.
6. For  $i = 0 : n_{oct} + 1$ , find the difference of Gaussian image  $\mathcal{D}_{\mathbb{S}^2}(\cdot; kt_{i+\kappa}(r)) = \mathcal{L}_{\mathbb{S}^2}(\cdot; kt_{i+1+\kappa}(r)) - \mathcal{L}_{\mathbb{S}^2}(\cdot; kt_{i+\kappa}(r))$ .



(a) Fisheye Image.



(b) Stereographic Image.

Figure 4.34: The change in appearance of a circular pattern in images of the same scene at different camera orientations. This change is shown as it appears in (a) the original fisheye image and (b) in the stereographic images. Although stereographic projection is conformal, and the appearance of the circular pattern in (b) remains approximately equal up to a scale change, there are still noticeable variations in the appearance. However, the appearance of this circular pattern on the view sphere would be identical up to a change in rotation.

7. Detect in the middle three of these difference of Gaussian images the pSIFT keypoints (see section 4.3). Obtain the corrected characteristic scale  $kt = kt(r)$  for each keypoint using the method described in section 4.4.2.1. The correct values of the principal point  $\mathbf{u}_0'$  and distance  $m'_p$  for this octave must be used to find the corrected scale. Set the support region angle as  $\psi_s = c\sqrt{2kt}$ .
8. Take the fourth ( $n_{spo} + 1$ ) scale-space image in the octave,  $\mathcal{L}_{\mathbb{S}^2}(\cdot; kt_{3+\kappa})$ , and halve it —  $kt_{3+\kappa} = 2kt_\kappa$ . This becomes the first scale-space image  $\mathcal{L}_{\mathbb{S}^2}(\cdot; kt_{3+\kappa})$  in the next octave of scale-space.
9. Increment the scale index,  $\kappa = \kappa + n_{spo}$ .
10. Repeat from step 5 for the required number of octaves.

Although not explicitly stated, the separable approach to convolution is used which requires finding only the middle row or column vector of the kernel  $\tilde{\mathcal{G}}_{\mathbb{S}^2}(\cdot; kt)$  for the required scale  $kt$ . These vectors are precomputed offline for  $l \in \{0, 1, \dots, 2048\}$  (see equation 4.98, pg. 209).

#### 4.4.7 Experiments: percentage correlation and number of correspondences

The experiments in section 4.3 were repeated using pSIFT. Note that only the fisheye images needed to be converted to stereographic images for pSIFT keypoint detection. For both the parabolic catadioptric and fisheye cameras, the input scale  $kt_{input}$  and set of scales  $kt$  were obtained as outlined in section 4.3.2 using the original parabolic catadioptric and fisheye camera models. The angle of the support region for the keypoints was set to  $\psi_s = \sqrt{2kt}$ . Although a full octave based approach can be used with pSIFT, in these experiments the image size was only halved after the first octave of scale-space to ensure the position and scale of keypoints were found accurately — the first octave still operates on the double sized image. In later experiments the full octave-based method is used which improves the speed of the algorithm.

##### 4.4.7.1 Results

The results for change in rotation and change in both rotation and scale are shown in figures 4.35 and 4.36. A summary of the results is presented in tables 4.4 and 4.5 for the parabolic catadioptric and fisheye camera respectively. The SIFT and sSIFT results have also been included for reference.

Image Transform	Keypoint Detector	DoG <sub>1</sub> = 0.01		DoG <sub>2</sub> = 0.02		DoG <sub>3</sub> = 0.03							
		% Correlation	# correspondences	% Correlation	# correspondences	% Correlation	# correspondences						
Rotation	SIFT	62.78	(62.54)	302.00	(458.14)	66.84	(66.28)	198.00	(289.57)	70.56	(69.72)	118.00	(166.83)
	sSIFT (256)	60.61	(57.04)	264.00	(440.40)	65.41	(61.44)	144.00	(229.12)	70.00	(66.23)	75.00	(117.04)
	sSIFT (256*)	76.53	(74.72)	261.00	(410.00)	79.73	(78.11)	139.00	(209.32)	82.83	(81.32)	71.00	(102.78)
	sSIFT (512)	70.71	(70.20)	260.00	(393.10)	74.72	(74.08)	166.00	(240.70)	78.60	(77.95)	95.00	(133.90)
	sSIFT (512*)	71.49	(71.09)	269.00	(403.68)	75.00	(74.78)	173.00	(249.91)	79.00	(78.78)	99.00	(138.72)
	pSIFT	62.80	(62.55)	302.00	(458.14)	66.85	(66.29)	198.00	(289.57)	70.55	(69.72)	118.00	(166.84)
Rotation & Scale	SIFT	30.63	(31.77)	151.00	(181.04)	34.11	(34.74)	104.00	(122.35)	37.36	(37.35)	60.00	(73.08)
	sSIFT (256)	10.40	(11.43)	46.00	(61.62)	14.89	(16.33)	35.00	(44.50)	19.59	(21.22)	23.00	(28.32)
	sSIFT (256*)	14.02	(16.44)	50.00	(65.74)	18.80	(21.50)	36.00	(43.96)	23.41	(26.02)	21.00	(26.11)
	sSIFT (512)	25.84	(28.54)	103.00	(123.17)	29.49	(31.96)	70.00	(83.27)	33.50	(35.38)	41.00	(49.54)
	sSIFT (512*)	26.31	(28.68)	108.00	(126.63)	29.88	(32.03)	75.00	(86.60)	34.26	(35.55)	43.00	(51.49)
	pSIFT	30.66	(31.77)	151.00	(181.03)	34.13	(34.74)	104.00	(122.34)	37.37	(37.35)	60.00	(73.08)

Table 4.4: Median percentage correlation and number of correspondences for the parabolic catadioptric camera using SIFT, sSIFT and pSIFT (mean values shown in brackets). DoG is the difference of Gaussian threshold.

Image Transform	Keypoint Detector	$DoG_1 = 0.01$		$DoG_2 = 0.02$		$DoG_3 = 0.03$	
		% Correlation	# correspondences	% Correlation	# correspondences	% Correlation	# correspondences
Rotation	SIFT	33.65 (34.73)	211.00 (400.39)	35.15 (35.87)	129.00 (241.26)	36.48 (37.05)	72.00 (134.67)
	sSIFT (256)	56.43 (52.76)	274.00 (470.42)	60.43 (56.50)	145.00 (238.67)	65.03 (61.03)	76.00 (121.61)
	sSIFT (256*)	74.28 (72.26)	267.00 (417.45)	77.10 (75.40)	139.00 (209.73)	80.12 (78.43)	68.00 (101.43)
	sSIFT (512)	67.99 (66.67)	480.00 (768.99)	72.91 (71.60)	262.00 (404.80)	77.60 (76.34)	133.00 (201.62)
	sSIFT (512*)	73.17 (72.29)	542.00 (864.16)	76.56 (75.87)	287.00 (439.53)	80.08 (79.31)	139.00 (211.40)
	pSIFT	59.60 (59.64)	465.00 (739.98)	66.15 (65.79)	263.00 (397.53)	71.43 (70.84)	137.00 (205.34)
Rotation & Scale	SIFT	11.22 (14.78)	78.00 (116.36)	11.85 (16.10)	51.00 (76.49)	12.48 (17.32)	29.00 (44.70)
	sSIFT (256)	8.32 (9.25)	41.00 (56.63)	12.28 (13.61)	31.00 (41.28)	16.44 (18.03)	21.00 (26.68)
	sSIFT (256*)	13.05 (15.15)	48.00 (63.62)	17.69 (19.99)	34.00 (42.29)	21.95 (24.32)	20.00 (24.87)
	sSIFT (512)	20.67 (22.70)	152.00 (196.44)	25.29 (27.39)	96.00 (120.13)	29.83 (31.91)	52.00 (66.29)
	sSIFT (512*)	19.80 (22.50)	157.00 (200.88)	24.37 (27.19)	98.00 (121.78)	29.25 (31.71)	53.00 (66.26)
	pSIFT	23.54 (25.61)	191.00 (243.22)	27.81 (29.64)	115.00 (141.48)	32.03 (33.35)	61.50 (77.33)

Table 4.5: Median percentage correlation and number of correspondences for the fish-eye camera using SIFT, sSIFT and pSIFT (mean values shown in brackets). DoG is the difference of Gaussian threshold.

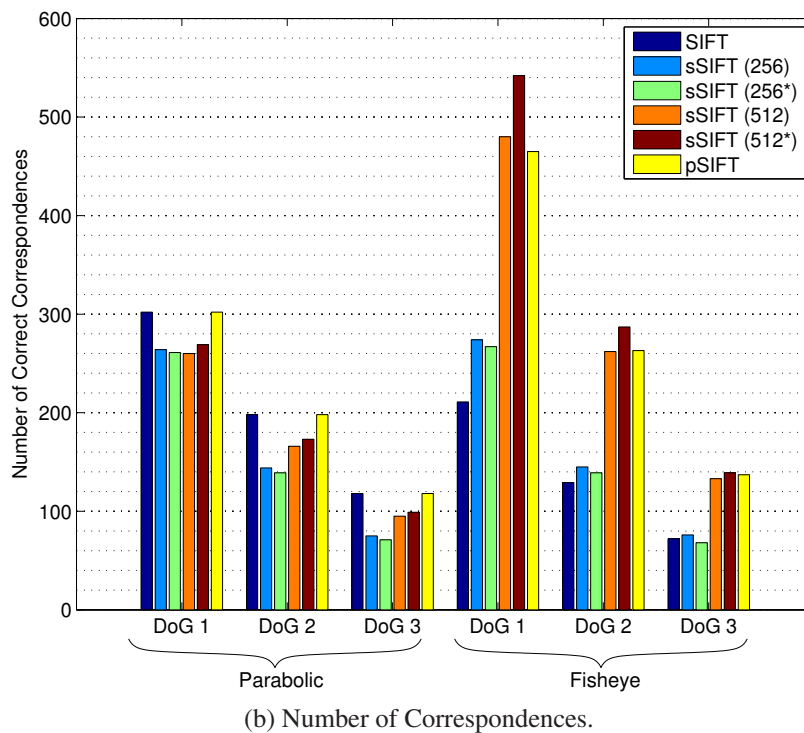
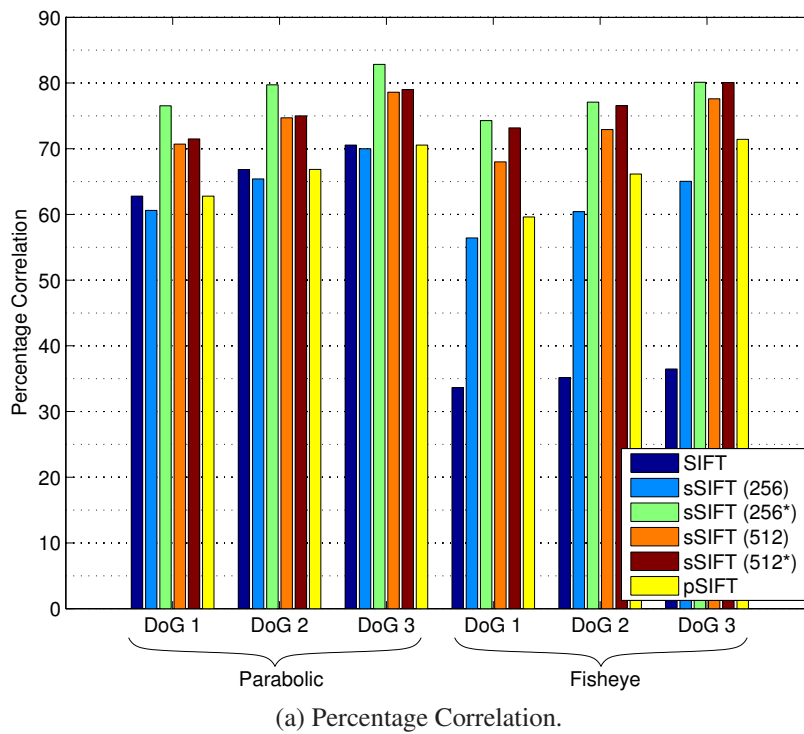


Figure 4.35: Median percentage correlation and number of correspondences for images subject to change in rotation for SIFT, sSIFT and pSIFT. The difference of Gaussian thresholds are  $DoG_1 = 0.01$ ,  $DoG_2 = 0.02$  and  $DoG_3 = 0.03$ .

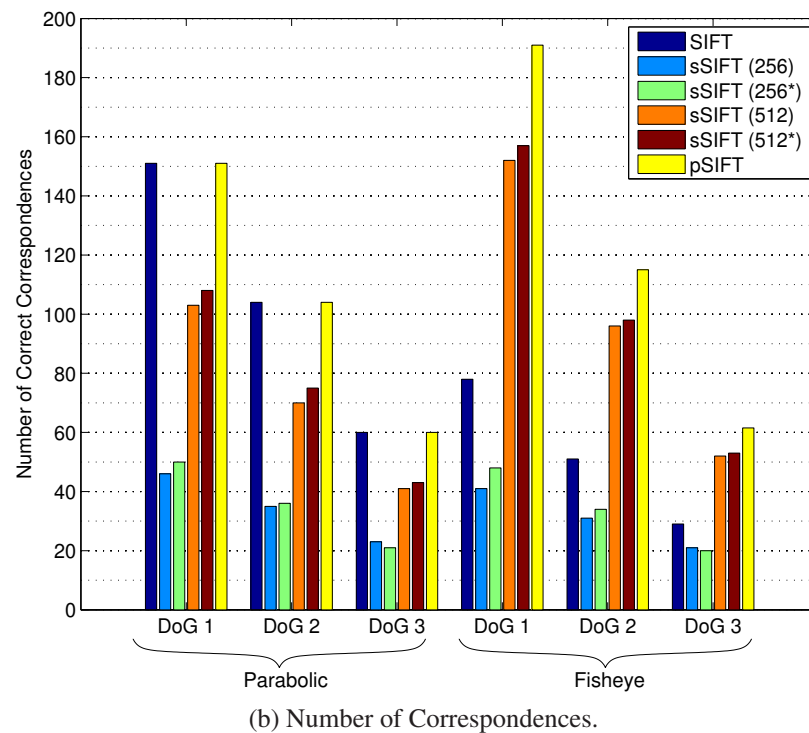
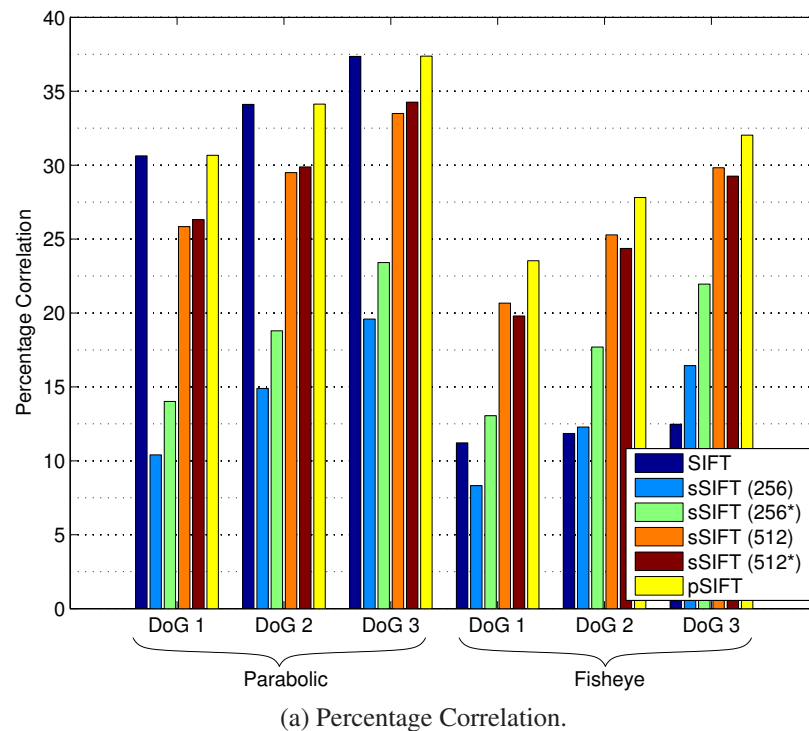
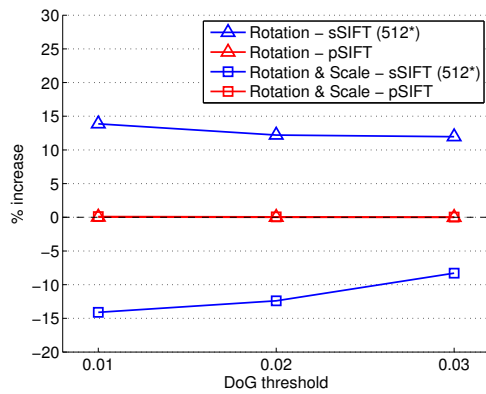
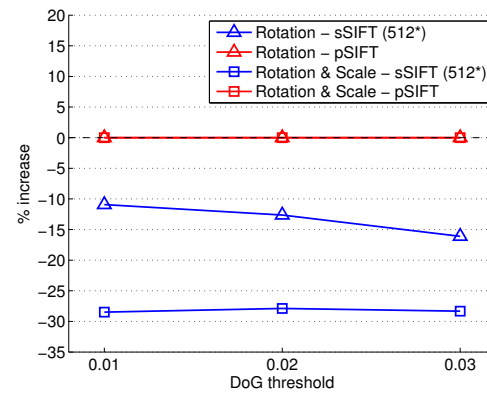


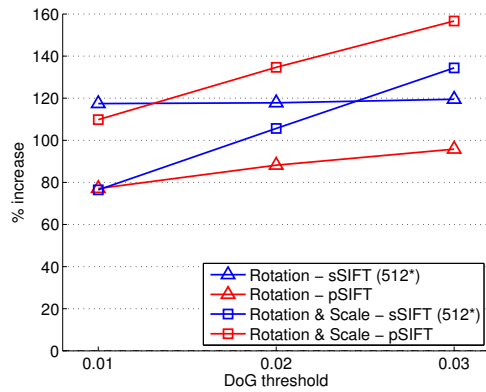
Figure 4.36: Median percentage correlation and number of correspondences for images subject to change in both rotation and scale for SIFT, sSIFT and pSIFT. The difference of Gaussian thresholds are  $DoG_1 = 0.01$ ,  $DoG_2 = 0.02$  and  $DoG_3 = 0.03$ .



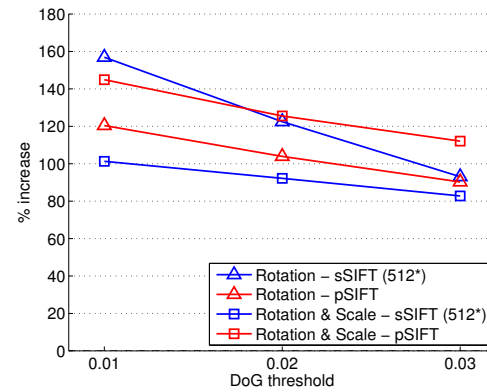
(a) Increase in percentage correlation for the parabolic camera.



(b) Increase in number of correspondences for the parabolic camera.



(c) Increase in percentage correlation for the fisheye camera.



(d) Increase in number of correspondences for the fisheye camera.

Figure 4.37: Percentage increase in the percentage correlation of keypoints and the overall number of correspondences for sSIFT(512\*) and pSIFT relative to SIFT.

#### 4.4.7.2 Discussion

The increase in the percentage correlation and number of correspondences for each camera using pSIFT over SIFT is shown in figure 4.37. This figure also shows the results for sSIFT (512\*) presented in figure 4.27. Overall, the results obtained in these experiments show that for both cameras, pSIFT gave improved or comparable performance to SIFT.

The results in figures 4.37a and 4.37b show that there was almost no increase in performance using pSIFT for the parabolic catadioptric camera. This result is not surprising considering that the  $\tilde{G}_{S^2}(\cdot; kt)$  and  $G(\cdot; \sigma)$  are similar in shape on a stereographic image, as shown previously in figure 4.13. Furthermore, as stereographic projection is a conformal mapping, setting a circular support region for SIFT keypoints on a parabolic catadioptric image closely approximates a circular support region defined



on the sphere. With the exception of the percentage correlation for change in rotation, pSIFT outperforms sSIFT(512\*) for all DoG thresholds. The most likely explanation as to why sSIFT(512\*) has an improved percentage correlation for change in rotation is the fact that the scale-space images  $\mathcal{L}_{\mathbb{S}^2}(\cdot; kt)$  obtained by sSIFT have a uniform diffusion scale  $kt$ . This means that for any two images obtained at different camera orientations, a keypoint can be detected in the same range of scales  $kt$  irrespective of its position in the image. In contrast, the set of scale-space images  $\mathcal{L}_{\mathbb{S}^2}(\cdot; kt(r))$  obtained by pSIFT having a non-uniform diffusion scale  $kt(r)$  (see figure 4.30, pg. 212). This reduces the range of scales  $kt$  in which the same keypoint can be detected at any position in the image.

The results for the fisheye camera in figures 4.37c and 4.37d indicate that when compared to SIFT, an increase in both the percentage correlation and the number of correspondences is found using pSIFT for all DoG thresholds and each camera transform. Although the results show that pSIFT outperforms sSIFT in the percentage correlation and number of correspondences for change in both rotation and scale, the opposite is true for change in rotation. This can again be attributed to the fact that pSIFT detects keypoints in the set of scale-space images  $\mathcal{L}_{\mathbb{S}^2}(\cdot; kt(r))$  having a non-uniform diffusion scale.

#### 4.4.8 Experiments: performance versus image position

It is of interest to compare how reliably SIFT, sSIFT and pSIFT can detect corresponding keypoints in different wide-angle images of the same scene as a function of image position. For example, can one keypoint detector find correspondences towards the peripheries of the images more reliably than the others. The experiments in this section consider how this performance varies versus distance from the principal point in a wide-angle image, which is represented as an angle of colatitude  $\theta$  on the sphere. Given any two images, this reliability is measured as the ratio of the number of correspondences in an image versus the number of all keypoints in an image within a range of angles of colatitude.

In the previous experiments, SIFT, sSIFT and pSIFT keypoints were detected in different synthetically generated wide-angle images. The correspondences were then found between images separated by a change in rotation, and both rotation and scale-change using the methodology in section 4.3.6.3. This same data set is used in these experiments (i.e. the keypoints and the correspondences found). Note that the angle of colatitude of all keypoints is known, and the set of corresponding keypoints between

image pairs in a given reference image set is also known.

For each camera and keypoint detector, each pair of images in a reference image set separated by a change in rotation, or a change in both rotation and scale was taken. For each of these images, the number of keypoints with an angle of colatitude within some fixed interval was found, and the number of correspondences with an angle of colatitude within this same fixed interval was found. These results were accumulated for all 40 reference image sets separately for each camera, keypoint detector and image transform (rotation, and rotation and scale change). The probability distribution of the angles of colatitude of all the keypoints was then be found. For each of the angle of colatitude intervals, the ratio of the number of correspondences versus the number of all keypoints was then found. Ten equal sized intervals were used within the range of angles  $\theta = 0$  to  $\theta = \pi/2$ . The centres of the  $n = 10$  intervals are

$$\theta_i = \frac{\pi}{2n}i + \frac{\pi}{4n}, \quad i \in \{1, 2, \dots, n\}. \quad (4.120)$$

#### 4.4.8.1 Results

The results are presented in figure 4.38 for the parabolic catadioptric camera and figure 4.39 for the fisheye camera. Note that the probability distributions are shown as line plots for convenience. The values in the figures are marked at the centre of the angle of colatitude intervals.

#### 4.4.8.2 Discussion

For both cameras and image transforms (rotation, and both rotation and scale), the probability distributions of all keypoints are very similar. However, there are significant variations in the reliabilities of the keypoint detectors (i.e. the ratio of the number of correspondences versus the number of keypoints).

The first observation that can be made is that in general, the use of the anti-aliasing interpolation filter used by sSIFT improves the results for all angles of colatitude. The exception is for a change in both rotation and scale for the fisheye camera in figure 4.39b where sSIFT(512) performs better than sSIFT(512\*).

For change in rotation for the parabolic camera, overall the reliability of sSIFT (excluding sSIFT(256)) is better than SIFT and pSIFT for large angles of colatitude, that is, for points near the periphery of the image. Although sSIFT(256\*) outperforms all other keypoint detectors for a change in rotation with the parabolic camera across all

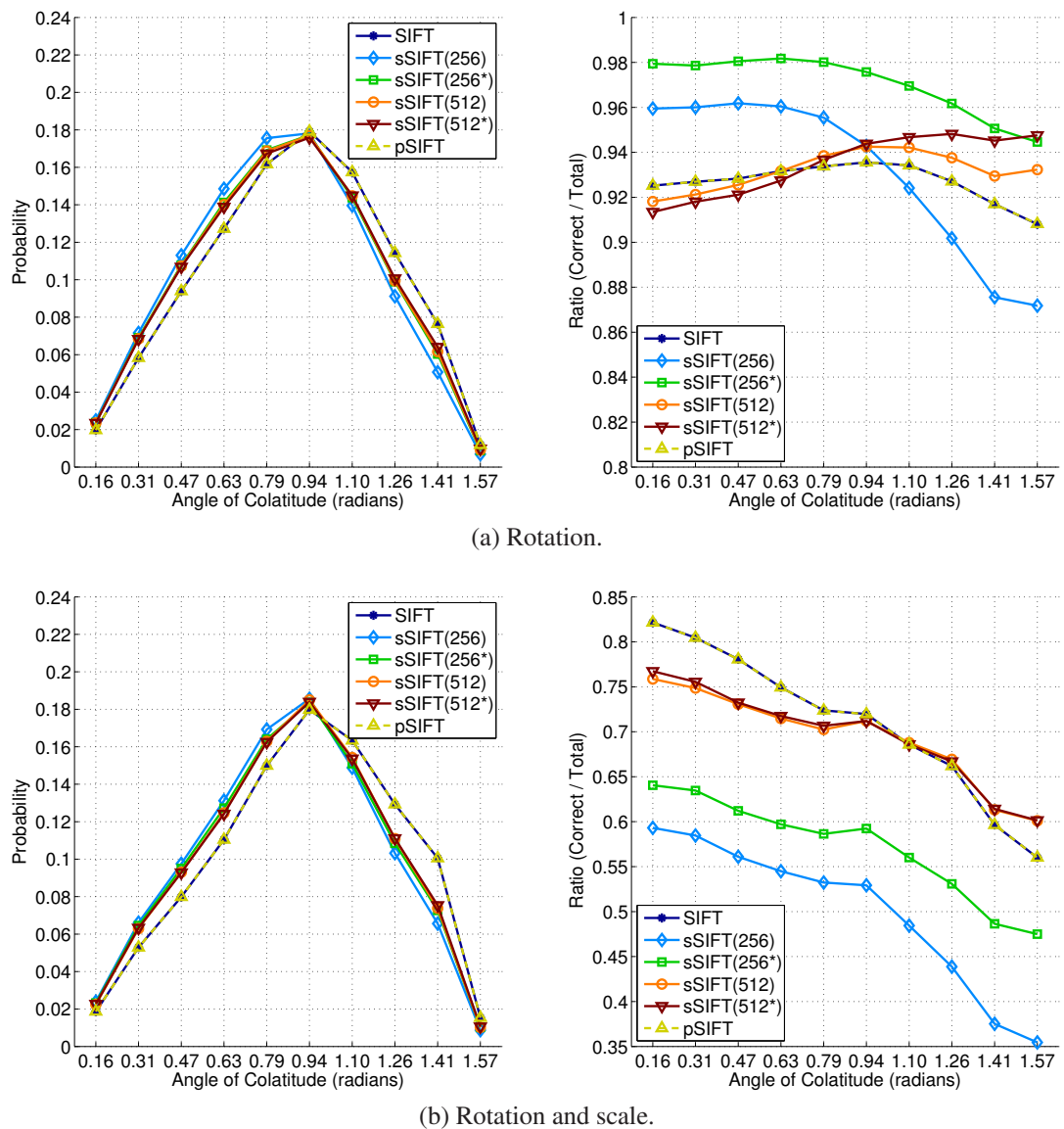


Figure 4.38: The probability distribution of all the keypoints (left) and the ratio of the number of correspondences versus the number of all keypoints as function of angle of colatitude (right) for the parabolic catadioptric camera. Observe that the results for SIFT and pSIFT are nearly identical.

angles of colatitude, the results for a change in both scale and rotation in figure 4.38b are very different, and SIFT and pSIFT perform consistently well compared to all others for all angles of colatitude — it is only for the largest angles of colatitude that they are less reliable than sSIFT(512) and sSIFT(512\*). It can also be seen in figure 4.38b that overall the reliability of all keypoint detectors decreases with increasing angle of colatitude. As was the case in the previous experiments, SIFT and pSIFT have nearly identical results for the parabolic camera for both image transforms.

The results for the fisheye camera in figure 4.39 show that overall SIFT performs

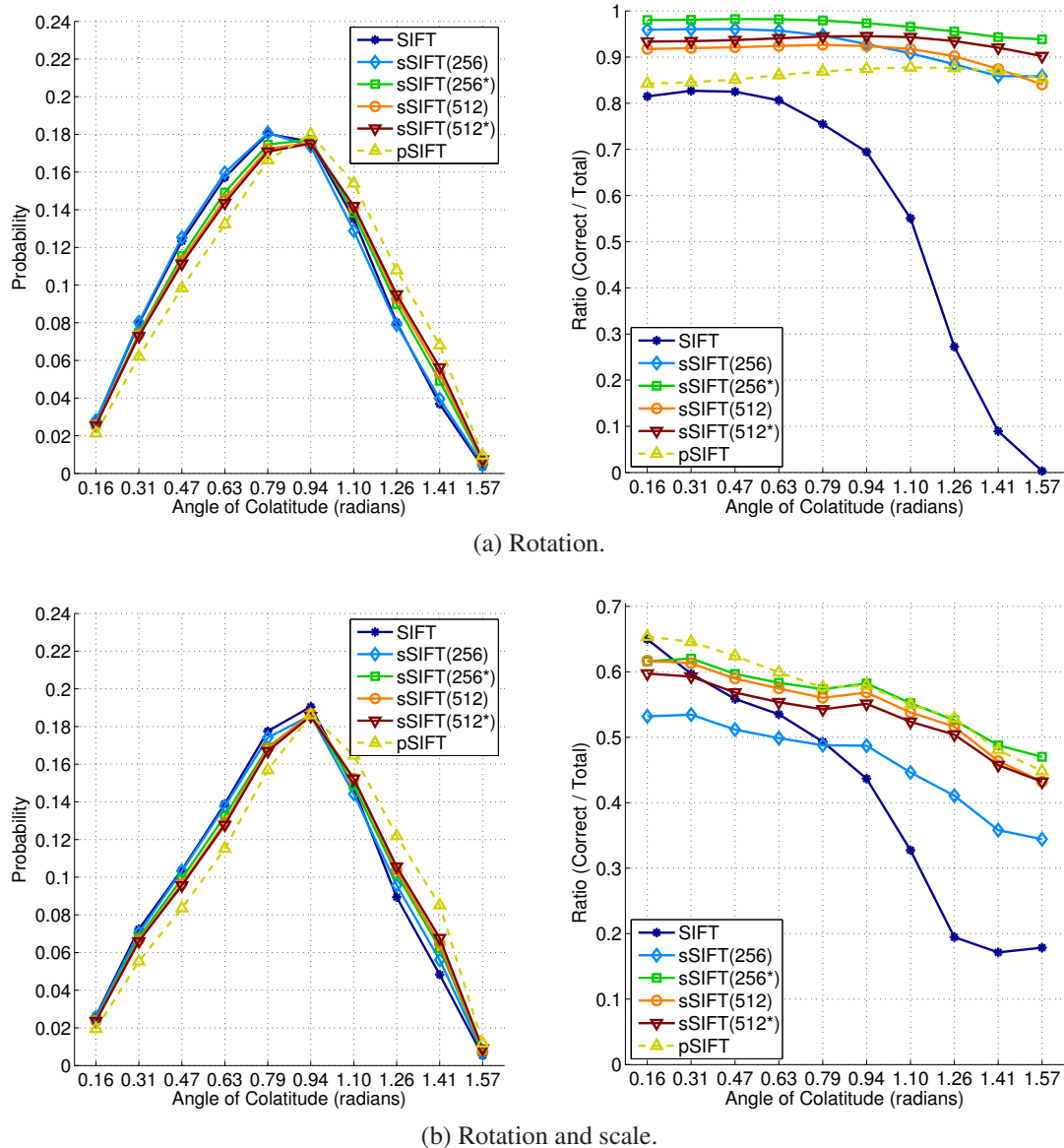


Figure 4.39: The probability distribution of all the keypoints (left) and the ratio of the number of correspondences versus the number of all keypoints as function of angle of colatitude (right) for the fisheye camera.

poorly in comparison to sSIFT and pSIFT for both image transforms, especially for large angles of colatitude where the reliability of SIFT is significantly less than that of sSIFT and pSIFT. The results in figure 4.39a show that, for a change in rotation, sSIFT overall outperforms pSIFT for all angles of colatitude. As discussed previously, pSIFT finds scale-space images with a non-uniform diffusion scale, and this can limit its ability to find correspondences between images separated by a change in camera rotation. However, for changes in both scale and rotation, figure 4.39b shows that, compared to the other keypoint detectors, pSIFT ranks consistently high for the fisheye camera.

### 4.4.9 Conclusions

The pSIFT keypoint detector was developed in this section. As was the case for sSIFT, the scale-space representation of a wide-angle image is defined as the convolution of the spherical Gaussian with the wide-angle image mapped to the sphere. pSIFT approximates this convolution operation efficiently as a convolution operation on the stereographic image plane. Although the resulting approximating produces scale-space images with a non-uniform diffusion scale, this is not a major problem as keypoints are detected in the image across a wide range of scales. Unlike, sSIFT, pSIFT is not limited by sample rate problems. pSIFT detects keypoints as local extrema in the difference of scale-space images, and uses the same edge removal and keypoint interpolation schemes used by sSIFT. The support region for a keypoint is a circle on the sphere that is centred at the position of the keypoint on the sphere, and whose size is set relative to the characteristic scale of the keypoint.

The results to date give evidence that both sSIFT(512\*) and pSIFT are more suited to scale-invariant keypoint detection in wide-angle images than SIFT. However, only the keypoint detection stages has been compared. The next section describes the method used to evaluate SIFT descriptors for both sSIFT and pSIFT keypoints. By matching sSIFT or pSIFT descriptors, correspondences can be found between two images of the same scene.

## 4.5 sSIFT and pSIFT Keypoint Descriptors

For any sSIFT or pSIFT keypoint, the greyscale intensity values within the keypoint's circular support region on the sphere are projected to a fixed sized patch. A SIFT descriptor for the keypoint is then evaluated for this patch. Figure 4.40 illustrates the circular support region, parameterised by the angle  $\psi_s = c\sqrt{2kt}$ , for a keypoint with characteristic scale  $kt$ . This angle is measured from the axis that passes through the centre of the sphere and the keypoint's position  $\eta$  on the sphere. Referring to the same figure,  $\psi$  is any angle from this axis within the support region, and  $\beta$  as an angle of rotation about this axis.

It is necessary to find a suitable method for projecting the greyscale intensity values within a keypoint's support region to the fixed sized patch. One solution would be to align the planar patch orthogonal to the axis that passes through the centre of the sphere and the keypoint's position  $\eta$  on the sphere, and then project the intensity values to the patch using either a perspective projection (from the centre of the sphere)

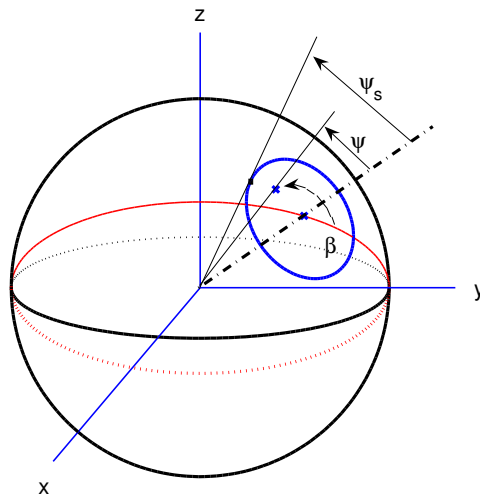


Figure 4.40: The radius of the support region for an sSIFT or pSIFT keypoint  $\psi_s = c\sqrt{kt}$ , where  $kt$  is the characteristic scale of the keypoint. This angle is measured from the axis that passes through the centre of the sphere and the position of the keypoint on the sphere.  $\psi$  is any angle from this axis within the support region, and  $\beta$  is an angle of rotation about this axis.

or stereographic projection (from the point  $-\eta$ ). To determine which projection is most suitable, it is argued that for any keypoint with characteristic scale  $kt$  detected at position  $\eta$ , the appearance of the spherical Gaussian  $G_{\mathbb{S}^2(\eta')}(\cdot; kt)$  centred at  $\eta$  should appear identical when projected to the fixed sized patch.

It was observed from the results in figure 4.12 (pg.178) that the ratio  $G_{\mathbb{S}^2}(\theta = \sqrt{2kt}, \phi; kt) / G_{\mathbb{S}^2}(0, \phi; kt)$  remains constant for all scales  $kt$ . It is assumed then that, with reference to figure 4.40, the spherical Gaussian  $G_{\mathbb{S}^2(\eta')}(\psi, \beta; kt)$  can be written in scale-normalised coordinates as

$$G_{\mathbb{S}^2(\eta')} \left( \psi', \alpha'; \frac{1}{c^2} \right) = G_{\mathbb{S}^2(\eta')}(\psi, \alpha; kt), \quad (4.121)$$

where  $\psi_s = c\sqrt{2kt}$ ,  $\psi' = \frac{\psi}{\psi_s}$  and  $\alpha' = \alpha$ . To ensure that the spherical Gaussian appears the same when projected to the fixed sized patch, the projection needs to be a function of the scale normalised coordinates  $\psi', \beta'$ :

$$r_p = f(\psi'), \quad (4.122)$$

$$\zeta_p = \beta' + N, \quad (4.123)$$

where  $r_p$  and  $\zeta_p$  are the polar coordinates of a point on the patch, and  $N$  is any real constant.

If a perspective projection were used to map the greyscale intensity values within the support region to the patch, where the patch is orthogonal to the axis passing through the centre of the sphere and the keypoint position  $\eta$  on the sphere, then

$$r_p = \frac{1}{2}(p-1) \left( \frac{\tan(\psi)}{\tan(\psi_{support})} \right), \quad \zeta_p = \beta'. \quad (4.124)$$

If a stereographic projection from the point  $-\eta$  were used, then

$$r_p = \frac{1}{2}(p-1) \left( \frac{\tan\left(\frac{\psi}{2}\right)}{\tan\left(\frac{\psi_{support}}{2}\right)} \right), \quad \zeta_p = \beta'. \quad (4.125)$$

Both perspective and stereographic projection are not ideal as  $r_p$  cannot be written as a function of the scale normalised coordinate  $\psi'$ . Perspective projection would also be unsuitable if  $\psi_s \geq \pi/2$ . It is proposed that an *equiangular* projection can be used, where

$$r_p = \frac{1}{2}(p-1) \left( \frac{\psi}{\psi_{support}} \right) \quad (4.126)$$

$$= \frac{1}{2}(p-1)\psi' \quad (4.127)$$

$$\zeta_p = \beta'. \quad (4.128)$$

This projection is termed equiangular as the sample rate  $\partial\psi/\partial r_p$  is constant.

Up to now, the angle  $\beta$  has been defined simply as an angle about the axis that passes through the centre of the view sphere and the keypoint position  $\eta$ . If  $R = R_y R_z$  is the rotation matrix that rotates the north pole  $\mathbf{n}$  to the point  $\eta = R\mathbf{n}$ , then  $R^T \eta'(\psi, \beta) = \eta''(\theta, \phi)$ . Here,  $\eta'(\psi, \beta)$  is a point on the sphere parameterised by the angles  $\psi, \beta$  from  $\eta$ , and  $\eta''(\theta, \phi)$  is the point on the sphere at an angle of colatitude  $\theta$  and angle of longitude  $\phi$ . For a keypoint detected in the difference of Gaussian image  $\mathcal{D}_{\mathbb{S}^2}(\cdot; kt_i)$ , for each pixel on the patch, the point  $\eta'(\psi, \beta)$  is found. The position  $\mathbf{u}'$  of the point  $\eta'(\psi, \beta)$  in the scale-space image  $\mathcal{L}_{\mathbb{S}^2}(\cdot; kt_i)$  is then found, and a linear interpolation used to sample the greyscale intensity value  $\mathcal{L}_{\mathbb{S}^2}(u', v'; kt_i)$ .

The size of the support region for an sSIFT or pSIFT keypoint is set to  $\psi_s = 10\sqrt{2kt}$ , where  $kt$  is the characteristic scale of the keypoint. The size of the patch used is  $41 \times 41$  pixels, which was the same size used to evaluate keypoint descriptors by Ke and Sukthankar [120] and Mikolajczyk et al [203]. The 128 dimensional SIFT descriptor is evaluated from the greyscale intensity values on the patch, and consists of  $4 \times 4$  histograms of weighed gradient orientations, each with 8 histogram bins —



refer to section 3.3.3.1 (pg.129) for a detailed description of the SIFT descriptor. The gradient magnitudes and orientations used to find the SIFT descriptor are calculated at each point  $\mathbf{x}$  on the patch  $p$  as<sup>8</sup>

$$\begin{aligned} dx(x,y) &= p(x+1,y) - p(x-1,y) \\ dy(x,y) &= p(x,y+1) - p(x,y-1) \\ \phi(x,y) &= \arctan\left(\frac{dy(x,y)}{dx(x,y)}\right). \end{aligned} \quad (4.129)$$

## 4.5.1 Experiments

Experiments were conducted to determine empirically which of the perspective, stereographic and equiangular projections is most suited for mapping the greyscale intensity values within a keypoint's support region to the fixed sized patch.

### 4.5.1.1 Data

For each of the sSIFT(512\*) keypoints detected in the wide-angle images in the previous experiments, the size of the support region for each keypoint was set to  $\psi_s = 10\sqrt{2kt}$ . A separate SIFT descriptor was then evaluated for each keypoint using a perspective, stereographic and equiangular projection to map the greyscale intensity values within the keypoint's support region to the fixed sized  $41 \times 41$  pixel patch. The size of the support region  $\psi_s$  never exceeded  $\pi/2$ , so the perspective projection could be used for all keypoints.

For each of the 40 reference images, keypoint correspondences were found between all image pairs in the set using the SIFT descriptors (there are 45 parabolic catadioptric and 45 fisheye images for each reference image). For a pair of images, the correspondences were found as follows. For each keypoint in the image with the fewest number of keypoints, the Euclidean distances between its SIFT descriptor and the descriptors of all the keypoints in the other image were found. The corresponding keypoint in the other image was selected as the one with the smallest Euclidean distance between the descriptors. The similarity of this correspondence was then defined using the ambiguity metric, which is the ratio of the smallest to second smallest of the Euclidean distances. The smaller the ambiguity, the more similar the keypoints are

<sup>8</sup>It is possible to compute the derivatives of the scale-space function  $L_{S^2}(\cdot; kt)$  in orthogonal directions on the sphere using directional filters [247]. However, this is far more computationally expensive than simply evaluating the derivatives from the greyscale intensity values on the patch.



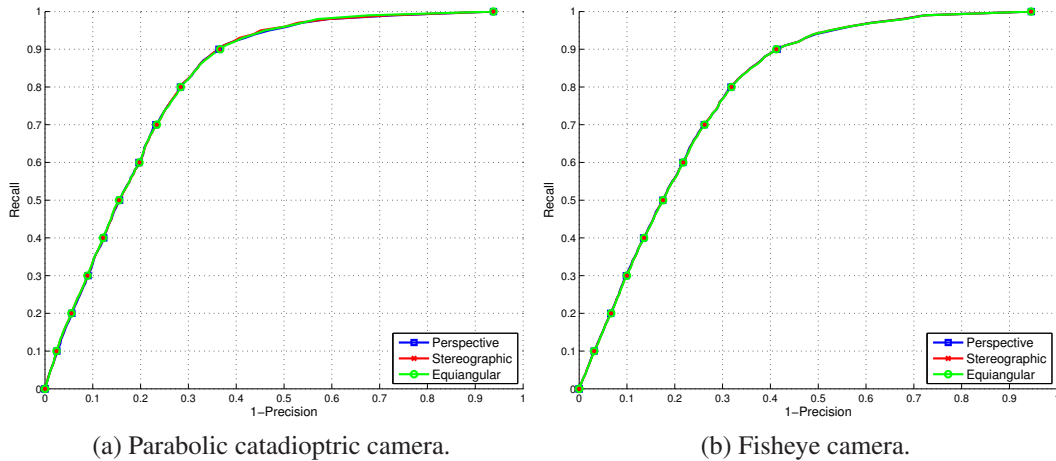


Figure 4.41: Recall versus 1-precision results for each camera using a perspective, stereographic and equiangular projection to map the greyscale intensity values within a keypoint's support region to a fixed sized patch.

considered to be. The correct correspondences were then identified using the exact same method used in the previous experiments (see section 4.3.6.3, pg.192).

#### 4.5.1.2 Performance Metric

For each camera and projection mode (perspective, stereographic and equiangular), the set of all correspondences for all image pairs and reference images were combined into a single set and ordered in descending order of similarity. The threshold on the similarity was then incrementally decreased to obtain the recall versus 1-precision results, where

$$\text{recall} = \frac{\text{number of correct correct correspondences}}{\text{total number of correct correspondences}} \quad (4.130)$$

and

$$1 - \text{precision} = \frac{\text{number false correspondences}}{\text{total number of all correspondences}}. \quad (4.131)$$

The number of correct and false correspondences refers to the number in the subset of correspondences with a similarity score above the threshold.

#### 4.5.1.3 Results

Figures 4.41a and 4.41b show, for each of the projection modes, the recall versus 1-precision results for the parabolic catadioptric and fisheye camera respectively. Ideally, a recall value of 1 would be found for all values of 1-precision.

#### 4.5.1.4 Discussion and Conclusions

Although it was anticipated that the equiangular projection would outperform the perspective and stereographic projections, no significant differences between the results for each projection mode are observed in figure 4.41. This is most likely due to the fact that the majority of keypoint are detected at small scales  $kt$ , which means that the size of their support regions defined by the angle  $\psi_s$  is relatively small. As  $\tan \psi$  is approximately linear for small angles  $\psi$ , the perspective and stereographic projections in equations 4.124 and 4.125 respectively are approximately equivalent to the equiangular projection in equation 4.126. The similarity in the projection modes for small support regions  $\psi_s$  is illustrated in figure 4.42, which shows the appearance of the local region within a keypoint's support region mapped to a fixed sized patch using each projection mode for a range of support region sizes  $\psi_s$ . It is only when the size of the support region  $\psi_s$  is increased that the variations between the appearance of the image content on the patch for each sample mode becomes apparent.

The equiangular projection is used for the remainder of the thesis to find descriptors for sSIFT and pSIFT keypoints. It was argued that it is theoretically more suited than a perspective or stereographic projection, and it is the most efficient of the three projections to compute.

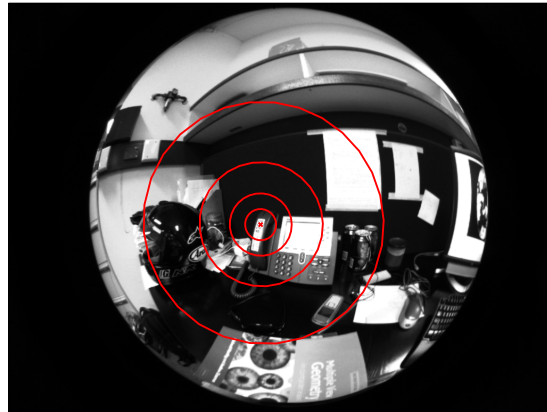
## 4.6 Experiments: Keypoint Detection, Description and Matching

The experiments in this section compare the abilities of SIFT, sSIFT and pSIFT to detect, describe and correctly match corresponding keypoints between successive images in three wide-angle image sequences for a number of different *frames rates*. A frame-rate of 1 finds correspondences between every consecutive image in the sequences, a frame-rate of two finds correspondences between every second second image in the sequence and so on. The ability to reliably find correct correspondences between image pairs is necessary in many vision-based localisation tasks such as visual odometry.

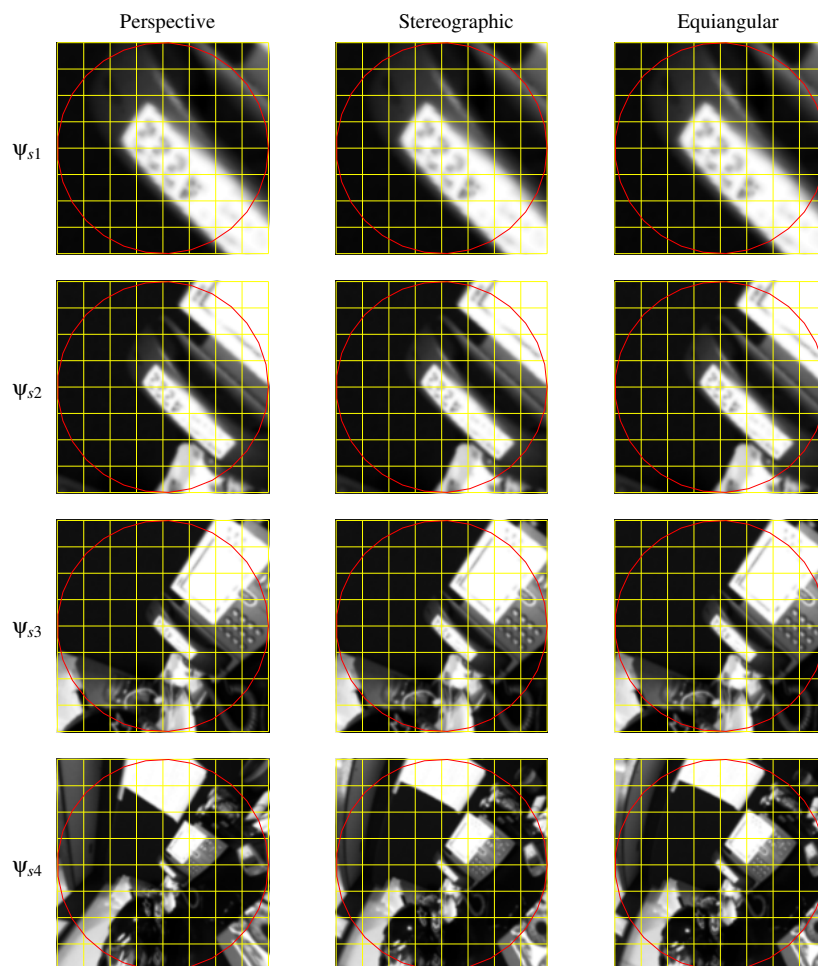
### 4.6.1 Keypoint Types

To summarise, the keypoints types compared are:

1. SIFT (wide-angle): SIFT keypoints detected in the original greyscale wide-angle



(a) A keypoint with support regions  $\Psi_{s1}$ ,  $\Psi_{s2}$ ,  $\Psi_{s3}$  and  $\Psi_{s4}$  of increasing size as they appear on the fisheye image.



(b) The appearance of patches using a perspective (left column), stereographic (middle column) and equiangular (right column) projection.

Figure 4.42: The appearance of the image content (greyscale intensity) within a keypoint's support region when mapped to a fixed sized patch using a perspective, stereographic and equiangular projection.

images using David Lowe's binary code [142]<sup>9</sup>.

2. SIFT (perspective): SIFT keypoints detected in rectified greyscale perspective images using David Lowe's binary code [142]<sup>9</sup>. The method used to convert the wide-angle images to perspective images will be discussed.
3. sSIFT: sSIFT keypoints detected in the first  $n_{oct} = 5$  octaves of scale-space using  $n_{spo} = 3$  scales per octave. For each image sequence, the original wide-angle camera intrinsic parameters are used to select the set of scales  $kt$  as outlined in section 4.3.2. The difference of Gaussian threshold used is 0.01 (images have greyscale values in the range 0-1), and the edge removal threshold used is  $r_{edge} = 10$ . The spectrum  $\hat{I}_{S^2}$  of the greyscale wide-angle image is found using a sample rate  $b = 512$ . For each image sequence, the bandwidth of the camera is estimated using the method described in section 4.6.3. This bandwidth is used to determine if the anti-aliasing interpolation filter is used. The size of the support region for a keypoint is set to  $\psi_s = 10\sqrt{2kt}$ , where  $kt$  is the characteristic scale of a keypoint.
4. pSIFT: pSIFT keypoints detected in the first  $n_{oct} = 5$  octaves of scale-space using  $n_{spo} = 3$  scales per octave. The full octave based approach is used, halving the image size after each octave of scale-space. The difference of Gaussian threshold used is 0.01 (images have greyscale values in the range 0-1), and the edge removal threshold used is  $r_{edge} = 10$ . For each image sequence, the method outlined in section 4.4.1 is used to convert the original wide-angle images to greyscale stereographic images. In previous experiments the input scale  $kt_{input}$  and remaining scales  $kt$  were selected using the original wide-angle camera intrinsic parameters. The experiments in this section find pSIFT keypoints using the scales  $kt_{input}$  and  $kt$  obtained with the stereographic camera intrinsic parameters (the camera model that would produce the stereographic images), and with the original wide-angle camera intrinsic parameters. The support region for a keypoint with characteristic scale  $kt$  is set as  $\psi_s = 10\sqrt{2kt}$ .

## 4.6.2 Producing Perspective Images

For each image sequence, it was necessary to convert each of the wide-angle images to a perspective image to find the SIFT (perspective) keypoints. As the wide-angle cameras used in each image sequence all have in excess of a hemispherical field of view, the wide-angle images in each sequence were converted to perspective images

<sup>9</sup>Available <http://people.cs.ubc.ca/~lowe/keypoints/>

with a diagonal field of view of  $fov = 160^\circ$ . A larger field of view was not used as the perspective images contained severe interpolation artifacts toward the periphery of the image. The perspective images produced were all the same size as the original wide-angle images, having  $nr$  rows of pixels and  $nc$  columns of pixels.

For a diagonal field of view of  $fov = 160^\circ$ , the required distance  $d$  of the perspective image plane from the centre of the views sphere was obtained as

$$d = \frac{\sqrt{nr^2 + nc^2}}{2 \tan(fov/2)}. \quad (4.132)$$

To produce the perspective image, each pixel  $\mathbf{u}$  on the perspective image was projected to the point  $\eta$  on the sphere:

$$\theta = \arctan \left( \frac{\sqrt{(u - u_0)^2 + (v - v_0)^2}}{d} \right), \quad (4.133)$$

$$\phi = \arctan \left( \frac{v - v_0}{u - u_0} \right), \quad (4.134)$$

where  $\mathbf{u}_0(u_0, v_0)$  is the position of the principal point. The position of the principal point used for each image sequence will be specified in the next section. This point  $\eta$  was then projected to a pixel position  $\mathbf{u}'$  on the original wide-angle image, and a linear interpolation was used to sample the greyscale intensity value on the wide-angle image at the position  $\mathbf{u}'$ .

### 4.6.3 Wide-Angle Image Sequences

#### 4.6.3.1 Fisheye: Fisheye Camera

The fisheye sequence includes 1100 colour images of size  $1024 \times 768$  pixels captured by a Point Grey Research firewire camera and OmniTech Robotics fisheye lens during an approximately 2km walking tour of an outdoor industrial site (Queensland Centre for Advanced Technologies — QCAT). Figure 4.43a shows an aerial view of the QCAT site. The camera path is not shown as no ground truth data was obtained.

**Calibration and Image Transformations:** The fisheye camera is the same camera calibrated in chapter 2. It is modelled using the unified image model with equal scaling in the  $u, v$  coordinates and zero shear. The camera intrinsic parameters can be found in table 2.6 (pg.86).

Each colour image was converted to a greyscale image  $I$  using the transform  $I = (0.3R + 0.59G + 0.11B)/255$ , where  $R, G$  and  $B$  are the red, green and blue colour channels respectively with values in the range 0 to 255. The greyscale image has values in the range 0 to 1. The stereographic images used by pSIFT were obtained using the method described in section 4.4.1 (pg. 206). The perspective images were produced using the method just described in section 4.6.2, where the position of the principal point on the perspective images was set to  $\mathbf{u}_0 = (nc/2, nr/2)$ , where  $nr = 768$  and  $nc = 1024$  are the number of rows and columns of pixels in the image respectively. Figure 4.43b shows an example of three of the consecutive greyscale fisheye images in the sequence. These same images converted to stereographic images and perspective images are shown in figures 4.43c and 4.43d respectively. The green region in the images is the image mask. Any keypoints detected in the green regions were removed.

**Image Bandwidth and anti-aliasing interpolation filter:** The bandwidth of an image captured by the fisheye camera was estimated previously in section 4.3.1.2, and is shown in figure 4.6a (pg.172). The anti-aliasing filter was used to find the sSIFT keypoints as the maximum image bandwidth exceeds the maximum computationally feasible sample rate  $b = 512$  (i.e. sSIFT (512\*) keypoints were found).

#### 4.6.3.2 Hyperion: Equiangular Catadioptric Camera

The Hyperion sequence consists of 2000 colour images of size  $640 \times 480$  pixels taken by an equiangular catadioptric camera mounted on the mobile robot Hyperion. The images were taken as the mobile robot traversed through the Atacama desert [48]<sup>10</sup>. Figure 4.44a shows the  $x, y$  coordinates of the vehicle path obtained from GPS. Although no GPS data is available in the  $z$  direction, the vehicle path is known to be approximately planar.

**Calibration and Image Transformations:** The equiangular catadioptric camera model is illustrated in figure 2.11 (pg.43) and given in equation 2.11 (pg.43). To recap, a pixel  $\mathbf{u}$  on the image at position  $\mathbf{x}(r, \zeta) = \mathbf{u} - \mathbf{u}_0$  relative to the principal point, where  $r$  and  $\zeta$  are the polar coordinates of  $\mathbf{x}$ , is projected to a point  $\eta(\theta, \phi)$  on the sphere by

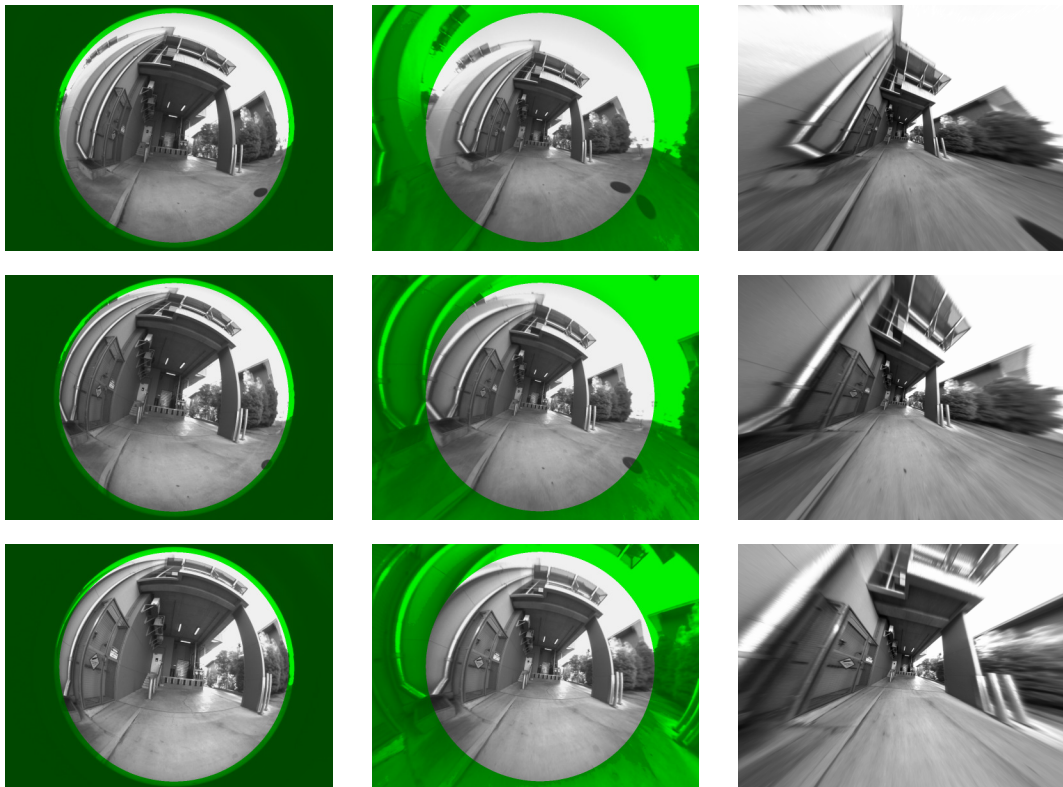
$$\eta(\theta, \phi) = \begin{bmatrix} \sin(\rho \tan^{-1}(R/f)) \cos(\zeta) \\ \sin(\rho \tan^{-1}(R/f)) \sin(\zeta) \\ \cos(\rho \tan^{-1}(R/f)) \end{bmatrix}, \quad (4.135)$$

<sup>10</sup>This data set has been provided courtesy of Carnegie Mellon University and Peter Corke





(a) Aerial view of Queensland Centre for Advanced Technologies (QCAT). The scale in the upper right corner is distance in metres.



(b) Greyscale fisheye images ( $1024 \times 768$  pixels).

(c) Greyscale stereographic images ( $1024 \times 768$  pixels).

(d) Greyscale perspective images ( $1024 \times 768$  pixels).

Figure 4.43: The fisheye image sequence includes 1100 images obtained by a fisheye camera on an approximately 2 kilometre outdoor walking tour of an outdoor industrial environment (QCAT). The red dashed lines in 4.43b show the field of view of the perspective images in figure 4.43d. The green highlighted regions are the image mask.

where  $\rho$  and  $f$  are the camera intrinsic parameters. The camera intrinsic parameters were obtained from [48] and are

$$\mathbf{u}_0 = \begin{pmatrix} 310.80 \\ 241.22 \end{pmatrix} \quad (4.136)$$

$$f = 1433.15 \quad (4.137)$$

$$\rho = 10.60. \quad (4.138)$$

As was the case for the fisheye sequence, the transform  $I = (0.3R + 0.59G + 0.11B)/255$  was used to convert each colour image to a greyscale image  $I$  with values in the range 0-1. Each image was then converted to greyscale stereographic image using the method described in section 4.4.1 (pg. 206), and to a greyscale perspective image using the method described in section 4.132. The position of the principal point in the perspective images was set to  $\mathbf{u}_0 = (nc/2, 50 + nr/2)^T$ , where  $nr = 480$  is the number of rows in the image and  $nc = 640$  is the number of columns in the image. The additional 50 pixel offset was used to increase the region in the image where valid keypoints could be detected (i.e. regions where the robot is not imaged). Three of the consecutive greyscale equiangular catadioptric images in the sequence are shown in figure 4.44b. Figures 4.44c and 4.44d show these same images converted to the stereographic and perspective images. The green highlighted regions represent the image mask. Any keypoints detected in green highlighted regions were removed.

**Image Bandwidth:** An estimate of the equiangular catadioptric image bandwidth was derived algebraically using the general procedure outlined in section 4.3.1.2 (pg. 167). Recall that the change in angle  $d\psi$  on the sphere can be parameterised by the change in polar coordinates on the sphere by  $d\psi^2 = d\theta^2 + \sin^2 \theta d\phi^2$ . The variables  $d\theta^2$ ,  $\sin^2 \theta$  and  $d\phi^2$  were derived for the equiangular camera model, and are

$$d\phi^2 = d\zeta^2, \quad (4.139)$$

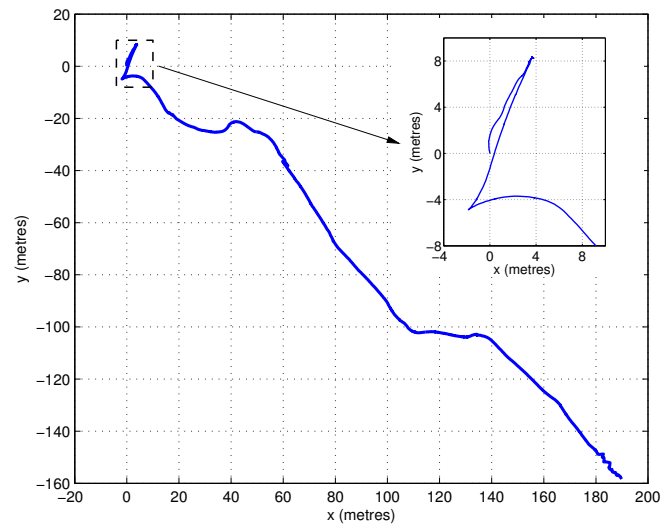
$$\sin^2(\theta) = \sin^2 \left[ \rho \tan^{-1} \left( \frac{r}{f} \right) \right], \quad (4.140)$$

and

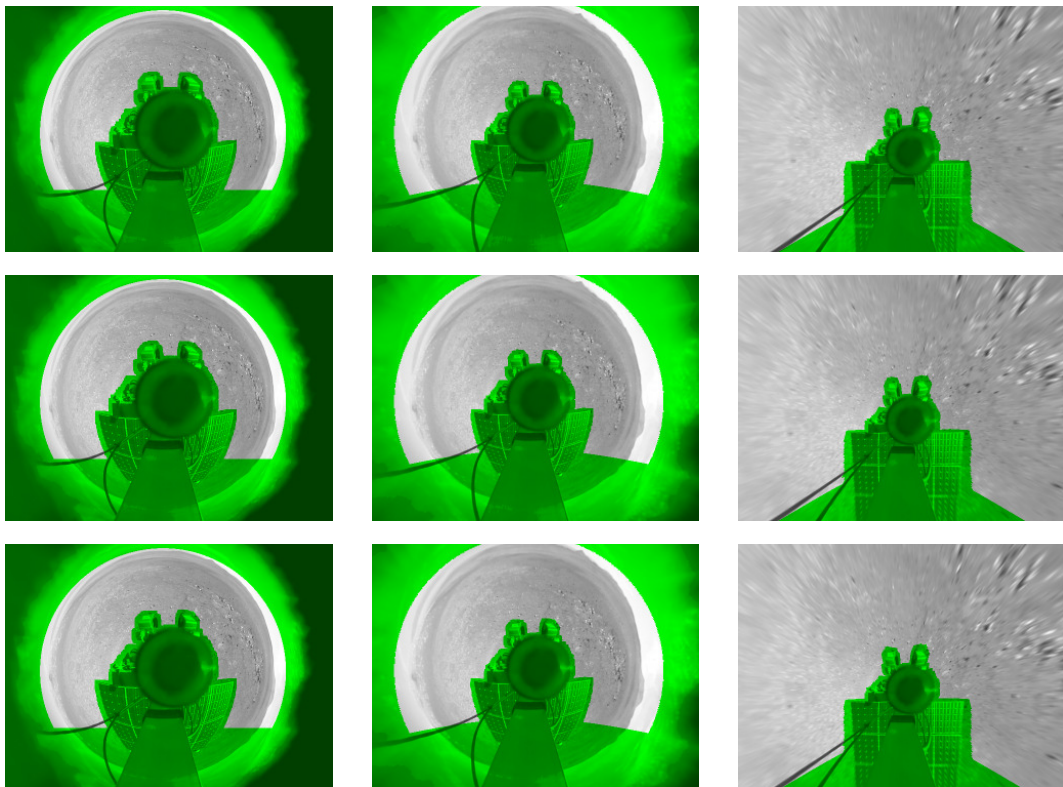
$$d\theta^2 = \frac{\rho^2 f^2}{(f^2 + r^2)^2} dr^2, \quad (4.141)$$

where  $r, \zeta$  are the polar coordinates of the point  $\mathbf{x}(r, \zeta) = \mathbf{u} - \mathbf{u}_0$  on the image. From the equations which relate a small shift  $dP(\alpha)$  to the polar coordinates  $(r, \zeta)$  on the image plane given in equations 4.49 and 4.50, the following estimate of the image bandwidth





(a)  $x,y$  coordinates of the vehicle path obtained from GPS (non-differential) measurements.



(b) Greyscale equiangular catadioptric images ( $640 \times 480$  pixels).

(c) Greyscale stereographic images ( $640 \times 480$  pixels).

(d) Greyscale perspective images ( $640 \times 480$  pixels).

Figure 4.44: The Hyperion image sequence includes 2000 images captured by an equiangular catadioptric camera mounted on the mobile robot Hyperion. The images were taken as the robot traversed through the Atacama desert. The red dashed lines in 4.44b show the field of view of the perspective images in 4.44d. The green highlighted regions are the image mask.

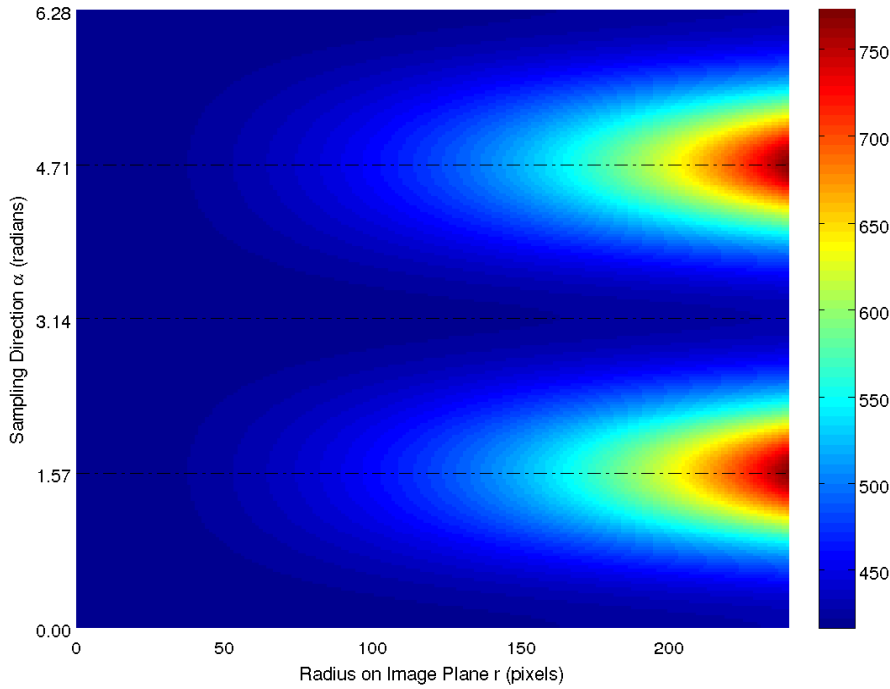


Figure 4.45: Estimated bandwidth of the images captured by the equiangular catadioptric camera (Hyperion sequence).

was obtained:

$$b_I(r, \alpha) = \pi / \frac{d\Psi}{dP}(r, \alpha) \quad (4.142)$$

$$= \pi / \left[ \frac{\rho^2 f^2}{(f^2 + r^2)^2} dP^2 \cos^2(\alpha) + \sin^2 \left( \rho \tan^{-1} \left( \frac{r}{f} \right) \right) \frac{dP^2 \sin^2(\alpha)}{r^2} \right]^{1/2}. \quad (4.143)$$

The estimate of the bandwidth is given in figure 4.45 for radii  $r$  extending to the limit of the camera's field of view (the camera's field of view extends to an angle of colatitude of approximately  $100^\circ$  for all angles of longitude). No anti-aliasing interpolation filter was used by sSIFT as the estimated bandwidth only exceeds the sample rate  $b = 512$  towards the edge of the camera's field of view (i.e. sSIFT (512) keypoints were found).

### 4.6.3.3 Tractor: Equiangular Catadioptric Camera

The Tractor sequence consists of several thousand greyscale images of size  $1024 \times 768$  pixels taken by an equiangular catadioptric camera mounted on a mobile robotic tractor<sup>11</sup>. The images were taken as the mobile robot moved outdoors through a university

<sup>11</sup>The 'Tractor' image sequence was provided courtesy of Kane Usher and Jonathan Roberts.

campus. The approximate path of the robot is illustrated in figure 4.46a. The first 1000 images in the sequence were used in these experiments.

**Calibration and Image Transformations:** The equiangular camera model is the same as that of the equiangular catadioptric camera in the Hyperion sequence. The camera intrinsic parameters were obtained from the dissertation of Usher [232], and are

$$\mathbf{u}_0 = \begin{pmatrix} 519.01 \\ 371.01 \end{pmatrix} \quad (4.144)$$

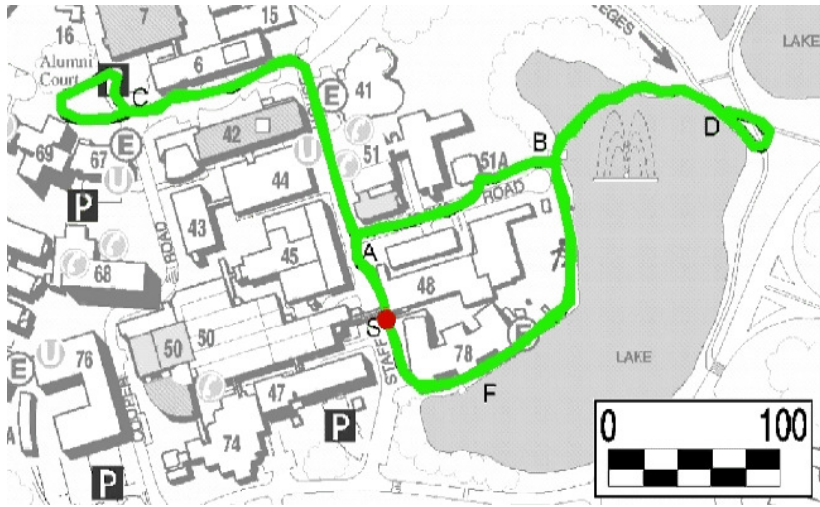
$$f = 1411.02 \quad (4.145)$$

$$\rho = 7.142. \quad (4.146)$$

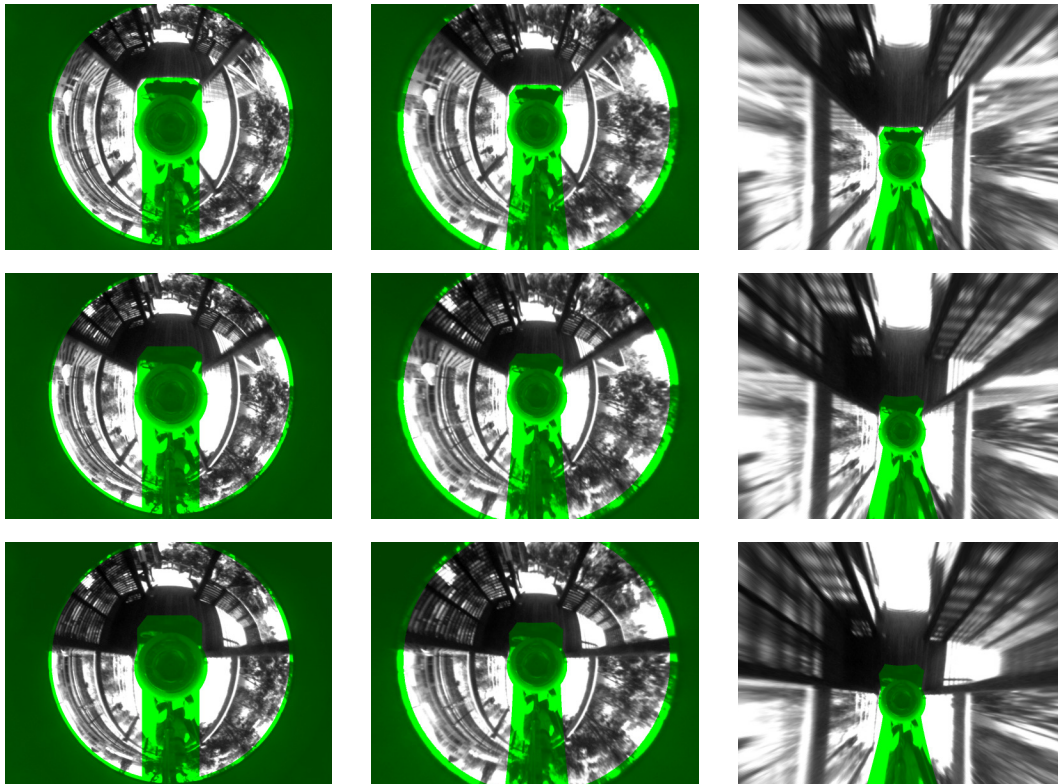
Each equiangular catadioptric image was converted to a stereographic and perspective image using the methods described in sections 4.4.1 (pg. 206) and 4.132 respectively. The position of the principal point in the perspective images was set to  $\mathbf{u}_0 = (nc/2, 100 + nr/2)^T$ , where  $nr = 768$  and  $nc = 1024$ . As was the case for the hyperion sequence, an additional offset was added to the coordinate  $v_0$  to increase the region in the image in which valid keypoints could be detected (i.e. regions where the robot was not imaged). Figure 4.46b shows three of the original equiangular catadioptric images in the sequence. These same images converted to stereographic and perspective images are shown in figures 4.46c and 4.46d respectively. Again, the green region is the image mask. A keypoint was not retained if it was detected in the green region.

### Image Bandwidth

As the camera model is the same as the equiangular catadioptric camera used in the Hyperion sequence, an estimate of the equiangular catadioptric image bandwidth was obtained from equation 4.142. This estimated bandwidth is shown in figure 4.47 for all radii  $r$  on the image within the camera's field of view (the camera's field of view extends to an angle of colatitude of approximately  $110^\circ$  for all angles of longitude). The anti-aliasing interpolation filter was used by sSIFT as the estimated bandwidth exceeds the maximum computationally feasible sample rate of  $b = 512$  for all radii  $r$  and sample directions  $\alpha$  (i.e. sSIFT (512\*) keypoints were found).



(a) Approximate path of the vehicle through university campus (University of Queensland, Australia). The scale is distance in metres.



(b) Greyscale equiangular catadioptric images ( $1024 \times 768$  pixels).

(c) Greyscale stereographic images ( $1024 \times 768$  pixels).

(d) Greyscale perspective images ( $1024 \times 768$  pixels).

Figure 4.46: The tractor image sequence includes several thousand images obtained by a equiangular catadioptric camera mounted on a mobile robotic tractor. The images were taken as the robot moved outdoors through a university campus. The red dashed lines in 4.46b show the field of view of the perspective images in 4.46d. The green highlighted regions are the image mask. The first 1000 images in the sequence were used in the experiments.

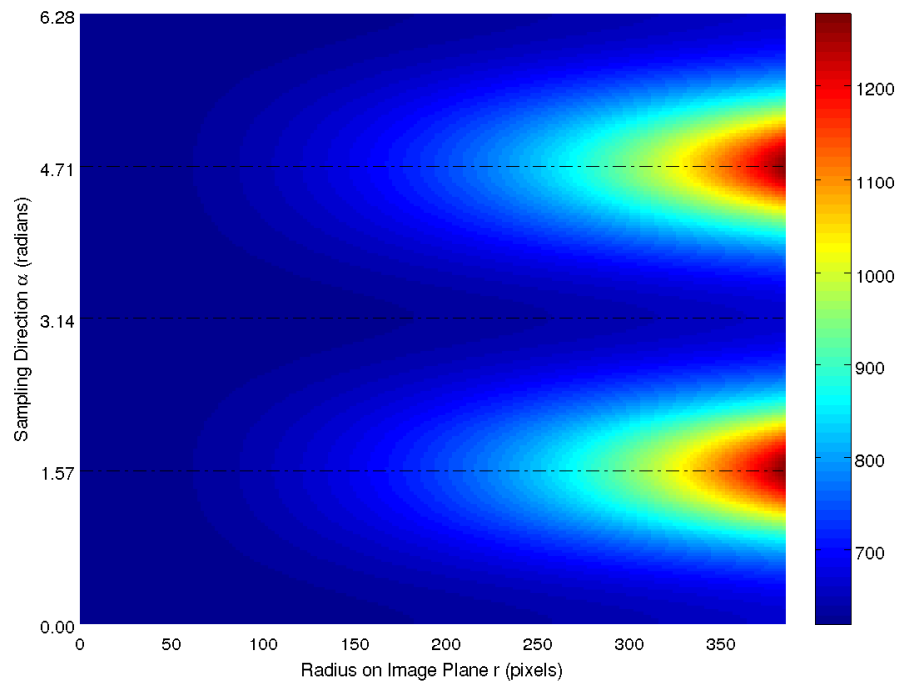


Figure 4.47: Estimated bandwidth of the images captured by the equiangular catadioptric camera (Tractor sequence).

#### 4.6.4 Performance metrics

The performance of a keypoint type is measured using recall versus 1-precision statistics and the mean number of correct keypoint correspondences found between successive image pairs in the sequence for a given frame-rate. For each keypoint type, frame-rate and image sequence, keypoints are matched between the image pairs using the keypoint descriptors. Four different methods used to match the keypoints and assign a similarity to each of the correspondences was used, and these will be discussed in section 4.6.5. The correct correspondences were then identified using epipolar constraints between the image pairs. This process will be described in section 4.6.6. The set of all correspondences for all image pairs were then combined into a single set of correspondences and ranked in descending order of similarity. By incrementally decreasing the similarity threshold, the recall versus 1-precision values were obtained, where recall and precision were defined precisely in equations 4.130 and 4.131. Recall versus 1-precision is the same metric used in other similar studies used to compare keypoint detectors and descriptors [120, 203], and is a more appropriate metric to use in these experiments than receiver operating characteristics (ROC) as that the exact number of false correspondences between image pairs cannot be determined exactly. This is the case as only an estimate of the epipolar geometry between views is used to



distinguish between correct and incorrect correspondences.

### 4.6.5 Similarity Metrics

Four different methods were used to match the correspondences and assign a similarity score. To simplify the discussion, for a given pair of images, let  $k'$  be the set of all keypoints in the image with the fewest number of keypoints, and let  $k''$  be the set of keypoints in the image with the largest number of keypoints. The notation  $d(k'_i, k''_j)$  is defined to mean the Euclidean distance between the SIFT descriptors of keypoints  $k'_i$  and  $k''_j$ .

**1. Euclidean distance (L2 norm) and 2. Ambiguity:** For each keypoint  $k'_i$ , find the Euclidean distances  $d(k'_i, k''_j)$  between  $k'_i$  and all keypoints  $k''_j$ . The corresponding keypoint  $k''_j$  is the one having the smallest Euclidean distance. The similarity score for this corresponding pair of keypoints is assigned using two metrics. The first similarity score is the Euclidean distance  $d(k'_i, k''_j)$  (1. Euclidean distance). The second similarity score is the ambiguity ratio, which is the ratio of  $d(k'_i, k''_j)$  to the second smallest Euclidean distance  $d(k'_i, k''_l)$  (2. Ambiguity).

**3. Mutual Euclidean distance (mutual L2 norm) and 4. Mutual Ambiguity:** For each keypoint  $k'_i$ , find the Euclidean distances  $d(k'_i, k''_j)$  between  $k'_i$  and all keypoints  $k''_j$ . A corresponding keypoint  $k''_j$  is found only if the Euclidean distance  $d(k'_i, k''_j) \leq d(k'_i, k''_l)$ , and  $d(k'_i, k''_j) \leq d(k'_l, k''_j)$ . This is a ‘mutual consistency’ check and ensures that a keypoint can only ever correspond to one other keypoint in the other image. If a corresponding pair of keypoints is found, the similarity of the corresponding pair is assigned again using two metrics. The first similarity score is the Euclidean distance  $d(k'_i, k''_j)$  (3. Mutual Euclidean distance). The second similarity score is the ambiguity ratio, which is the ratio of  $d(k'_i, k''_j)$  to the second smallest Euclidean distance  $d(k'_i, k''_l)$  (4. Mutual Ambiguity).

### 4.6.6 Selecting Correct Correspondences

Correct correspondences between image pairs were identified using epipolar constraints and the assumption that scene points remain rigid between views. For each image pair, the set of all calibrated keypoint correspondences for each keypoint type having a mutual ambiguity score above the empirically selected value of 0.85 were combined into

a single set of correspondences. The essential matrix  $E$  was then estimated using the five-point algorithm of Nistér [212] and RANSAC [70] to remove outliers. Note that more recently a simplified implementation of Nistér’s original algorithm was developed by Li and Hartley [129]. A pair of corresponding keypoints with coordinates  $\eta$  and  $\eta'$  in image 1 and 2 respectively were deemed correct if  $|\eta' E \eta^T| < 0.005$ . This threshold was selected empirically. Using the combined set of correspondences for all keypoint types enabled an accurate estimate of the essential matrix  $E$  to be obtained. Furthermore, it ensured that the same estimate of the epipolar geometry between views was used to identify the correct keypoint correspondences.

### 4.6.7 Results

Results were obtained for frame-rates 1 and 2 for the fisheye and tractor sequences, and for frame-rates 1,2 and 4 for the Hyperion sequence. The magnitude of the camera translation between consecutive images in the Hyperion sequence was in general less than that for the fisheye and Tractor sequences.

The recall versus 1-precision results are shown for the fisheye sequence in figures 4.48 and 4.49 for frame-rates 1 and 2, for the Hyperion sequence in figures 4.50, 4.51 and 4.52 for frames rates 1,2 and 4, and for the Tractor sequence in figures 4.53 and 4.54 for frame-rates 1 and 2. The mean number of correct keypoint correspondences between image pairs for each sequence, keypoint type and frame-rate are given in table 4.6.

### 4.6.8 Discussion and Conclusions

#### Fisheye

The recall versus 1-precision results show that for each frame-rate and keypoint type, when compared to the results using the L2 similarity metrics, improved performance is found using the ambiguity metrics. These results validate to claim made by Lowe [142] that the ambiguity of two SIFT keypoint descriptors is a more robust means for assigning the similarity score between correspondences than the L2 norm (Euclidean distance) between SIFT descriptors.

Overall sSIFT shows the best recall versus 1-precision results for each similarity metric and frame-rate followed closely by pSIFT (parabolic scales). As with sSIFT and pSIFT (parabolic scales), the recall versus 1-precision results show that overall pSIFT (fisheye scales) outperforms both SIFT (wide-angle) and SIFT (perspective) for

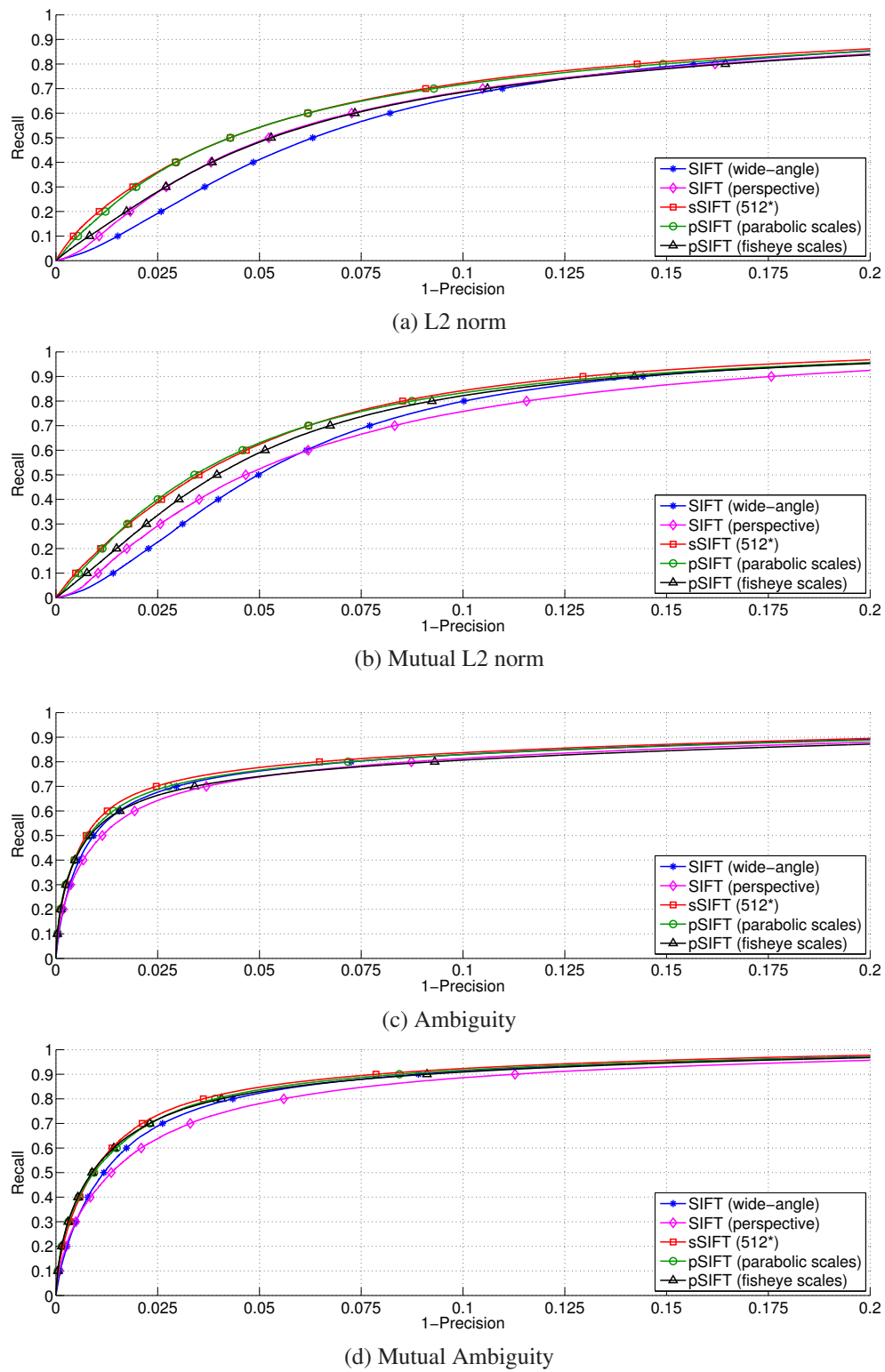
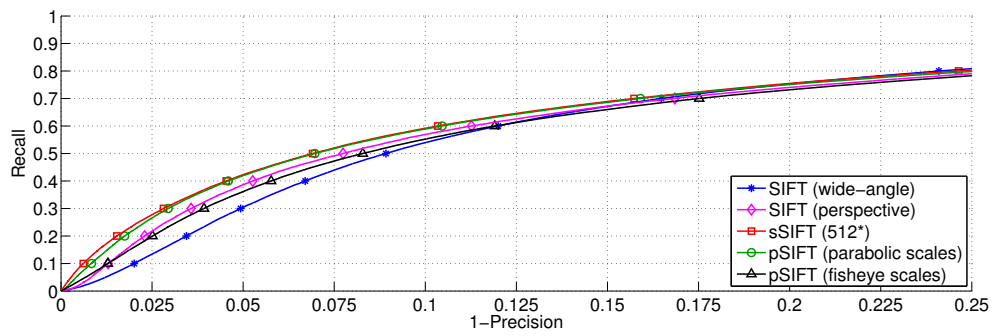
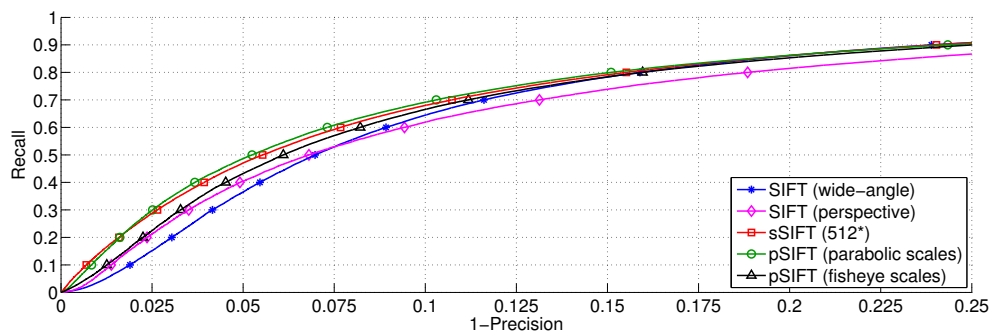


Figure 4.48: Recall vs. 1-precision results for the fisheye sequence (frame-rate 1).

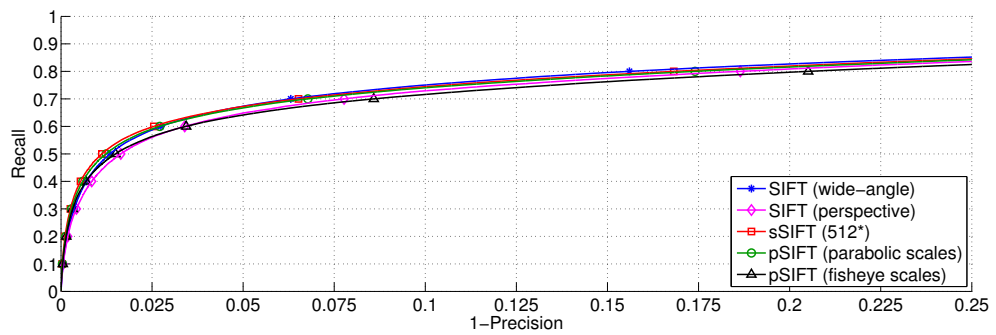




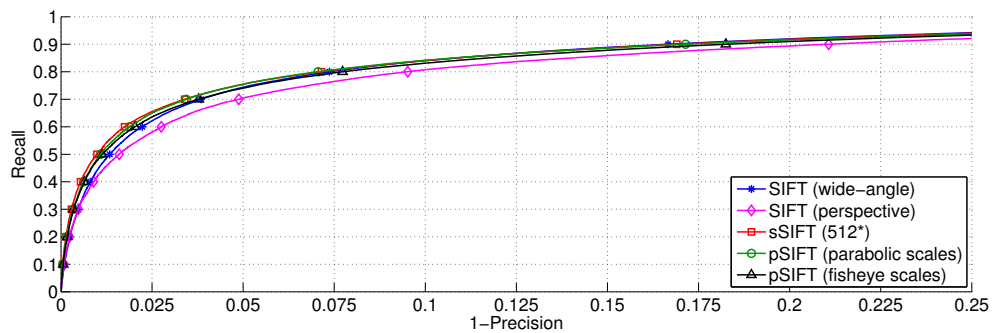
(a) L2 norm



(b) Mutual L2 norm



(c) Ambiguity



(d) Mutual Ambiguity

Figure 4.49: Recall vs. 1-precision results for the fisheye sequence (frame-rate 2).

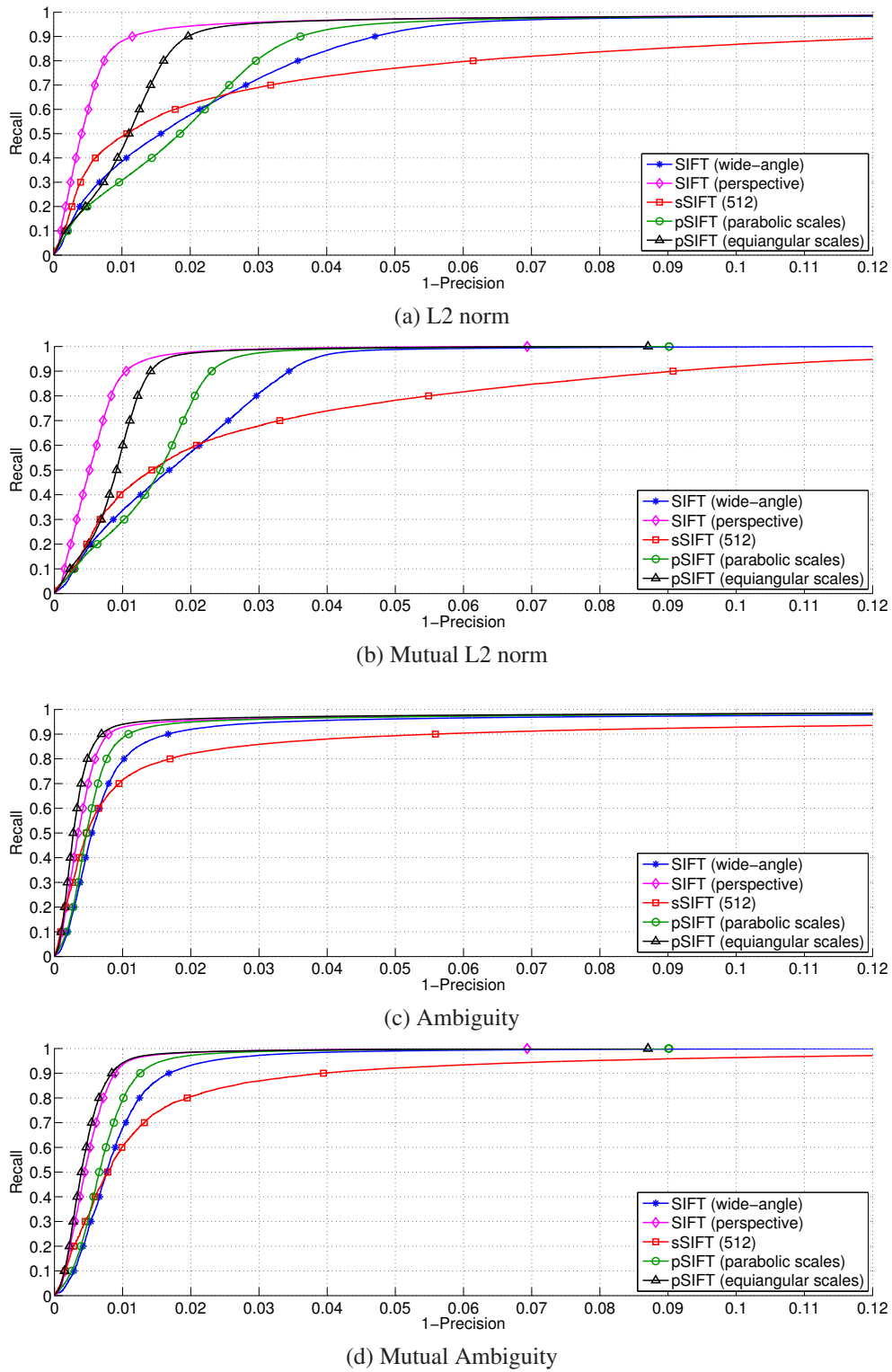


Figure 4.50: Recall vs. 1-precision results for the Hyperion sequence (frame-rate 1).

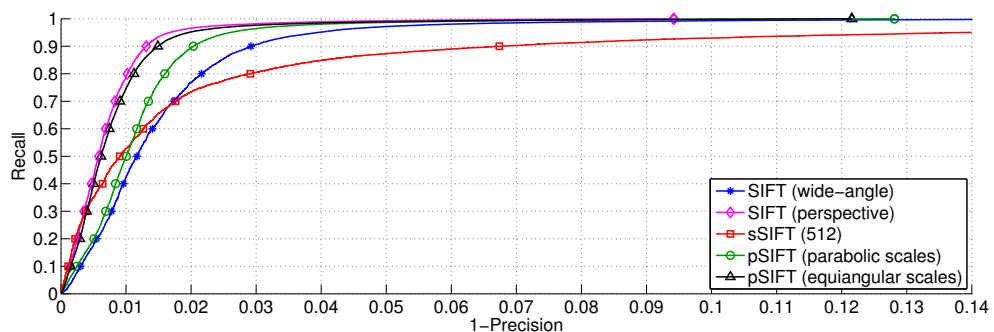
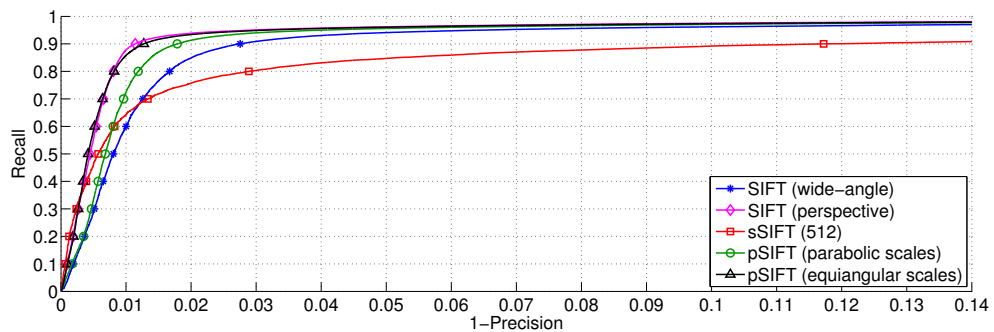
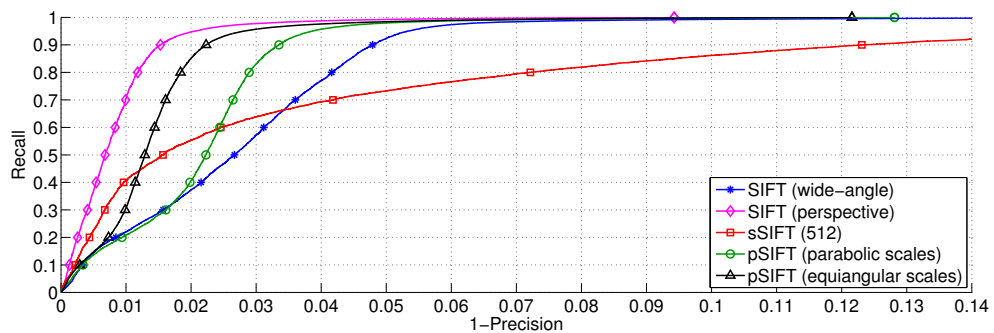
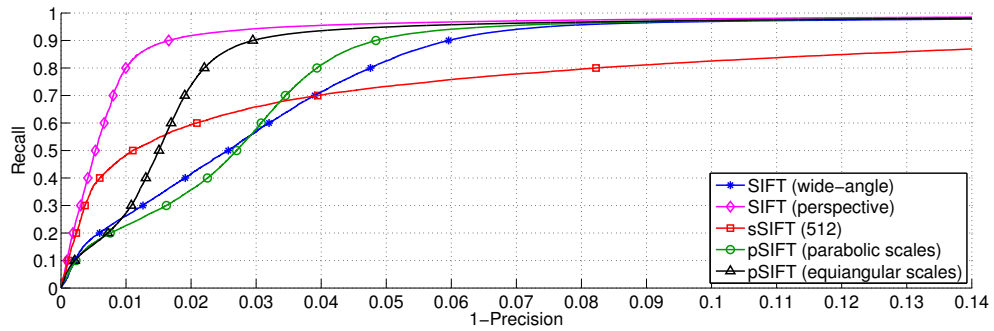


Figure 4.51: Recall vs. 1-precision results for the Hyperion sequence (frame-rate 2).

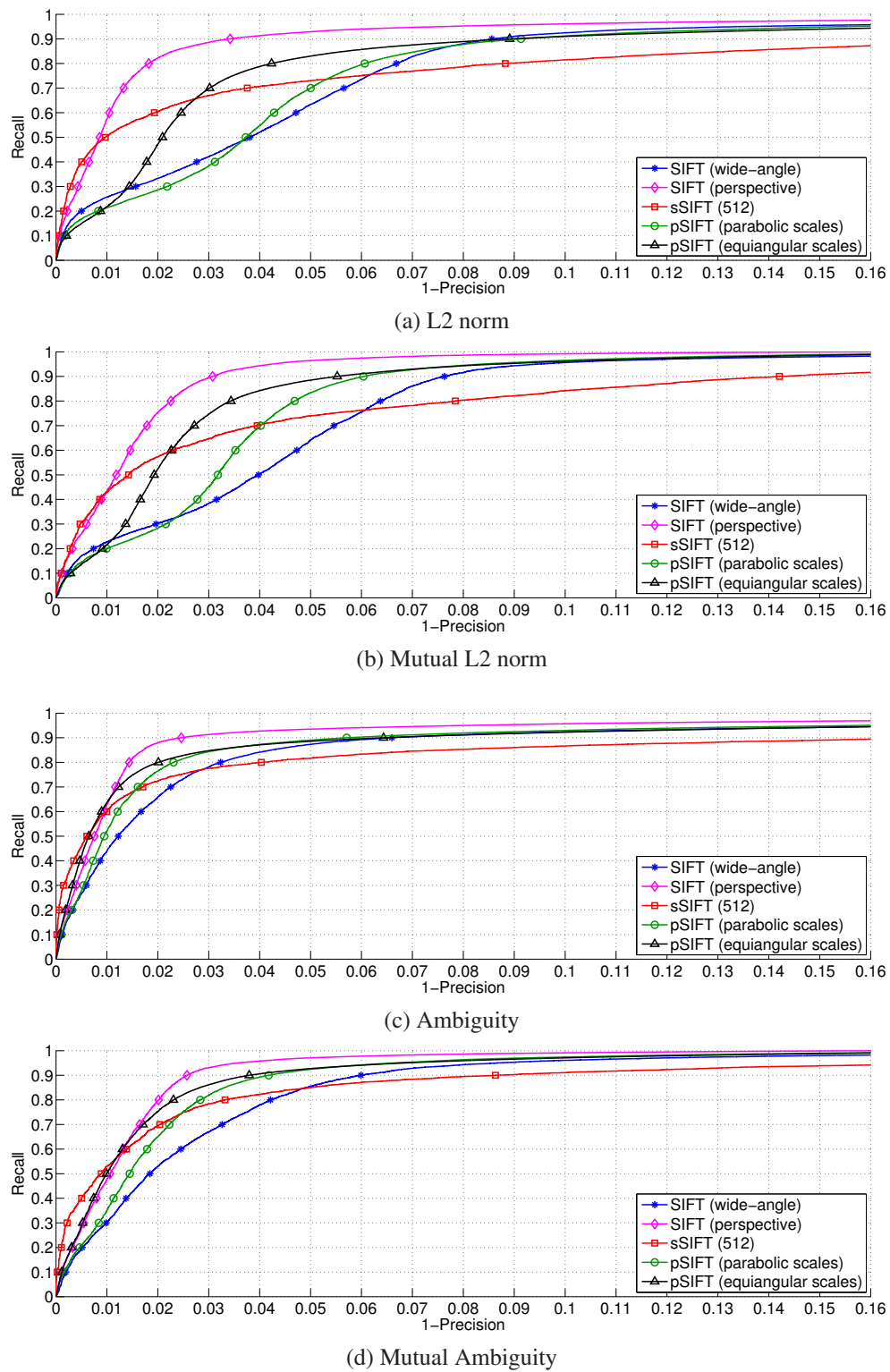
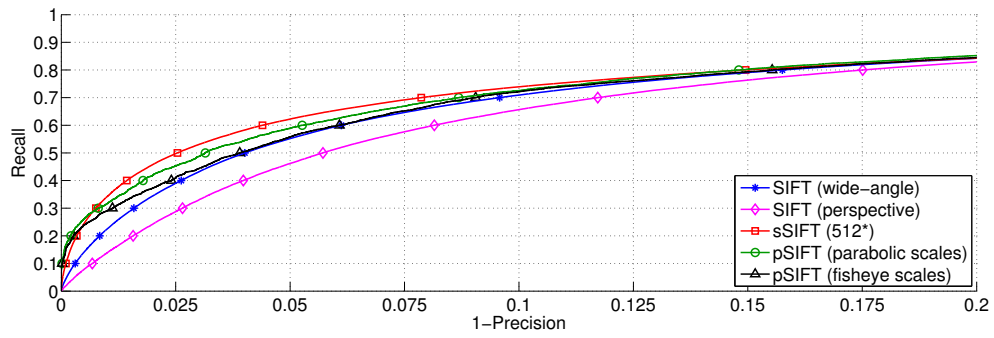
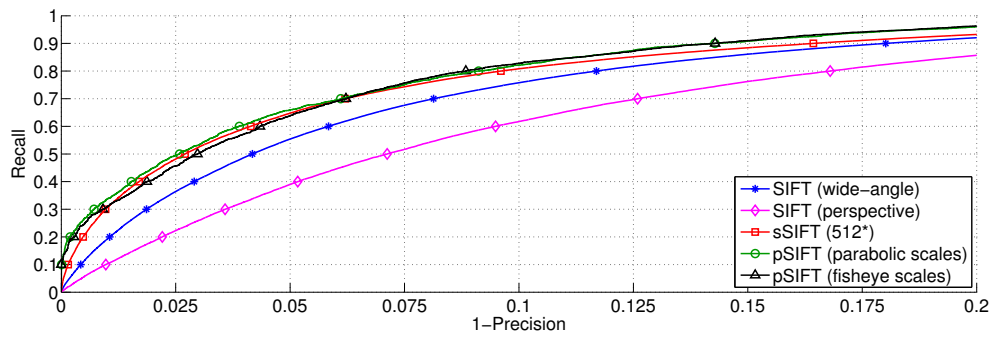


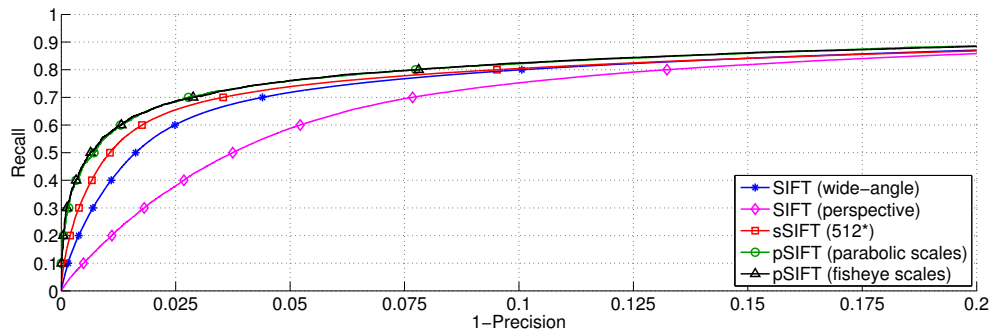
Figure 4.52: Recall vs. 1-precision results for the Hyperion sequence (frame-rate 4).



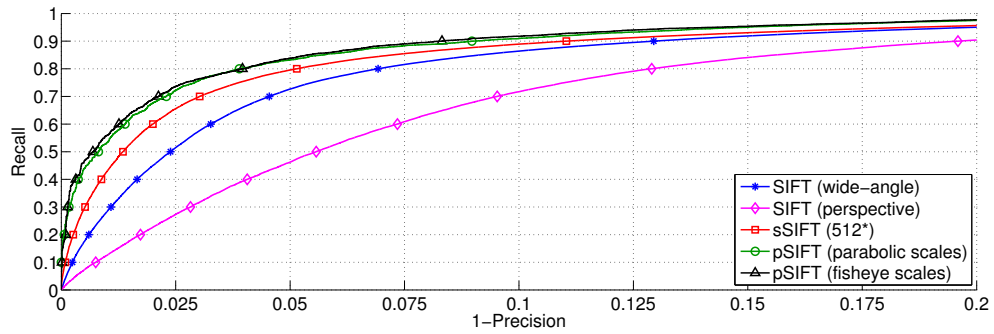
(a) L2 norm



(b) Mutual L2 norm



(c) Ambiguity



(d) Mutual Ambiguity

Figure 4.53: Recall vs. 1-precision results for the tractor sequence (frame-rate 1).

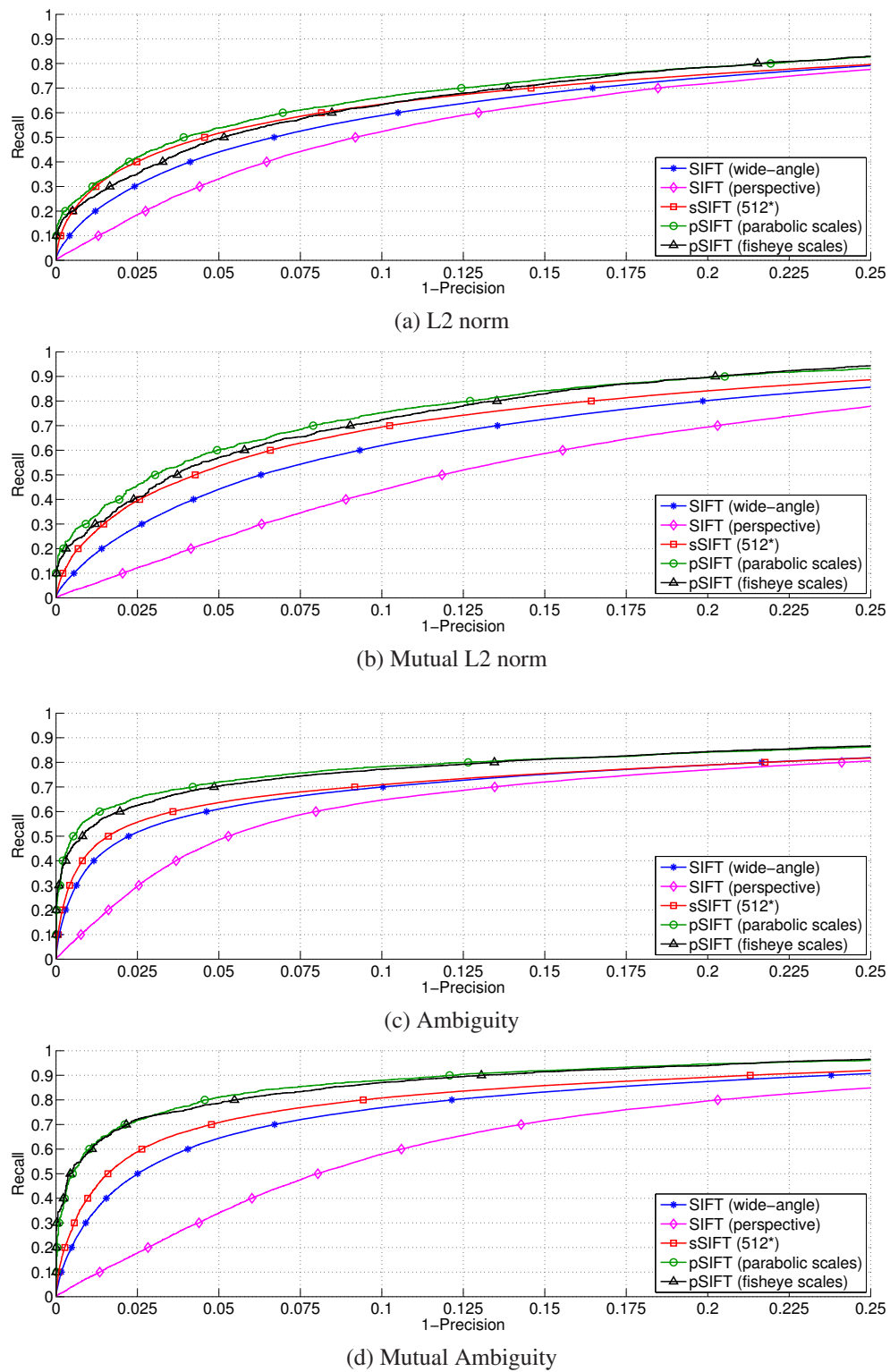


Figure 4.54: Recall vs. 1-precision results for the tractor sequence (frame-rate 2).

Image Sequence	Keypoint	Frame-rate = 1		Frame-rate = 2		Frame-rate = 4	
		L2 & Ambiguity	Mutual L2 & Mutual Ambiguity	L2 & Ambiguity	Mutual L2 & Mutual Ambiguity	L2 & Ambiguity	Mutual L2 & Mutual Ambiguity
Fisheye	SIFT (wide-angle)	500.7	431.6	338.6	276.7	-	-
	SIFT (perspective)	184.2	158.6	119.5	97.5	-	-
	sSIFT (512*)	572.6	498.4	362.6	294.6	-	-
	pSIFT (parabolic scales)	498.2	418.0	316.4	247.5	-	-
	pSIFT (fisheye scales)	606.4	493.8	393.4	298.0	-	-
Hyperion	SIFT (wide-angle)	108.1	104.1	94.1	89.9	70.5	66.1
	SIFT (perspective)	181.5	176.4	168.8	163.5	142.0	136.3
	sSIFT (512)	39.9	37.6	30.9	28.5	23.0	20.7
	pSIFT (parabolic scales)	249.6	241.4	225.7	216.8	183.3	172.7
	pSIFT (equiangular scale)	505.2	492.1	460.9	446.2	368.0	347.4
Tractor	SIFT (wide-angle)	477.5	412.7	320.1	259.3	-	-
	SIFT (perspective)	101.3	88.7	61.8	51.0	-	-
	sSIFT (512*)	260.1	220.5	184.1	145.5	-	-
	pSIFT (parabolic scales)	324.7	277.3	221.8	168.0	-	-
	pSIFT (equiangular scales)	307.0	261.2	202.4	158.6	-	-

Table 4.6: Mean number of correct correspondences between image pairs for each keypoint type, frame-rate and similarity metric.



all frame-rates and similarity metrics. The recall versus 1-precision results for SIFT (perspective) and better than SIFT (wide-angle) using the L2 similarity metrics for each frame-rate, however, the opposite is true for the ambiguity similarity metrics.

The mean number of correct correspondences found for each keypoint type reduces as the frame-rate increases. There is also a small reduction in this number when the mutual consistency constraint is used for both the L2 and ambiguity similarity metrics. Overall, pSIFT (fisheye scales) finds the greatest number of correct correspondences for each similarity metric and frame-rate, followed in descending order by sSIFT, SIFT (wide-angle), pSIFT (parabolic scales) and SIFT (perspective). SIFT (perspective) performs poorly in comparison to all other keypoint types as it finds less than half as many average correct correspondences than all other keypoint types.

The results obtained suggest that SIFT (wide-angle) outperforms SIFT (perspective) for the fisheye sequence. Using the ambiguity similarity metrics, which overall give improved performance compared to the L2 metrics, SIFT (wide-angle) shows improved recall versus 1-precision results and finds a greater number of mean correct correspondences compared to SIFT (perspective). SIFT (wide-angle) also has the added advantage that it can detect keypoints in the full field of view of the camera.

It is concluded from the results obtained that sSIFT, pSIFT (parabolic scales) and pSIFT (fisheye scales) are more suited for keypoint detection and matching in the fish-eye sequence than both SIFT (wide-angle) and SIFT (perspective). However, it is interesting to observe that SIFT (wide-angle) still produced reasonable results, particularly when the ambiguity metrics were used. One explanation for this can be made with reference to figure 4.43b. The change in appearance of the imaged scene between views is affected by both the change in pose and the radial distortion of the camera. It is only when a region in the scene changes position between two images that the *change* in appearance resulting from the camera's radial distortion becomes apparent. Regions in the scene near the centre of the first image in figure 4.43b for example do not change position significantly in the other images — the camera motion is predominantly forward translation, so these regions are near the focus of expansion. Therefore, the change in appearance caused by the camera's radial distortion is minimal. However, for large camera rotation the effect of the camera's radial distortion on the appearance of the scene becomes more apparent. Figure 4.55 for example shows the 50 most similar SIFT and sSIFT keypoint correspondences found between two images separated by a large change in camera rotation using the ambiguity metric. As sSIFT is designed to be invariant to a camera's radial distortion and shift invariant to camera rotation, sSIFT finds a greater number of correct correspondences than SIFT

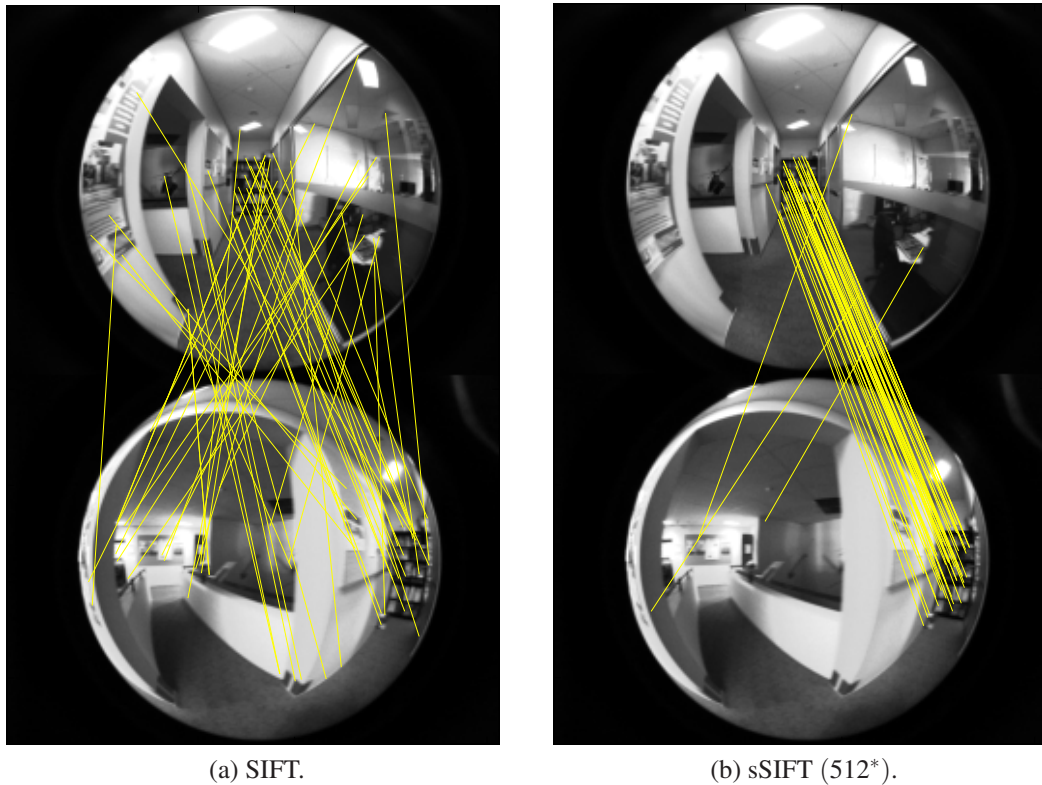


Figure 4.55: The 50 most similar (a) SIFT (wide-angle) and (b) sSIFT(512\*) keypoint correspondences found between two fisheye images separated by a large change in camera pose. The ambiguity metric was used to find the keypoint matches.

(wide-angle).

### Hyperion

As was the case for the fisheye sequence, when compared to the results found using the L2 metrics, improvements in the recall versus 1-precision results for each frame-rate and keypoint type were observed using the ambiguity metrics.

In contrast to the results for the fisheye sequence, in general the recall versus 1-precision results for SIFT (perspective) are better than all other keypoint types for each frame-rate and similarity metric. The exception is the results for frame-rate 1 using the ambiguity metrics where pSIFT (equiangular scales) outperformed all other keypoint types. This result can be explained by the fact that the mobile robot Hyperion traversed over a near planar surface, and that the camera's principal axis was approximately orthogonal to this plane. Successive images in the sequence are therefore related by an approximate Euclidean transform (shift and rotation of the image function). As SIFT is designed to detect keypoints and produce a SIFT descriptor for each keypoint in a manner invariant to these changes, SIFT (perspective) is well suited for the Hyperion

sequence. However, the disadvantage of SIFT (perspective) is its inability to detect keypoints in an image spanning the full field of view of the camera.

The recall versus 1-precision results show that both pSIFT (parabolic scales) and pSIFT (equiangular scales) outperform SIFT (wide-angle) for all similarity metrics and frame-rates, with pSIFT (parabolic scales) giving slightly better results than pSIFT (equiangular scales). sSIFT performs well for small values of 1-precision, however, the results deteriorate more quickly than the other keypoint types. sSIFT also finds significantly fewer correct keypoint correspondences than the other keypoint types for all frame-rates and similarity metrics. One possible explanation for this result is the fact that the equiangular sequence contains predominantly fine detailed structure (rocks, pebbles etc.). To implement sSIFT, the image needs to be sampled on an equiangular  $\theta, \phi$  image to obtain the spectrum  $\hat{I}_{S^2}$  of the image. Once the set of scale-space images  $L_{S^2}(\cdot; kt)$  are found, they are then mapped back to the set of scale-space images  $L_{S^2}(\cdot; kt)$  in the original wide-angle image plane. Both these steps require an interpolation of image data, which is effectively a smoothing operation, that ‘destroys’ the fine detailed structure in the image and introduces artifacts. This reduces the number of keypoints detected in the image and the number of correct keypoint correspondences found between image pairs.

sSIFT finds the smallest mean number of correct correspondences followed in ascending order by SIFT (wide-angle), SIFT (perspective), pSIFT (parabolic scales) and pSIFT (equiangular scales). The fact that SIFT (perspective) finds on average more correct correspondences than SIFT (wide-angle), coupled with the improvements in the recall versus 1-precision results, suggests that SIFT (perspective) is a more suitable to use than SIFT (wide-angle) for this image sequence. Overall, pSIFT (equiangular scales) has the second best recall versus 1-precision results and finds on average the most correct correspondences between image pairs. Although the recall versus 1-precision results for SIFT (perspective) are marginally better than those of pSIFT (equiangular scales), it is concluded that pSIFT (equiangular scales) is more suited for this sequence than SIFT (perspective) as it finds on average many more correct correspondences between image pairs. Furthermore, it can detect keypoints in an image spanning the full field of view of the camera. pSIFT (parabolic scales) also performs well and is again able to find on average more correct correspondences between image pairs than SIFT (perspective).

### Tractor

For each frame-rate and keypoint type, the recall versus 1-precision results obtained

using the ambiguity metrics are better than those obtained using the L2 metrics. The fact that the same observation was made for the fisheye and Hyperion sequences gives strong evidence that the ambiguity metrics are more reliable than the L2 metrics for assigning a similarity score to keypoint correspondences.

The recall versus 1-precision performance of SIFT (perspective) is poor in comparison to all other keypoint types for all similarity metrics and frame-rates. This is very different to the results for the Hyperion sequence in which the recall versus 1-precision performance of SIFT (perspective) ranked consistently high. The reason for this is the fact that the Tractor traversed through a non-planar environment. Therefore, unlike the Hyperion sequence, successive images in the Tractor sequence are not related by an approximate Euclidean transform (shift and rotation of the image function). As SIFT (perspective) also finds the smallest mean number of correct correspondences and is unable to detect keypoints in images which span the full field of view of the camera, the results suggest that SIFT (perspective) is the least suitable keypoint type for this image sequence.

Of the remaining keypoint types, the recall versus 1-precision results for sSIFT, pSIFT (parabolic scales) and pSIFT (equiangular scales) rank consistently higher than SIFT (wide-angle) for all similarity metrics and frame-rates. Using the L2 similarity metrics, sSIFT, pSIFT (parabolic scales) and pSIFT (equiangular scales) all show similar recall versus 1-precision performance. Using the ambiguity similarity metrics there is a noticeable improvement in the recall versus 1-precision performance of both pSIFT (parabolic scales) and pSIFT (equiangular scales) scales over sSIFT for each frame-rate. The results for both pSIFT (parabolic scales) and pSIFT (equiangular scales) are very similar.

As just mentioned, SIFT (perspective) finds the smallest mean number of correct correspondences between image pairs for each similarity metric and frame-rate, followed in increasing order by sSIFT, pSIFT (equiangular scales), pSIFT (parabolic scales) and SIFT (wide-angle). It is interesting to note that pSIFT (parabolic scales) finds on average more correct correspondences than pSIFT (equiangular scales) which contradicts the results for the Hyperion and fisheye sequences. In all sequences, the initial scale  $kt_0$  found using the parabolic scales is higher than that found using the original camera model scales (i.e. equiangular or fisheye).

Overall, the results suggest that pSIFT (equiangular scales) and pSIFT (parabolic scales) are the most suitable keypoint types to be used for the Tractor sequence in these experiments. This is particularly true when the ambiguity metrics are used which, as previously discussed, were found to be a more reliable means for assigning a keypoint

similarity score than the L2 metrics (Euclidean distance). Although both found on average fewer correct correspondences between image pairs than SIFT (wide-angle), their recall versus 1-precision performance was significantly better than those of all other keypoint types using the ambiguity metrics, in particular SIFT (wide-angle) and SIFT (perspective).

## 4.7 Conclusions

Two variants of the Scale-Invariant Feature Transform (SIFT) were developed in this chapter, termed sSIFT and pSIFT, that are designed for use with central projection wide-angle cameras. Both reformulated SIFT as an image processing algorithm on the sphere, and when applied to a wide-angle image mapped to the sphere, detect scale-invariant keypoints in a manner invariant to the radial distortion in the image and shift invariant to camera rotation. This approach to image processing with wide-angle images was inspired by the work of Daniilidis et al [56].

sSIFT and pSIFT are required to find a set of scale-space representations of a wide-angle image mapped to the sphere. The ability to do this was made possible by the work of Bülow [31] who proposed the underlying scale-space for functions on the sphere as the solution of the spherical heat diffusion equation and recommended its use for scale-space analysis with wide-angle images [30]. For each method, the scale-space representation of an image is the result of the convolution of the image mapped to the sphere with the spherical Gaussian.

sSIFT implements this convolution in the spherical Fourier domain which requires finding the discrete spherical Fourier transform (spectrum) of a wide-angle image using some sample rate  $b$ . A methodology to estimate the bandwidth of a wide-angle image mapped to the sphere was presented. This bandwidth is dependent on the camera intrinsic parameters and can be used to select a minimum required sample rate  $b$ . An anti-aliasing interpolation filter was designed to be used when this sample rate exceeds the maximum permissible value. pSIFT approximates the convolution of an image mapped to the sphere with the spherical Gaussian as an efficient convolution operation on the stereographic image plane. Although the scale-space images produced have a non-uniform scale, this is not a limiting factor as the image is analysed across a wide range of scales. A number of practical methods used to further improve computational efficiency were implemented, including cascade filtering, approximate separable convolution, and the same octave based approach to image processing used by SIFT. The resulting computation expense of keypoint detection using pSIFT is equivalent to

SIFT. A method to select a suitable set of scales for sSIFT and pSIFT was presented based on the scales used by SIFT and the camera intrinsic parameters.

After sSIFT and pSIFT obtain a set of scale-space images, candidate keypoints are detected as local extrema in the difference of scale-space (difference of Gaussian) images. Edge responses are then removed using the same method as SIFT to find the final set of keypoints. The accuracy of the position and characteristic scale of these keypoints are then improved using the quadratic interpolation scheme developed by Brown and Lowe [29], which is the same scheme used by SIFT. The support region for an sSIFT or pSIFT keypoint is defined as a circle on the sphere, centred at the position of the keypoint on the sphere, and whose size is set dependent on the characteristic scale of the keypoint. The greyscale intensity values within this support region are then mapped to a fixed sized patch using an equiangular projection. The SIFT descriptor is then evaluated from greyscale intensity values on this patch. This process used to find the descriptors is invariant to the camera's distortion.

The suitability of the use of sSIFT and pSIFT with wide-angle images was validated through extensive experiments. The percentage correlation and number of correspondences obtained using SIFT, sSIFT and pSIFT keypoints detected in synthetically generated wide-angle parabolic catadioptric and fisheye images was compared. For these experiments, SIFT was applied directly to the wide-angle images. Only the keypoint detection phase of each algorithm was compared in these experiments, and it was found that overall sSIFT and pSIFT gave either comparable or improved performance over SIFT. SIFT was also found to perform poorly in comparison to sSIFT and pSIFT with respect to the percentage of correct correspondences found towards the periphery of the fisheye images. The second set of experiments compared the abilities of SIFT, sSIFT and pSIFT to correctly detect and match corresponding keypoints in three real outdoor wide-angle image sequences. SIFT was applied directly to the wide-angle images, and to the rectified perspective images produced from the original wide-angle images. The performance metrics used to compare the relative performances were recall versus 1-precision and the mean number of correct correspondences. Overall, pSIFT was found to perform consistently well and on average better than that of SIFT applied to both the wide-angle and perspective images. SIFT applied to the perspective images outperformed all other keypoint types on one image sequence, but performed poorly in comparison to sSIFT and pSIFT in the other sequences.

It is concluded that of all the keypoint types compared in the experiments, the observations made give evidence that pSIFT was the ideal keypoint detector and descriptor to use with the wide-angle images. It is efficient to implement, is not limited



by the same sample rate issued faced by sSIFT, and can be used with any central projection wide-angle camera. The next chapter will further validate the pSIFT keypoint detector by applying it fundamental vision based localisation tasks.

## Chapter 5

# Applications to Vision Based Localisation

*The new approach to keypoint detection in wide-angle images using pSIFT is applied to vision based localisation tasks, including visual odometry and vision-based place recognition. Experimental results for both visual odometry and visual place recognition are given using real-world equian-gular catadioptric and fisheye camera image sequences in unstructured outdoors environments.*

### 5.1 Introduction

Scale-invariant keypoint detection has numerous applications in computer vision, and this chapters considers the particular problem of vision-based localisation. This chapter demonstrates the use of the pSIFT keypoint detection algorithm for visual odometry estimation and visual place recognition with wide-angle images in unstructured outdoors environments.

As discussed in chapter 1, the significant advantage of using wide-angle images for visual odometry applications is the ability to reliably decouple rotation and translation in the estimation of camera egomotion. When using wide-angle images, this requires corresponding keypoints to be both detected and correctly matched in different images across the full field of view of the camera. As shown in chapter 4, the pSIFT keypoint detection algorithm is well suited for this purpose with wide-angle images and is used throughout this chapter. Visual odometry estimates are found using pSIFT for two wide-angle image sequences which have GPS ground truth synchronised with the





Figure 5.1: Fisheye image Sequence used for visual odometry estimates and place recognition. The length of the transit is approximately 4.4 kilometres. The scale in the upper left is distance in metres.

images; the same Hyperion sequence described in detail in section 4.6.3.2 (pg.240), and a fisheye image sequence. The GPS ground truth data for the fisheye sequence is illustrated in figure 5.1. This is the same fisheye camera (with the same camera intrinsic parameters) that was used to obtain the separate image sequence described in detail in section 4.6.3.1 (pg. 239). The sequence contains 1600 fisheye images of size  $1028 \times 764$  pixels.

The visual odometry estimates are found using various constraints on the position of keypoints and motion of the camera. For the Hyperion sequence, a ground plane constraint is used which assumes that the world points associated with all keypoint correspondences are coplanar. Two variations of the standard direct linear transform (DLT), which is used to estimate camera egomotion using a ground plane constraint, are developed that are tailored for use with pSIFT keypoints. They use the same procedure as the ‘standard’ DLT to find an estimate of camera egomotion, and vary from the

standard DLT only with respect to keypoint coordinates used. Both are shown through extensive experiments to find more consistent egomotion estimates than the standard DLT for different camera field of views.

It was argued previously that extending the baseline between incremental estimates of camera egomotion has potential advantages for visual odometry. This was posed as a signal to noise ratio problem. A variable frame-rate algorithm discussed in section 5.2.2 is implemented, based on the algorithm of Mouragnon et al [170], which automatically selects the number of frames between successive estimates of camera egomotion based on a minimum number of keypoints that can be *tracked* through successive frames. The aim of this approach is to increase the magnitude of the change in pose between camera views from which the egomotion estimates are obtained. The accuracy of visual odometry estimates found by integrating the incremental estimates of camera egomotion using this method will be compared experimentally to those where a fixed frame-rate is used.

Finally, applications related to visual place recognition, or vision based loop-closure, are considered in section 5.3. Place recognition results are presented only for the fish-eye sequence as it contains loop closure events. The advantage of using wide-angle images is shown to be the ability to detect previously visited regions of the operating environment despite very large differences in viewpoint. The place recognition algorithm uses the appearance based “Video Google” algorithm [210] which is extended to incorporate a visual word *reliability* metric that is validated through experiments. This algorithm is selected as it is ideal for demonstrating the ability of the pSIFT keypoint detector to find the same keypoints in different wide-angle images of the same scene taken at different viewpoints. The results found using the pSIFT keypoint detector suggest that it would be ideally suited for higher level visual SLAM algorithms [52].

## 5.2 Visual Odometry

A fundamental requirement of all visual odometry algorithms is the ability to estimate the relative pose, to scale, between two camera views which defines the camera egomotion. Using well established fundamentals of two view geometry in computer vision [95], this can be achieved by measuring quantitatively the change in appearance of the environment between views by first detecting keypoints in each image, finding the keypoint correspondences between the images, and then finding the change in pixel coordinates of the corresponding keypoints between the images.

More formally, consider some camera which observes a scene at two different positions and refer to these as camera frame 1 and 2 respectively. Assume that there exists a set of perfect keypoint correspondences  $\mathbf{u} \leftrightarrow \mathbf{u}'$ , where each  $\mathbf{u}_i$  and  $\mathbf{u}'_i$  are the coordinates of the keypoints in image. As each pair of corresponding keypoints  $\mathbf{x}_i \leftrightarrow \mathbf{x}'_i$  implies the world point correspondence  $\mathbf{X}_i \leftrightarrow \mathbf{X}'_i \in \mathbb{R}^3$ , where  $\mathbf{X}_i$  and  $\mathbf{X}'_i$  are defined in camera frame 1 and 2 coordinates respectively, the set of all correspondences  $\mathbf{X} \leftrightarrow \mathbf{X}'$  are related by the rigid body transform

$$\mathbf{X}' = R\mathbf{X} + \mathbf{t}, \quad \forall \mathbf{X}_i, \mathbf{X}'_i, \quad (5.1)$$

where  $R \in SO(3)$  is a rotation and  $\mathbf{t} \in \mathbb{R}^3$  is a translation. This rotation and translation describes the change in pose of the camera from frame 1 to 2.

Recall from chapter 2 that for a perspective camera, the camera matrix  $P$  is defined by the camera's intrinsic and extrinsic parameters. For simplicity, it is assumed from now on that the camera matrix is defined only by the camera extrinsic parameters. Then for any world point  $\mathbf{X}$ , the position  $\eta$  of the world point on the unit view sphere centred at the position of the camera is obtained as

$$\check{\mathbf{x}} = P\mathbf{X}, \quad \eta = \frac{\check{\mathbf{x}}}{\|\check{\mathbf{x}}\|}. \quad (5.2)$$

The parameter  $\check{\mathbf{x}} = (\check{x}, \check{y}, \check{w})^T$  should not be confused with the coordinate  $\mathbf{x} = (x, y)^T = \mathbf{u} - \mathbf{u}_0$  of a point in an image. When estimating the camera egomotion between views, it is convenient to fix the position of the first camera  $P_1$  at the origin of the world coordinate frame and let  $P_1 = [I_{3 \times 3} | \mathbf{0}]$ , where  $I_{3 \times 3}$  is the  $3 \times 3$  identity matrix. Then, for the rotation  $R$  and translation  $\mathbf{t}$  defined in equation 5.1, the second camera matrix  $P_2$  is

$$P_2 = [R | \mathbf{t}], \quad (5.3)$$

which is often written as

$$P_2 = R[I | -\mathbf{C}'], \quad (5.4)$$

where  $\mathbf{C}' = -R^{-1}\mathbf{t}$  is the position of the second camera in the world coordinate frame — the position of the first camera is  $\mathbf{C} = (0, 0, 0)^T$ . By integrating incremental estimates of the camera egomotion, the pose of the camera (position and rotation) relative to some reference start position can be found. This is the core of visual odometry. As a special note, although the relative pose between views described in equation 5.1 is defined using the position of the world points  $\mathbf{X}, \mathbf{X}'$ , in general for monocular visual odometry the position of these points cannot be recovered directly from the position of

the keypoints  $\mathbf{u}, \mathbf{u}'$  in the image. However, the change in position in the image plane of corresponding keypoints is sufficient for deriving the relative pose between views using standard methods [95], albeit often up to an unknown scale ambiguity in the magnitude of the translation  $\mathbf{t}$ .

Several methods are used to estimate visual odometry in this chapter, and they differ in the constraints/assumptions made regarding the position of world points and camera egomotion. They are:

**Ground Plane - Euclidean (section 5.2.6):** The set of keypoint correspondences between views are coplanar world points, lying in the ground plane. The camera motion is assumed to be constrained to a translation in this plane, and rotation about an axis orthogonal to this plane. The height of the camera from the plane is used to resolve the correct scale of the camera translation. This method is used to estimate visual odometry for the Hyperion sequence only.

**Ground Plane - Triggs' method (section 5.2.7):** The set of keypoint correspondences between views are coplanar world points, lying in the ground plane. No assumptions are placed on the camera motion. Again, the height of the camera from the plane is used to resolve the correct scale of the camera translation. This method is used to estimate visual odometry for the Hyperion sequence only.

**Generalised (section 5.2.9):** The world points associated with all keypoint correspondences can lie anywhere in Euclidean space. No assumptions are placed on the camera motion. This can be considered as the most *generalised* means for estimating the visual odometry. Unlike the ground plane methods, the magnitude of the translation  $\mathbf{t}$  cannot be directly resolved. However, it can be found relative to previous frames as will be discussed. This method is used to estimate visual odometry for both the Hyperion and the fisheye sequences.

Before discussing how the visual odometry estimates are found using each of the methods and presenting experimental results, the process used to find the keypoint correspondences between frames is first discussed in section 5.2.1. This is followed by an outline of the variable frame-rate selection method in section 5.2.2, and the methodology used to measure the accuracy of the visual odometry estimates, for each of the ground plane constraints, in section 5.2.4.

### 5.2.1 Keypoint Detection and Matching

For each image in both the Hyperion and fisheye sequences, pSIFT keypoints were detected in the first  $n_{oct} = 5$  octaves of scale-space using  $n_{spo} = 3$  scales per octave. The original wide-angle camera intrinsic parameters were used to select the scales, as described in section 4.3.2 (pg. 174). The process described in section 4.4.1 was used to convert each of the original greyscale wide-angle images in each image sequence to a stereographic image, as required by pSIFT. The edge removal threshold used for each image sequence was  $r_{edge} = 10$ , and the difference of Gaussian thresholds used for the fisheye and Hyperion sequences was 0.01 and 0.0075 respectively (the stereographic images have greyscale intensity values in the range 0-1). A smaller threshold was used for the Hyperion sequence as the images contain predominantly fine detail structure, for example small rocks. Although this threshold was set empirically, further research could consider an adaptive threshold based on on-line learning.

In the following experiments, corresponding keypoints are found between image pairs by matching the keypoint descriptors using the ambiguity metric, as described previously in section 4.6.5 (pg.248). The ambiguity threshold score used was set to 0.9. False positives are removed using RANSAC [70] and Nistér’s five-point algorithm [180] which solves for the essential matrix  $E$  between views from a minimum of 5 keypoint correspondences — a detailed discussion of the essential matrix will be reserved for section 5.2.9. Note that the essential matrix is suitable to use for rigid and unstructured environments. For the case where the ground plane constraint is enforced, a more specific error metric could be used. However, for generality the essential matrix is used. When using RANSAC, given an estimate of the Essential matrix  $E$ , a corresponding pair of keypoints with spherical coordinates  $\eta$  and  $\eta'$  is considered correct if  $|\eta'^T E \eta| < thresh_E$ , where  $thresh_E$  is some threshold value.

### 5.2.2 Frame-Rate Selection for Egomotion Estimation

In the context of visual odometry, the frame-rate is defined to mean the number of images processed per second. Often the frame rate is set as fast as possible and typically constrained by computation time. Although this may seem a logical approach, and has been shown to provide reliable results in visual SLAM implementations [59], there are two conflicting effects which need to be considered for visual odometry applications.

For a fast frame rate there will typically be a large number of keypoint correspondences between successive images but poor signal to noise ratio in the change of key-



point positions on the image. This has the potential to limit both the accuracy of the egomotion estimates between frames and the accurate reconstruction of the scene points (particularly in the depth direction). This is analogous to the case of trying to reliably resolve the depth of a point from stereo correspondences with a small baseline between the cameras. If the frame rate is reduced, although the signal to noise ratio will be improved, there will in general be fewer keypoint correspondences between images. In some instances, the number of keypoints could be less than the number required to resolve the motion of the camera. Furthermore, the accuracy of the egomotion estimate may be more sensitive to outliers. The accuracy of visual odometry estimates found using a *fixed* frame-rate and a *variable* frame-rate are compared in this chapter. Those found using a fixed frame-rate integrate the egomotion estimates between each successive frame in the sequence (i.e. with reference to the recall versus 1-precision experiments in chapter 4, the frame-rate is 1). The estimate found using a variable frame-rate integrates the egomotion estimates between automatically selected frames.

There are a number of different methods that can be used to select the frames for the variable frame-rate approach. Nistér [181] for example simply selected every  $n^{\text{th}}$  frame, which is in effect a fixed frame-rate of  $n$ . The method used in later experiments is based on the algorithm of Mouragnon et al [170], and selects the frames used to compute camera egomotion based on a minimum number of correspondences that can be tracked between them. This same approach was used more recently by Tardif et al [218] with success for visual odometry estimation in a large scale outdoor environment. Keypoints are tracked from some start image  $I$  over multiple frames until the number of tracked keypoints falls below a threshold of  $n = 150$  at image  $I'$ . Any additional correspondences between images  $I$  and  $I'$  are then found by matching directly the keypoints in images  $I$  and  $I'$ . If the number of total correspondences exceeds  $2n$  the tracking continues, otherwise the camera egomotion is estimated and the process restarted from image  $I'$ . The second step is used to account for cases where the camera remains stationary — the number of tracked keypoints degrades over time even if the camera remains stationary. In the context of this work, a keypoint is tracked simply if it can be repeatedly detected and matched (using the ambiguity measure) across multiple frames and should not be confused with feature based tracking methods such as KLT.

Figure 5.2 shows, for each image sequence, the number of frames between each estimate of camera ego-motion using the variable frame-rate algorithm and pSIFT keypoints. Notice that for the Hyperion sequence the number of frames is in excess of 100 on two separate occasions which occurred when the robot was temporarily stationary. No constraints regarding the distribution of the tracked keypoints in the im-

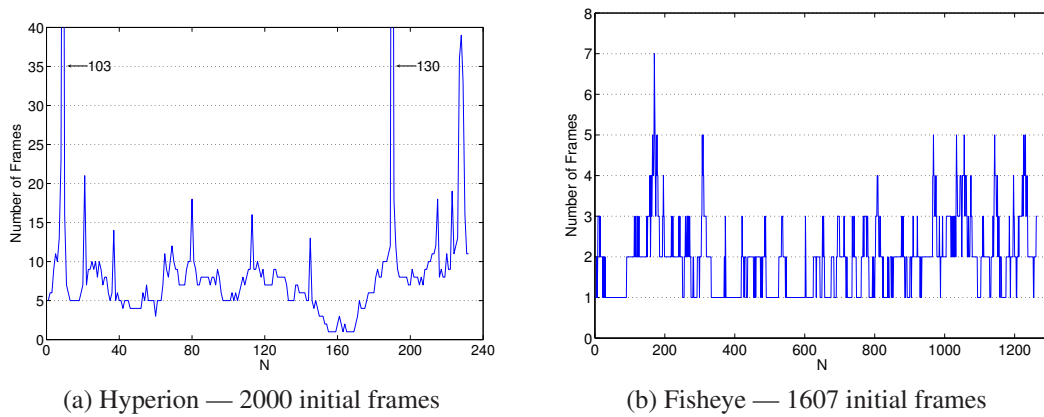


Figure 5.2: Number of frames between estimates of camera egomotion using the *variable* frame-rate algorithm and parabolic SIFT keypoints for the Hyperion and fisheye sequences. The parameter  $N$  is the current iteration of the egomotion estimate.

ages was used. The threshold used to reject outliers in these examples were set to  $thresh_E = 0.0025$  and  $thresh_E = 0.005$  for the fisheye and Hyperion sequences respectively. The reason for selecting a larger threshold for the Hyperion sequence is discussed in the next section.

### 5.2.3 Visual Odometry Trials

In the following visual odometry experiments for the Hyperion sequence, results are found for multiple *trials*. Using the fixed frame-rate, the same set of keypoints are used for each trial. However, the keypoint matching phase is implemented separately for each trial. As RANSAC is a random process, this means that for each trial there will potentially be a different set of corresponding keypoints between each pair of images. The threshold  $thresh_E$  used for each trial is set to 0.0025. Using the variable frame-rate, the original set of keypoint correspondences between the automatically selected frames are used. However, as the threshold  $thresh_E$  used to track the keypoints and automatically select these frames in the previous section used a threshold of  $thresh_E = 0.005$ , there are multiple incorrect correspondences between the frames used to compute the egomotion. For each trial using the variable frame-rate, RANSAC is applied again to these sets of correspondences between the automatically selected frames using a threshold of  $thresh_E = 0.0025$  — the actual frames used to compute the camera egomotion never change. Again, this means that for each trial there will be differences in the final sets of keypoints used to compute the egomotion. Multiple trials are used to increase the number of observations used to measure the accuracy of the visual odometry estimates found using a particular method.



### 5.2.4 Measuring Visual Odometry Accuracy

In the following experiments for the ground plane constraints (Euclidean and Triggs' method), the accuracy of the visual odometry estimates obtained using various methods are compared. Rather than compare the accuracy qualitatively by plotting the visual odometry estimates versus ground truth, a quantitative means for comparison is utilised.

Quantitative analysis of visual odometry accuracy was introduced by Johnson et al [109]. Given a visual odometry estimate and known ground truth (GPS), they measured the error in the visual odometry estimates over fixed length 100m segments. With the first segment starting from the origin, the start point was incremented in 1 metre intervals and a new 100m segment used. The error between segments was found by first shifting them to the same start point and then rotationally aligning them using the  $n^{\text{th}}$  observation in the visual odometry and GPS ground truth segments. The error was defined as the maximum distance between the ground truth position and the corresponding visual odometry estimate over the segment. The advantage of using small segments rather than the whole path is that it is less sensitive to any single inaccurate measurement, particularly rotation, which is integrated over the remaining length of the path.

Rather than simply used a single fixed length segment, the method of Johnson et al [109] was utilised by Nourani et al [184, 185] to measure visual odometry accuracy as a function of distance travelled, that is, by using a number of different length segments. For the ground plane constraint experiments using the Hyperion sequence, this general approach is adopted using different length segments. However, in contrast to Johnson et al and Nourani et al, a different means for measuring the error between a visual odometry and ground truth segment is used. Before justifying the selection of this method it will first be discussed.

Assume that a segment of the path has been selected where the corresponding visual odometry and ground truth data is available, as shown in figure 5.3a. This method is only used to measure the accuracy of the visual odometry estimates using a ground plane constraint, so the estimated visual odometry has only  $x, y$  coordinates in the ground plane. Referring to figure 5.3b, each segment is first aligned to a canonical orientation using the first and last observation in each and shifted so that the first and last observation in each are at equal distance from the origin. Finally, referring to figure 5.3c, each is then shifted in the  $y$  direction so that the centroid (mean) in the  $y$ -direction is zero. Figure 5.3d shows the visual odometry and ground truth segment

is overlaid where the error in the visual odometry estimate is taken to be twice the maximum Euclidean distance between all observations in the the visual odometry and the ground truth segments.

The reason for selecting this method over the metric used by Johnson et al [109] is that the GPS data may have inaccuracies, and aligning two segments using only the first few points in each segment will be sensitive to these inaccuracies. Furthermore, both the fixed and variable frame-rate algorithms need to be compared in following experiments. There will therefore be some bias in aligning the segments using only the first  $n$  observations since the distance travelled over these first  $n$  observations can be different for the fixed and variable frame-rates.

The comparisons made in the following experiments use segment lengths of 25, 50, 75, 100, 125, 150 and 175 metres. For each length, the start point for the first segment is the start of the transit. Once the error for the segment has been found, the start point is incremented to the next nearest observations two metres from the previous start point and the next segment taken. This process is completed for the given length segment over the total length of the path. The visual odometry errors are displayed as box plots with an example from one of the following experiments shown in figure 5.6a. When multiple trials are used, all observations (errors) for a given length segment over all trials are combined into a single set of observations. The box plot values are then found from this single set of observations. Any observations extending from the lower (25%) or upper (75%) quartile by more than 1.5 times the interquartile range are considered outliers and are removed. This is the quantitative means for comparing the accuracy of visual odometry estimates and shows how the visual odometry errors grow with distance travelled.

### 5.2.5 Ground Plane Visual Odometry with Coplanar World Points

A ground plane constraint is defined here to mean that the set of world points associated with the keypoint correspondences between views are coplanar; they lie in a single ground plane. This constraint has been used with success to estimate the visual odometry of a wide-angle camera in an outdoor environment by Scaramuzza [198]. A ground plane constraint is used to estimate visual odometry for the Hyperion sequence. As the robot on which the camera is mounted is known to traverse over an approximately planar environment, and the approximate orientation of the camera with respect to this plane is known, the correspondences associated with world points assumed to lie in this (ground) plane can be found. In the strictest sense this assumption is not correct

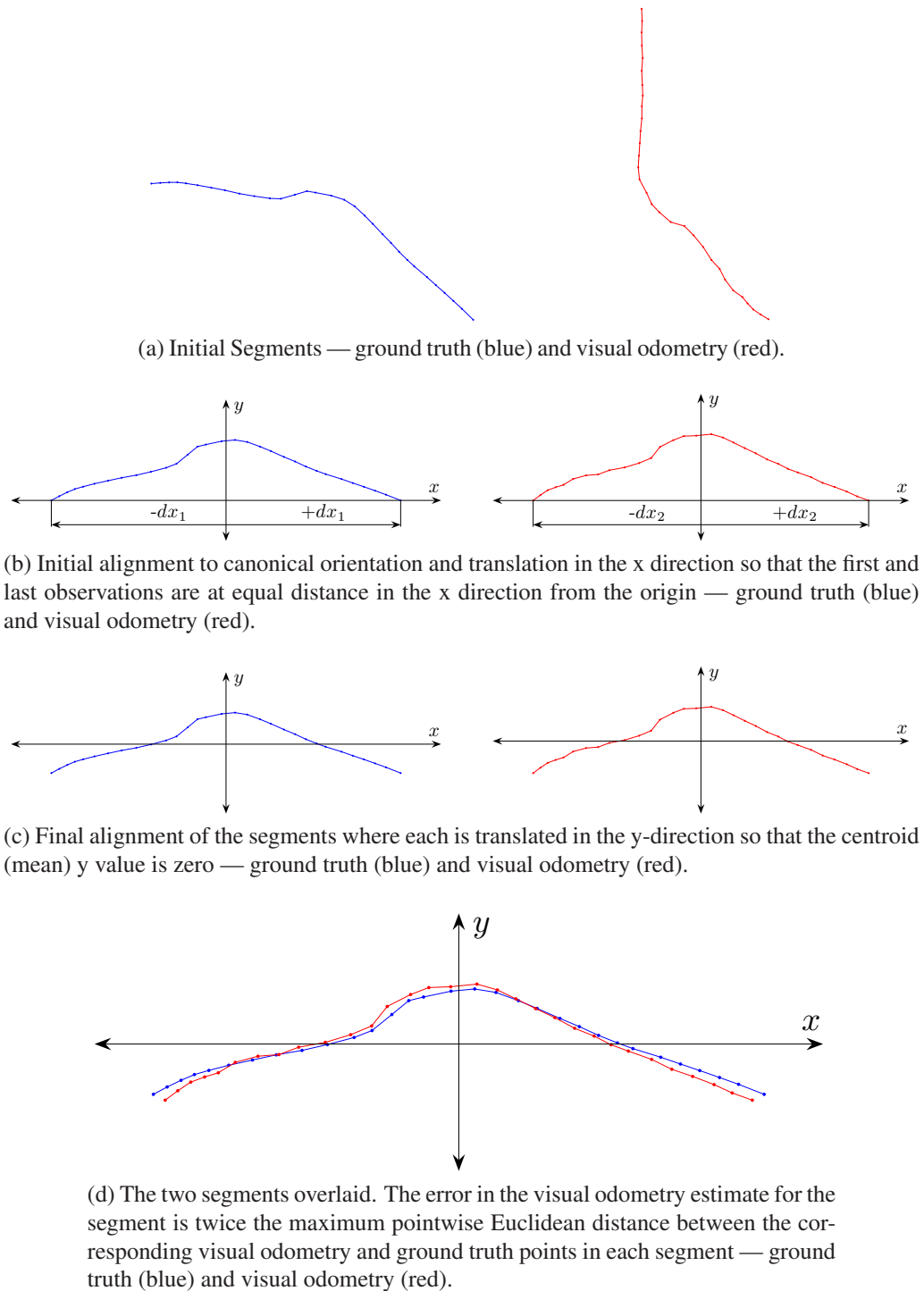


Figure 5.3: Alignment of visual odometry and ground truth segments used to find the error in the visual odometry estimate. The ground truth segments are shown in blue, and the visual odometry segments shown in red. The error of the visual odometry estimate for the segment is taken to be twice the maximum pointwise Euclidean distance between all corresponding points in the aligned segments shown in (d).

since many of the keypoints detected are small rocks (i.e. they are elevated above the ground plane). However, for practical purposes it is a valid assumption and carries with it one major advantage. Assuming that the camera remains at a fixed and known height from the ground plane allows the scale ambiguity in magnitude of the translation to be resolved, as will be shown in further discussions. This remainder of this section introduces the foundations for egomotion estimation using a ground plane constraint. Sections 5.2.6 and will describe how the egomotion is estimated using different constraints on the vehicle motion and present experimental results.

Recall from equation 5.1 that the points  $\mathbf{X}$  and  $\mathbf{X}'$  are related by a rigid body rotation and translation whereby  $\mathbf{X}' = R\mathbf{X} + \mathbf{t}$ . Let  $\mathbf{n}_p$  be the unit normal to the ground plane containing the world points. From the definition of a plane the following relationship can be written:

$$\mathbf{n}_p^T \mathbf{X} = h_p \quad (5.5)$$

where  $h_p$  is the distance of the plane from the camera. Equation 5.1 can then be rewritten using 5.5 as

$$\mathbf{X}' = \left( R + \frac{\mathbf{t}\mathbf{n}_p^T}{h_p} \right) \mathbf{X} \quad (5.6)$$

$$= H\mathbf{X}, \quad \forall X_i, X'_i, \quad (5.7)$$

where  $H$  is the planar homography from frame 1 to 2, and  $H^{-1}$  is the planar homography from frame 2 to 1 whereby  $\mathbf{X} = H^{-1}\mathbf{X}'$ .

Unless the exact pose of the camera with respect to the ground plane is known, the inhomogeneous coordinates  $\mathbf{X}$  and  $\mathbf{X}'$  of the world points cannot be found directly from the keypoint coordinates on the image. However, equation 5.6 can be rewritten using the spherical coordinates  $\eta$  and  $\eta'$  of the keypoints correspondences as

$$\eta' \sim H\eta, \quad \forall \eta_i, \eta'_i, \quad (5.8)$$

where the notation  $\sim$  defines an equivalence up to some unknown scale factor other than 0. In most standard texts dealing with perspective cameras [95], the equivalence in equation 5.8 is written using the homogeneous coordinates of keypoints detected in a perspective image. Given the spherical coordinates  $\eta, \eta'$  of keypoints detected in

wide-angle images, then

$$\check{\mathbf{x}}_i = \frac{\eta_i}{\eta(z)_i} = (\check{x}_i, \check{y}_i, \check{w}_i)^T, \quad (5.9)$$

$$\check{\mathbf{x}}'_i = \frac{\eta'_i}{\eta'(z)_i} = (\check{x}'_i, \check{y}'_i, \check{w}'_i)^T \quad (5.10)$$

can be considered as the the homogeneous coordinates of these keypoints if they were detected in a perspective image. Equation 5.8 is more commonly written using the homogeneous coordinates  $\check{\mathbf{x}}, \check{\mathbf{x}}'$  as

$$\check{\mathbf{x}}' \sim H \check{\mathbf{x}}, \quad \forall \check{\mathbf{x}}_i, \check{\mathbf{x}}'_i. \quad (5.11)$$

For a set of keypoint correspondences  $\check{\mathbf{x}} \leftrightarrow \check{\mathbf{x}}'$ , any two points  $H \check{\mathbf{x}}_i$  and  $\check{\mathbf{x}}'_i$  are known to be equivalent up to an unknown scale factor from equation 5.11. This equivalence can be expressed by the vector cross product  $\check{\mathbf{x}}'_i \times H \check{\mathbf{x}}_i = \mathbf{0}$ ; this equation states that for a perfect correspondence  $\check{\mathbf{x}}_i \leftrightarrow \check{\mathbf{x}}'_i$ , the vectors  $H \check{\mathbf{x}}_i$  and  $\check{\mathbf{x}}'_i$  are perfectly aligned (although not necessarily the same length). This equivalence gives rise to, for each keypoint correspondence, three linear equations in  $\mathbf{h}$ :

$$\begin{bmatrix} \mathbf{0}^T & -\check{w}'_i \check{\mathbf{x}}_i^T & \check{y}'_i \check{\mathbf{x}}_i^T \\ \check{w}'_i \check{\mathbf{x}}_i^T & \mathbf{0}^T & -\check{x}'_i \check{\mathbf{x}}_i^T \\ -\check{y}'_i \check{\mathbf{x}}_i^T & \check{x}'_i \check{\mathbf{x}}_i^T & \mathbf{0}^T \end{bmatrix} \mathbf{h} = \mathbf{0}, \quad \mathbf{A} \mathbf{h} = \mathbf{0}, \quad (5.12)$$

where  $\mathbf{h}$  is the column vector containing the elements of the homography  $H$ :

$$\mathbf{h} = (h_1, h_2, \dots, h_9)^T, \quad H = \begin{bmatrix} h_1 & h_2 & h_3 \\ h_4 & h_5 & h_6 \\ h_7 & h_8 & h_9 \end{bmatrix}. \quad (5.13)$$

For the purposes of later discussions, it is important to note here that the equivalence in equation 5.12 can be written as

$$\check{\mathbf{x}}'_i \times H \check{\mathbf{x}}_i = \|\check{\mathbf{x}}'_i\| \|H \check{\mathbf{x}}_i\| \sin \theta \mathbf{n}_{\check{\mathbf{x}}_i, \check{\mathbf{x}}'_i} = \mathbf{0}, \quad (5.14)$$

where  $\mathbf{n}_{\check{\mathbf{x}}_i, \check{\mathbf{x}}'_i}$  is the unit vector orthogonal to both  $\check{\mathbf{x}}'_i$  and  $H \check{\mathbf{x}}_i$ .

Since equation 5.12 has only two linearly independent equations, typically the last column is eliminated for the purposes of obtaining a solution from a minimum of four non-collinear points (the homography can be resolved up to an arbitrary scale factor such that there exist only eight degrees of freedom). A linear solution for  $\mathbf{h}$ , and

consequently the homography  $H$ , can be obtained from equation 5.12 using a minimum of four non-collinear keypoint correspondences. This method is typically referred to as the Direct Linear Transform (DLT) [95]. In most practical applications, the DLT is used to find a first estimate of the homography  $H$  which is then further improved through non-linear iterative refinement (optimisation).

### 5.2.6 Ground Plane Visual Odometry: Euclidean

The visual odometry estimates for the Hyperion sequence obtained in this section use a ground plane constraint (scene points are coplanar), and enforce strictly the assumption that the camera motion between frames is limited to translation  $\mathbf{t}$  in the ground plane, and rotation  $R_z(\phi)$  about the axis orthogonal to the ground plane, where

$$\mathbf{t} = \begin{bmatrix} t_x \\ t_y \\ 0 \end{bmatrix}, \quad R_z(\phi) = \begin{bmatrix} \cos \phi & -\sin \phi & 0 \\ \sin \phi & \cos \phi & 0 \\ 0 & 0 & 1 \end{bmatrix}. \quad (5.15)$$

Furthermore, the constraint is made that the camera's principal axis is parallel with the normal to the ground plane  $\mathbf{n}_p = (0, 0, 1)^T$ , which is known to be approximately true for the Hyperion sequence. By substituting  $\mathbf{n}_p = (0, 0, 1)^T$  and both the translation  $\mathbf{t}$  and rotation  $R$  given in equation 5.15 into equation 5.6, the homography  $H$  takes the specific form

$$H = \begin{bmatrix} \cos \phi & -\sin \phi & t_x/h_p \\ \sin \phi & \cos \phi & t_y/h_p \\ 0 & 0 & 1 \end{bmatrix}. \quad (5.16)$$

The homography  $H$  is a Euclidean transform.

Given both the normal  $\mathbf{n}_p$  to the ground plane and height of the camera  $h_p$  from the ground plane, the position of any world point  $\mathbf{X}_i$  can be found directly from the coordinate of a pSIFT keypoint  $\mathbf{x}_i = (x_i, y_i)$  on the stereographic image relative to the principal point. Letting  $n = m_p + 1$ , where  $m_p$  is the distance of the stereographic image plane from the centre of the view sphere, the inhomogeneous coordinate of a world point  $\mathbf{X}_i$  in the first camera's frame of reference is obtained as

$$\mathbf{X}_i = \left[ \frac{2h_p n x_i}{n^2 - r_i^2}, \frac{2h_p n y_i}{n^2 - r_i^2}, h_p \right]^T = (X_i, Y_i, Z_i)^T, \quad (5.17)$$

where  $r_i = \sqrt{x_i^2 + y_i^2}$ . The inhomogeneous coordinate of a world point  $\mathbf{X}'_i$  relative to

the second camera frame of reference is obtained as

$$\mathbf{X}'_i = \left[ \frac{2h_p n x'_i}{n^2 - r_i'^2}, \frac{2h_p n y'_i}{n^2 - r_i'^2}, h_p \right]^T = (X'_i, Y'_i, Z'_i)^T, \quad (5.18)$$

where  $r'_i = \sqrt{x_i'^2 + y_i'^2}$ . Then for any keypoint correspondence, replacing the coordinates  $\check{\mathbf{x}}_i$  and  $\check{\mathbf{x}}'_i$  in equation 5.12 with the coordinates  $\mathbf{X}_i$  and  $\mathbf{X}'_i$  respectively, and removing the redundant terms in  $\mathbf{h}$ , gives

$$\begin{bmatrix} X_i & -Y_i & 1 & 0 \\ Y_i & X_i & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos \phi \\ \sin \phi \\ t_x/h_p \\ t_y/h_p \end{bmatrix} = \begin{bmatrix} X'_i \\ Y'_i \end{bmatrix}. \quad (5.19)$$

Unfortunately, the Euclidean transform can not be obtained directly as equation 5.16 is not written as a function of the three linearly independent variables  $\phi, t_x, t_y$ . However, using the gold-standard algorithm in [95], a solution for the *affine* homography  $H_A$  can be found which is of the form

$$H_A = \begin{bmatrix} A & \mathbf{t} \\ \mathbf{0}^T & 1 \end{bmatrix}, \quad A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}. \quad (5.20)$$

Since the matrix  $A$  is an affine transform, there is no guarantee that its solution will be a  $2 \times 2$  orthonormal rotation  $R_z(\phi)$  as required. It is necessary to therefore find the best estimate of the rotation  $R_z(\phi)$  given the solution for  $A$ . As discussed in [95], the affine matrix  $A$  can be decomposed as

$$A = R_z(\phi)(R_z(\theta)^T S R_z(\theta)) \quad (5.21)$$

where  $(R_z(\theta)^T S R_z(\theta))$  is the component of the affine deformation and  $R_z(\phi)$  the orthonormal rotation given in equation 5.15. If  $USV = \text{svd}(A)$  is the singular value decomposition of the affine matrix  $A$ , then  $A$  can be written as

$$A = (UV^T)(VSV^T) \quad (5.22)$$

from which, referring to equation 5.21,  $R_z(\phi) = UV^T$ . Since both matrices  $U$  and  $V$  are orthogonal, their matrix product  $R(\phi)$  is orthogonal with determinant 1 as required for a rotation matrix. As described by Scaramuzza [198], this ‘best estimate’ for the



rotation  $R_z(\phi)$  is the optimal solution which minimises the Frobenius norm

$$\min_{R_z(\phi)} \|R_z(\phi) - A\|_F^2, \quad R_z(\phi) R_z(\phi)^T = I_{2 \times 2}, \quad (5.23)$$

where  $I_{2 \times 2}$  is the  $2 \times 2$  identity matrix. The estimate of the rotation  $R(\phi)$  can then be used to find the homography given in equation 5.16. Observe also that given the known height  $h_p$  of the camera from the ground plane, the correct magnitude of the translation  $\mathbf{t}$  can be resolved, which is an advantage of using a ground plane constraint when valid.

The gold standard algorithm in [95] uses the coordinates  $\mathbf{X}$  and  $\mathbf{X}'$  to solve equation 5.19. For convenience, this method will be referred to as the *standard* DLT. Care must be taken using the standard DLT for keypoints detected in wide-angle images. Since the Euclidean transformation is estimated for the keypoints mapped to the ground plane, the results are heavily biased for keypoints near the equator for the downward camera configuration used in the Hyperion sequence (i.e. the camera's principal axis is orthogonal to the ground plane). This can be illustrated by deriving the covariance matrix  $\Sigma = JJ^T$ , where  $J$  is the Jacobian of the transformation from the stereographic image plane to the ground plane. Since all world points are constrained to lie in the ground plane, for simplicity 2D coordinates can be used where, from equation 5.17, the Jacobian is

$$J = \begin{bmatrix} \frac{\partial X}{\partial x} & \frac{\partial X}{\partial y} \\ \frac{\partial Y}{\partial x} & \frac{\partial Y}{\partial y} \end{bmatrix} \quad (5.24)$$

$$= \frac{2h_p n}{n^2 - r^2} \begin{bmatrix} \frac{2x^2}{n^2 - r^2} + 1 & \frac{2xy}{n^2 - r^2} \\ \frac{2xy}{n^2 - r^2} & \frac{2y^2}{n^2 - r^2} + 1 \end{bmatrix}, \quad (5.25)$$

where  $\mathbf{x} = (x, y)^T$  is the coordinate of a keypoint on the stereographic image plane defined with respect to the principal point, and  $r = \sqrt{x^2 + y^2}$ . The covariance matrix is then

$$\Sigma = \left( \frac{2h_p n}{(n^2 - r^2)} \right)^2 \begin{bmatrix} 2n^2(x^2 - y^2) + n^4 + r^4 & 4n^2 xy \\ 4n^2 xy & 2n^2(y^2 - x^2) + n^4 + r^4 \end{bmatrix}. \quad (5.26)$$

Figure 5.4 shows a set of pSIFT keypoints on the image and their associated error (uncertainty) ellipsoids on the ground plane obtained from their covariance matrices. For this illustration, the uncertainty of the position for all keypoints on the stereographic image were assumed to be equal and normally distributed about the positions of the keypoints. It is evident from the figure that the uncertainty of the ground plane

Threshold	Statistic	Distance Travelled (metres)						
		25	50	75	100	125	150	175
$\theta < 90^\circ$	median	$\geq 1000$	$\geq 1000$	$\geq 1000$	$\geq 1000$	$\geq 1000$	$\geq 1000$	$\geq 1000$
	IQR	$\geq 1000$	$\geq 1000$	$\geq 1000$	$\geq 1000$	$\geq 1000$	$\geq 1000$	$\geq 1000$
$\theta < 89^\circ$	median	3.78	5.99	8.27	10.82	13.44	16.09	17.97
	IQR	2.68	2.88	3.00	4.12	4.80	4.31	3.03
$\theta < 80^\circ$	median	1.98	4.09	6.05	7.41	9.02	12.16	17.41
	IQR	1.12	2.09	2.01	3.78	8.31	10.78	10.67
$\theta < 70^\circ$	median	1.99	3.87	5.31	6.93	7.95	9.31	12.57
	IQR	1.12	2.29	1.91	1.86	4.25	6.08	6.32

(a) Fixed frame-rate (see figure 5.6a).

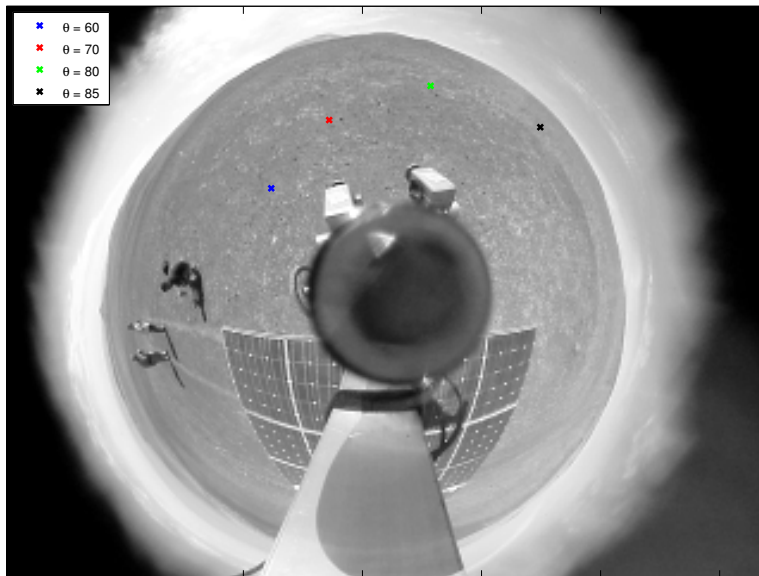
Threshold	Statistic	Distance Travelled (metres)						
		25	50	75	100	125	150	175
$\theta < 90^\circ$	median	24.58	135.50	$\geq 1000$	$\geq 1000$	$\geq 1000$	$\geq 1000$	$\geq 1000$
	IQR	392.02	$\geq 1000$	$\geq 1000$	$\geq 1000$	$\geq 1000$	$\geq 1000$	$\geq 1000$
$\theta < 89^\circ$	median	3.27	5.15	6.97	8.70	10.62	12.07	14.16
	IQR	2.42	2.49	5.14	6.80	7.67	7.83	8.20
$\theta < 80^\circ$	median	1.81	3.96	5.67	6.65	11.24	15.43	20.42
	IQR	1.00	1.98	2.45	7.57	12.72	15.25	13.51
$\theta < 70^\circ$	median	1.87	4.28	6.09	6.49	10.82	14.94	20.00
	IQR	0.81	2.04	3.31	7.70	12.45	15.09	14.20

(b) Variable frame-rate (see figure 5.6b).

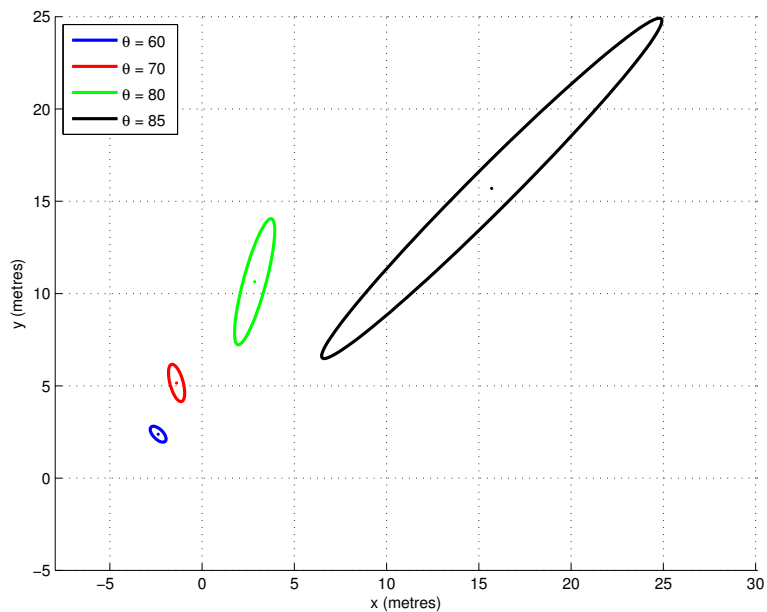
Table 5.1: Visual odometry error as a function of the distance travelled for the Hyperion sequence using the Euclidean ground plane constraint and standard DLT. The smallest median values for each angular threshold are shaded.

position increases significantly for those keypoints found towards the periphery of the stereographic image. Therefore, inaccuracies in keypoint positions on the image plane for those points towards the periphery may greatly effect the linear solution of the homography (Euclidean transform).

To investigate these effects, the visual odometry estimates for the Hyperion sequence were found using the standard DLT for the fixed and variable frame-rates using varying angle of colatitude thresholds  $\theta$  on the camera's effective field of view. If the coordinates any one of the keypoints in a corresponding pair has an angle of colatitude above the threshold, this correspondence is not used to estimate the camera egomotion. As the camera's principal axis is assumed to be orthogonal to the ground plane, a keypoint with spherical coordinate  $\eta(\theta, \phi)$  lies in the ground plane if  $\theta < 90^\circ$ . The threshold values on the angle of colatitude used were  $\theta < 90^\circ$ ,  $89^\circ$ ,  $80^\circ$ , and  $70^\circ$ , as illustrated in figure 5.5. The accuracy of the visual odometry estimates were obtained for both the fixed and variable frame-rate methods for 20 trials each. The results are shown in figure 5.6 and are summarised in tables 5.1a and 5.1b for the fixed and variable frame-rate methods respectively.



(a) Position of keypoints on the stereographic image. For each keypoint,  $\theta$  is the corresponding angle of colatitude on the sphere. The original image is of size  $640 \times 480$  pixels.



(b) Error ellipse for the position of each keypoint on the ground plane. An arbitrary scale factor of 10 has been applied to each for the purposes of visualisation.

Figure 5.4: Position of keypoints on the stereographic image and the corresponding error ellipses on the ground plane derived from the covariance matrix for each keypoint. The point at the centre of each ellipsoid is the position of the keypoint on the ground plane. The angle  $\theta$  is the angle of colatitude of the keypoint on the sphere.

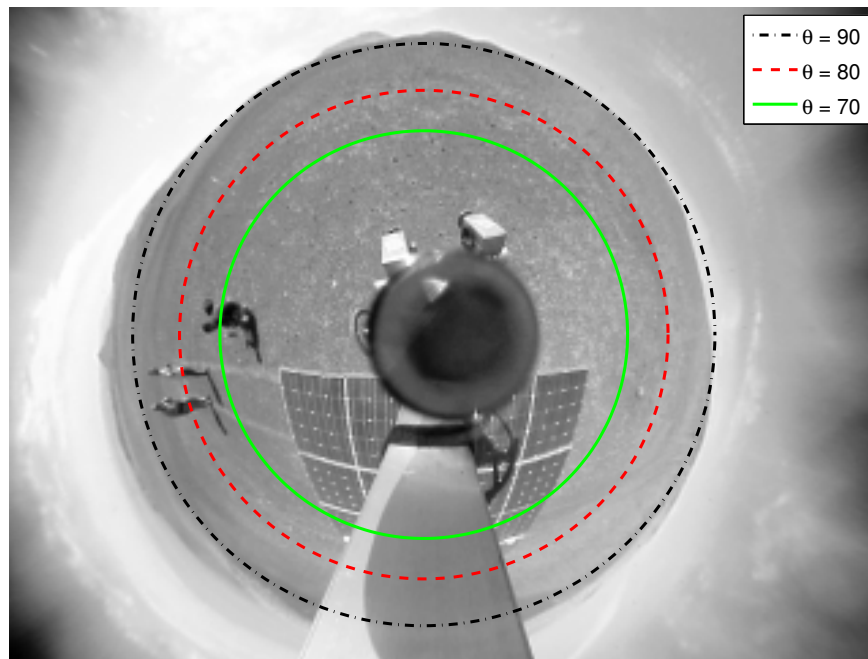
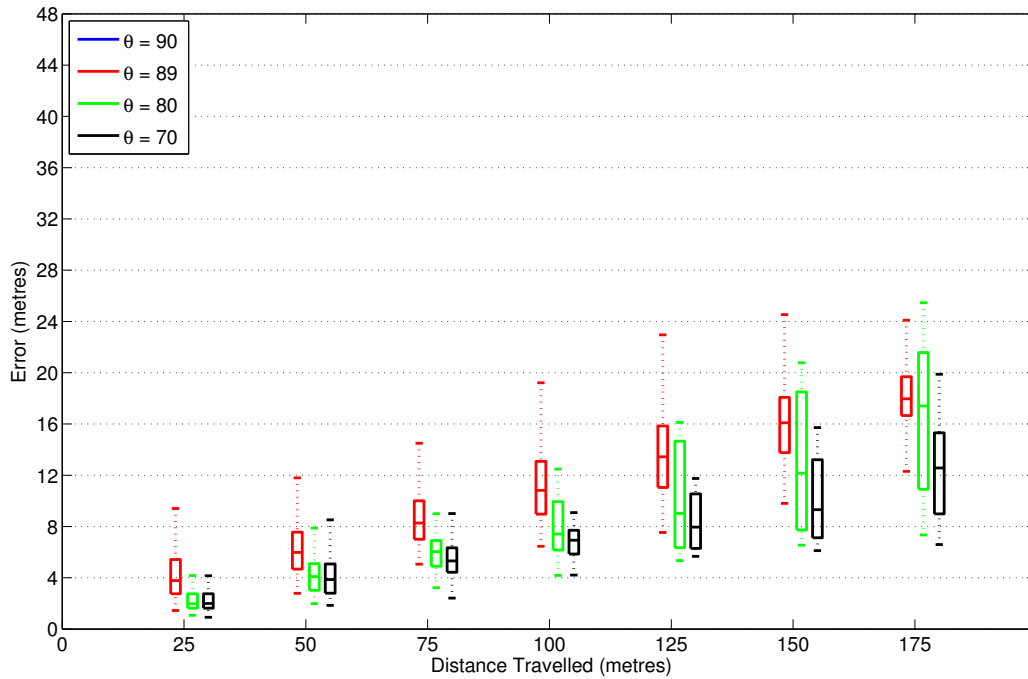


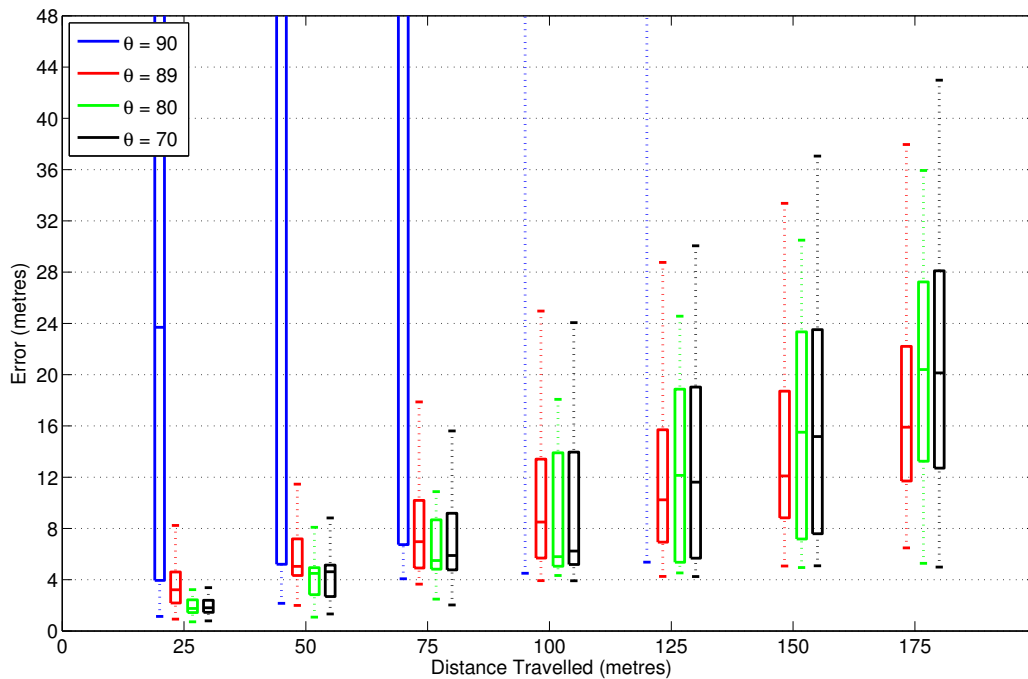
Figure 5.5: The circles on the stereographic image corresponding the angles of colatitude thresholds  $\theta$  (the circle corresponding to  $\theta = 89^\circ$  is not shown due to its close proximity to that for  $\theta < 90^\circ$ ). For a given angular threshold, only the keypoint correspondences with image coordinates  $\mathbf{u}$  and  $\mathbf{u}'$  within the regions enclosed by the circles are used to estimate the camera egomotion.

From inspection of the results, in general the accuracy of the egomotion estimates for the fixed frame-rate improve as the threshold is reduced — the results for the threshold  $\theta < 90^\circ$  are not visible in figure 5.6a as the errors are large. However, the results for the variable frame-rate show that this is not the case. It is observed also that in general, the variable frame rate errors have a higher mean and greater IQR than those for the fixed frame-rate.

Ideally, a linear estimate of the homography should be found which minimises an error defined on the image plane itself as this is the domain in which errors in keypoint location occur during detection. Although a linear solution for the homography has not been found that can achieve this, it is proposed that a modified version of the DLT can be used which attempts to make some account for the uncertainty of keypoint positions in the image plane when obtaining a solution of the homography. This modified DLT is referred to as the *weighted DLT* and is discussed in the next section.



(a) Fixed frame-rate



(b) Variable frame-rate

Figure 5.6: Box plot of the visual odometry errors as a function of the distance travelled for the Hyperion sequence using the Euclidean ground plane constraint and standard DLT. The results have been obtained over a total of 20 trials for both the fixed and variable frame-rates. IQR is the inter-quartile range.

### 5.2.6.1 Weighted DLT

Recall from equation 5.14 that the DLT finds a solution for a generalised homography  $H$  from the constraint

$$\check{\mathbf{x}}'_i \times H \check{\mathbf{x}}_i = \|\check{\mathbf{x}}'_i\| \|H \check{\mathbf{x}}_i\| \sin \theta \mathbf{n}_{\check{\mathbf{x}}_i, \check{\mathbf{x}}'_i} = \mathbf{0}, \quad (5.27)$$

where  $\check{\mathbf{x}}_i$  and  $\check{\mathbf{x}}'_i$  are the homogeneous coordinates of a corresponding pair of keypoints.

This equation can be interpreted as saying that the solution for the parameters  $\mathbf{h}$  of the homography are found which minimise the angles  $\theta_i$  between the vectors  $\check{\mathbf{x}}'_i$  and  $H \check{\mathbf{x}}_i$ , weighted by the magnitude of the same vectors (since  $\theta \approx \sin \theta$  for small  $\theta$ ). In many cases, a solution for the homography is obtained using the normalised homogeneous coordinates  $\check{\mathbf{x}}_i = (\check{x}_i, \check{y}_i, 1)^T$  and  $\check{\mathbf{x}}'_i = (\check{x}'_i, \check{y}'_i, 1)^T$  for all keypoint correspondences. For the camera configuration used in the Hyperion sequence, the camera's principal axis is orthogonal to the ground plane. This means that  $\mathbf{X}_i = h_p \check{\mathbf{x}}_i$  and  $\mathbf{X}'_i = h_p \check{\mathbf{x}}'_i$  for all keypoint correspondences. As shown previously in figure 5.4, this is not ideal for the Hyperion sequence. The reason is that the uncertainty of the positions  $\mathbf{X}_i$  and  $\mathbf{X}'_i$ , with respect to the uncertainty of the pSIFT keypoint positions on the stereographic image, is greater for keypoints near the periphery of the image than those near the principal point.

As noted in equation 5.8, the coordinates of the keypoints  $\eta_i, \eta'_i$  on the unit view sphere can be used to find a solution for the homography  $H$ , where

$$\eta'_i \times H \eta_i = \|\eta'_i\| \|H \eta_i\| \sin \theta \mathbf{n}_{\eta_i, \eta'_i} = \mathbf{0}. \quad (5.28)$$

Assuming for now that  $\|H \eta_i\|$  remains constant for all  $\eta_i$ , then  $\|\eta'_i\| \|H \eta_i\|$  remains constant for all  $\eta_i, \eta'_i$ . A solution for the parameters  $\mathbf{h}$  of the homography  $H$  could then be found which minimises, with equal weighting for each correspondence, the angle between the vectors  $H \eta_i$  and  $\eta'_i$ . The weighted DLT algorithm assumes that  $\|H \eta_i\|$  remains constant for all  $\eta_i$ , and applies an individual weighting to each corresponding pair of keypoints  $\eta_i$  and  $\eta'_i$  representative of their uncertainty in keypoint position on the stereographic image.

To find a suitable weighting, it is necessary to derive an expression which relates the uncertainty of a pSIFT keypoint's position  $\mathbf{x}$  on the stereographic image plane (relative to the principal point) to the uncertainty of the keypoint's position  $\eta$  on the sphere. Since stereographic projection is conformal, assuming that the uncertainty of a keypoint's position on the image is uniform in all directions, the uncertainty of

the keypoint's position on the sphere is locally uniform in all directions. Recall from chapter 4 that a change in angle  $d\psi^2$  along any great circle on the sphere corresponding to a change in spherical coordinates can be approximated as

$$d\psi^2 = d\eta(x)^2 + d\eta(y)^2 + d\eta(z)^2 = d\eta^2. \quad (5.29)$$

The following expression was also derived which relates a change in angle  $d\psi$  on the sphere to a change in pixel position  $dP$  in any direction from a point at a radius  $r$  from the principal point on the stereographic image plane:

$$d\psi^2 = \frac{4m_p^2}{(m_p^2 + r^2)^2} dP^2, \quad (5.30)$$

where  $m_p$  is the distance of the stereographic image plane from the centre of the view sphere, and  $r = \sqrt{x^2 + y^2}$ . The change in angle  $d\psi^2$  is the same irrespective of the direction  $\alpha$  of the unit pixel shift  $dP$  as stereographic projection is conformal. Equation 5.30 can be rewritten as

$$dP = \frac{m_p^2 + r^2}{2m_p} d\psi \quad (5.31)$$

from which the ratio of the displacement on the image plane  $dP(r)$  at radius  $r$  to  $dP(0)$  at radius  $r = 0$  is approximated as

$$\frac{dP(r)}{dP(0)} = \frac{m_p^2 + r^2}{m_p^2}. \quad (5.32)$$

This equation is used as the basis for applying a weighting to the spherical coordinates  $\eta_i$  and  $\eta'_i$ .

Define  $r(\eta_i)$  and  $r(\eta'_i)$  as the radii on the stereographic image plane from the principal point corresponding to the points  $\eta_i$  and  $\eta'_i$  respectively. Using equation 5.32, the weights  $w_i$  and  $w'_i$  applied to the points  $\eta_i$  and  $\eta'_i$  are

$$w_i = \left( \frac{m_p^2 + r(\eta_i)^2}{m_p^2} \right)^2 \quad \text{and} \quad w'_i = \left( \frac{m_p^2 + r(\eta'_i)^2}{m_p^2} \right)^2. \quad (5.33)$$

The weightings  $w_i$  and  $w'_i$  in equation 5.2.7 will be unity if the points  $\eta_i$  and  $\eta'_i$  respectively are at the north pole  $\mathbf{n}$ . This weighing is therefore the relative uncertainty of a keypoint's position on the sphere relative to a point at the north pole. Letting  $\tilde{\eta}_i = w_i \eta_i$  and  $\tilde{\eta}'_i = w'_i \eta'_i$ , the weighted DLT rewrites the first two linearly independent equations



in 5.12 for each correspondence as

$$\begin{bmatrix} \mathbf{0}^T & -\tilde{\eta}(z)'_i \tilde{\eta}_i^T & \tilde{\eta}(y)'_i \tilde{\eta}_i^T \\ \tilde{\eta}(z)'_i \tilde{\eta}_i^T & \mathbf{0}^T & -\tilde{\eta}(x)'_i \tilde{\eta}_i^T \end{bmatrix} \mathbf{h} = \mathbf{0}, \quad \mathbf{A}\mathbf{h} = \mathbf{0}, \quad (5.34)$$

which, for the specific case of the Euclidean ground plane constraint, takes a similar form to equation 5.19:

$$\begin{bmatrix} -\tilde{\eta}(z)'_i \tilde{\eta}(y)_i & \tilde{\eta}(z)'_i \tilde{\eta}(x)_i \\ -\tilde{\eta}(z)'_i \tilde{\eta}(x)_i & -\tilde{\eta}(z)'_i \tilde{\eta}(y)_i \\ 0 & \tilde{\eta}(z)'_i \tilde{\eta}(z)_i \\ -\tilde{\eta}(z)'_i \tilde{\eta}(z)_i & 0 \\ \tilde{\eta}(y)'_i \tilde{\eta}(z)_i & -\tilde{\eta}(x)'_i \tilde{\eta}(z)_i \end{bmatrix}^T \begin{bmatrix} \cos \phi \\ \sin \phi \\ t_x/h_p \\ t_y/h_p \\ 1 \end{bmatrix} = \mathbf{0}, \quad \tilde{\mathbf{A}}\mathbf{h} = \mathbf{0}. \quad (5.35)$$

Unlike the standard DLT used for the Euclidean ground plane constraint, the gold standard solution for affine transforms [95] can not be used as the explicit position of the world points  $\mathbf{X}_i$  and  $\mathbf{X}'_i$  are not used. The solution for the parameters of  $\mathbf{h}$  are therefore obtained from the singular value decomposition of the matrix  $\tilde{\mathbf{A}}$ . Letting  $\mathbf{U}\mathbf{S}\mathbf{V} = \text{svd}(\tilde{\mathbf{A}})$ ,  $\mathbf{h}$  is the column vector of  $\mathbf{V}$  corresponding to the smallest non-zero singular value subject to the condition  $\|\mathbf{h}\| = 1$ . The vector  $\mathbf{h}$  is rescaled by dividing through the last term and the known height of the camera  $h_p$  from the ground plane used to resolve the correct translation values  $\mathbf{t} = (t_x, t_y, 0)^T$ . As was the case using the standard DLT, the rotation angle  $\phi$  is not be written as a separable linear term. Thus, after rescaling the vector  $\mathbf{h}$  there is no guarantee that the values returned for  $\cos \theta$  and  $\sin \theta$  in equation 5.35 satisfy the required constraint  $\sin^2 \theta + \cos^2 \theta = 1$ . The best estimate of the orthonormal rotation  $R_z(\phi)$  is therefore found using the same methodology described in the previous equations 5.21 through 5.23.

For future reference, this same procedure can also be used to obtain a solution for  $\mathbf{h}$  with the weighted spherical coordinates  $\tilde{\eta}, \tilde{\eta}'$  in the matrix  $\tilde{\mathbf{A}}$  in equation 5.35 replaced with the unweighted spherical coordinates  $\eta, \eta'$ . The visual odometry estimates will be found using the unweighted spherical coordinates in the following sections. For convenience, this method will be referred to as the spherical DLT.

### 5.2.6.2 Iterative Weighted DLT

One potential limitation of the weighted DLT is the fact that  $\|H\eta_i\|$  is not guaranteed to be constant for all  $\eta_i$ ; the exception is for the case where the homography is a pure

rotation. Therefore, the magnitude of  $H\eta_i$  and hence the overall weighting applied to each pair of correspondences is dependent on both  $H$  and  $\eta_i$ . One could argue that if the coordinates of each point  $\eta_i$  were multiplied by some arbitrarily large scalar  $n$ , then the magnitude of the vectors  $nH\eta_i$  would approach a constant value. However, doing this may lead to numerical instability when solving for  $\mathbf{h}$  (the solution would be obtained from the singular value decomposition of  $\mathbf{h}$ ). This relates to the concept of numerical ‘preconditioning’ where, in applications specific to computer vision, large input values are to be avoided [95]. Ideally, it would be desirable to derive for each  $\eta_i$  some constant  $k_i$  for which  $\|k_i H\eta_i\| = 1$ . Unfortunately, without knowing the parameters of the homography  $H$  this constant  $k_i$  can not be obtained.

The iterative weighted DLT first finds an estimate of  $\mathbf{h}$  using the weighted DLT, where  $\|\mathbf{h}\| = 1$ , from which the homography  $H$  is obtained. The scale factor  $k_i$  for each keypoint position  $\eta_i$  is then found from the initial estimate of  $H$  as

$$k_i = \frac{1}{\sum (H\eta_i)^2}. \quad (5.36)$$

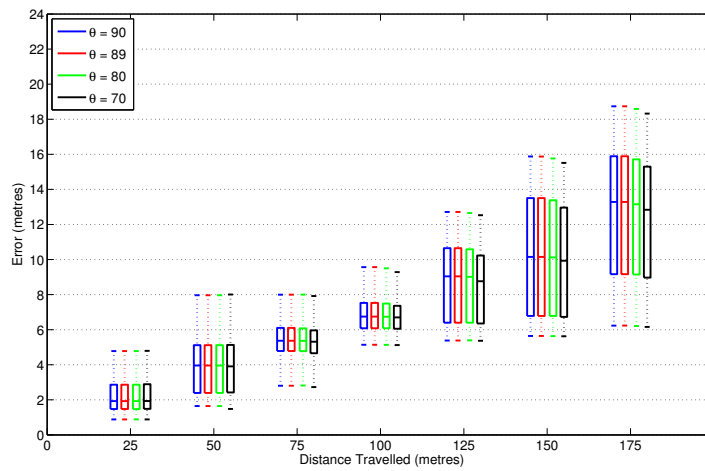
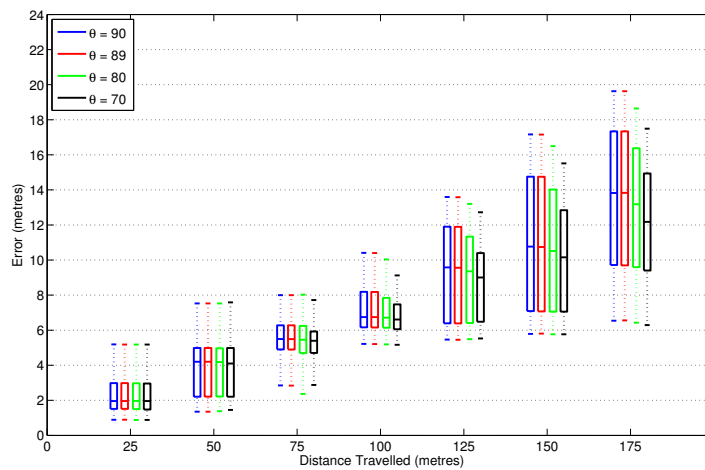
The weighted DLT is then used to find a new estimate of the homography using the weighted spherical coordinates  $\tilde{\eta}_i = k_i w_i \eta_i$  — the points  $\tilde{\eta}_i'$  remain the same. If desired, this process could be repeated indefinitely where for each iteration the constant  $k_i$  for each point  $\eta_i$  is recalculated for the current estimate of the homography  $H$  obtained from  $\mathbf{h}$ . In the following experiments, this process is not iterated, that is, the weighting factors  $k$  are estimated only once.

### 5.2.6.3 Experiments: Linear Estimate

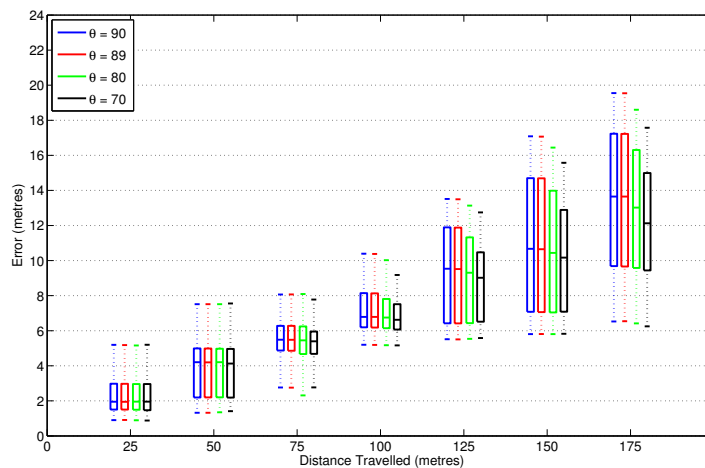
The visual odometry estimates for the Hyperion sequence were found again for varying angular thresholds  $\theta$ . These results were obtained separately using the spherical DLT, weighted DLT, and the iterative weighted DLT. The same set of keypoint correspondences used in each of the 20 trials used to find the results in figure 5.6 were used again.

## Results

Figures 5.7 and 5.8 show the box plots of the visual odometry errors for the fixed frame-rate and variable frame-rate respectively. The median errors and interquartile ranges for the fixed frame-rate and variable frame-rate are given in tables 5.2 and 5.3.

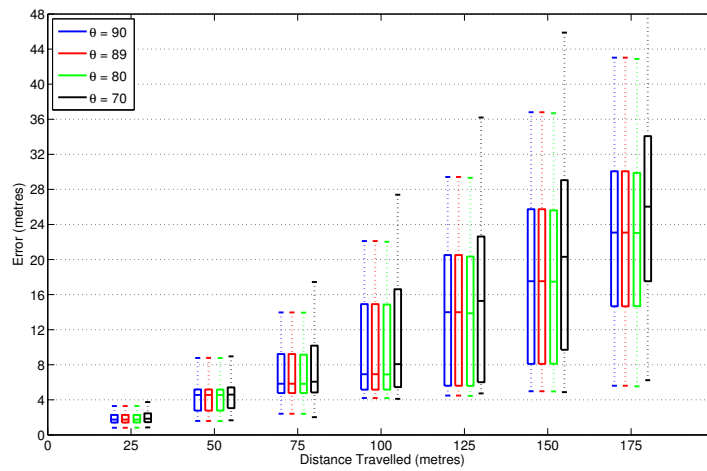
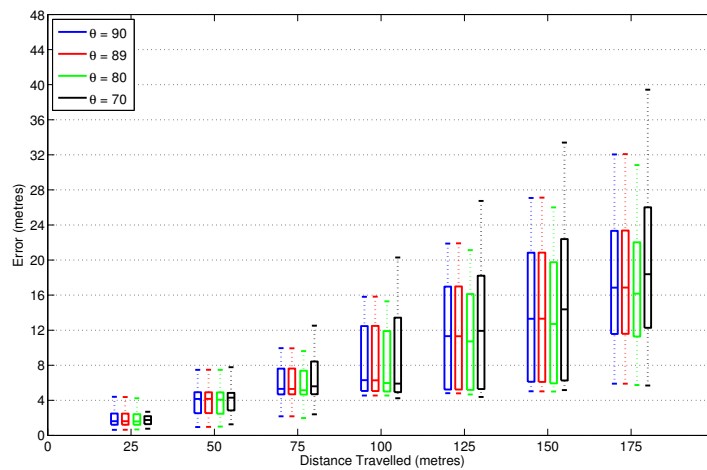
(a) Points on sphere  $\eta, \eta'$ 

(b) Weighted DLT

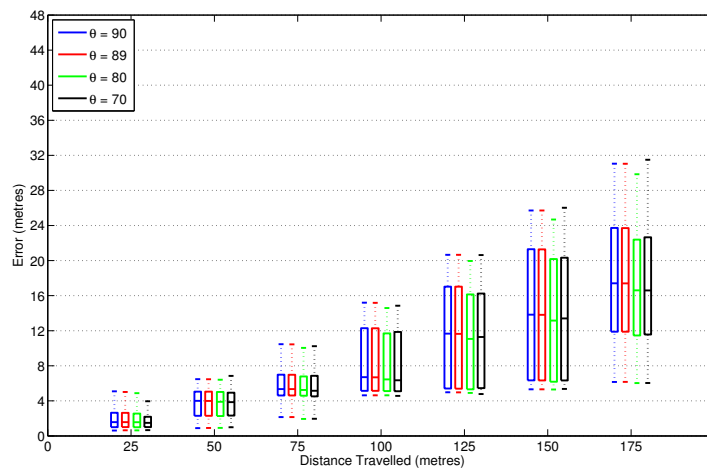


(c) Iterative Weighted DLT

Figure 5.7: Box plot of the visual odometry errors as a function of the distance travelled for the Hyperion sequence using a Euclidean ground plane constraint and fixed frame-rate. The results were found for the spherical, weighted and iterative weighted DLT's for the same 20 trials used to find the results in figure 5.6a.

(a) Points on sphere  $\eta, \eta'$ 

(b) Weighted DLT



(c) Iterative Weighted DLT

Figure 5.8: Box plot of the visual odometry errors as a function of the distance travelled for the Hyperion sequence using a Euclidean ground plane constraint and variable frame-rate. The results were found for the spherical, weighted and iterative weighted DLT's for the same 20 trials used to find the results in figure 5.6b.

DLT Mode	Threshold	Statistic	Distance Travelled (metres)							
			25	50	75	100	125	150	175	
Standard	$\theta < 90^\circ$	median	$\geq 500$	$\geq 500$	$\geq 500$	$\geq 500$	$\geq 500$	$\geq 500$	$\geq 500$	$\geq 500$
		IQR	$\geq 500$	$\geq 500$	$\geq 500$	$\geq 500$	$\geq 500$	$\geq 500$	$\geq 500$	$\geq 500$
	$\theta < 89^\circ$	median	3.78	5.99	8.27	10.82	13.44	16.09	17.97	
		IQR	2.68	2.88	3.00	4.12	4.80	4.31	3.03	
	$\theta < 80^\circ$	median	1.98	4.09	6.05	7.41	9.02	12.16	17.41	
		IQR	1.12	2.09	2.01	3.78	8.31	10.78	10.67	
	$\theta < 70^\circ$	median	1.99	3.87	5.31	6.93	7.95	9.31	12.57	
		IQR	1.12	2.29	1.91	1.86	4.25	6.08	6.32	
Spherical	$\theta < 90^\circ$	median	1.93	3.96	5.37	6.75	9.04	10.15	13.29	
		IQR	1.39	2.72	1.32	1.45	4.25	6.72	6.73	
	$\theta < 89^\circ$	median	1.93	3.96	5.37	6.75	9.04	10.15	13.29	
		IQR	1.39	2.72	1.32	1.45	4.25	6.72	6.73	
	$\theta < 80^\circ$	median	1.93	3.95	5.36	6.74	9.02	10.12	13.15	
		IQR	1.39	2.72	1.30	1.41	4.19	6.60	6.57	
	$\theta < 70^\circ$	median	1.93	3.91	5.31	6.70	8.76	9.93	12.84	
		IQR	1.40	2.71	1.31	1.31	3.88	6.23	6.33	
Weighted	$\theta < 90^\circ$	median	1.96	4.20	5.50	6.75	9.58	10.76	13.82	
		IQR	1.47	2.78	1.37	2.03	5.51	7.67	7.62	
	$\theta < 89^\circ$	median	1.96	4.20	5.49	6.75	9.56	10.74	13.83	
		IQR	1.47	2.78	1.38	2.03	5.51	7.68	7.64	
	$\theta < 80^\circ$	median	1.97	4.18	5.45	6.71	9.36	10.51	13.18	
		IQR	1.47	2.75	1.55	1.70	4.93	6.97	6.78	
	$\theta < 70^\circ$	median	1.96	4.10	5.40	6.61	9.00	10.15	12.17	
		IQR	1.48	2.78	1.22	1.42	3.91	5.78	5.54	
Iterative Weighted	$\theta < 90^\circ$	median	1.95	4.21	5.49	6.79	9.53	10.67	13.65	
		IQR	1.48	2.80	1.41	1.95	5.47	7.62	7.55	
	$\theta < 89^\circ$	median	1.95	4.19	5.48	6.79	9.52	10.65	13.65	
		IQR	1.47	2.78	1.42	1.95	5.46	7.63	7.56	
	$\theta < 80^\circ$	median	1.95	4.20	5.45	6.75	9.31	10.44	13.02	
		IQR	1.47	2.77	1.57	1.67	4.89	6.94	6.72	
	$\theta < 70^\circ$	median	1.95	4.13	5.40	6.62	9.02	10.17	12.12	
		IQR	1.49	2.77	1.28	1.45	3.95	5.81	5.55	

Table 5.2: Visual odometry errors (metres) as a function of the distance travelled for the Hyperion sequence using the Euclidean ground plane constraint, fixed frame-rate, and the spherical, weighted and iterative weighted DLT's. The blue entries are the smallest median values for the angular thresholds. IQR is the inter-quartile range.

## Discussion and Conclusions

**Fixed frame-rate:** From initial inspection of the results, the spherical, weighted and iterative weighted DLT's gives more consistent accuracy than the standard DLT with respect to the angular threshold  $\theta$  used. As was the case with the standard DLT, the results suggest that the most accurate visual odometry estimates are still found for the smallest angular threshold of  $\theta = 70^\circ$ . Although the iterative weighted DLT provides marginal improvements over the weighted DLT, neither appears to give more accurate

DLT Mode	Threshold	Statistic	Distance Travelled (metres)						
			25	50	75	100	125	150	175
Standard	$\theta < 90^\circ$	median	24.58	135.50	$\geq 500$	$\geq 500$	$\geq 500$	$\geq 500$	$\geq 500$
		IQR	392.02	$\geq 500$	$\geq 500$	$\geq 500$	$\geq 500$	$\geq 500$	$\geq 500$
	$\theta < 89^\circ$	median	3.27	5.15	6.97	8.70	10.62	12.07	14.16
		IQR	2.42	2.49	5.14	6.80	7.67	7.83	8.20
	$\theta < 80^\circ$	median	1.81	3.96	5.67	6.65	11.24	15.43	20.42
		IQR	1.00	1.98	2.45	7.57	12.72	15.25	13.51
	$\theta < 70^\circ$	median	1.87	4.28	6.09	6.49	10.82	14.94	20.00
		IQR	0.81	2.04	3.31	7.70	12.45	15.09	14.20
Spherical	$\theta < 90^\circ$	median	1.81	4.19	5.98	7.06	13.21	17.32	22.86
		IQR	0.78	1.87	3.08	8.27	13.81	16.90	14.82
	$\theta < 89^\circ$	median	1.81	4.19	5.98	7.06	13.21	17.32	22.86
		IQR	0.78	1.87	3.08	8.26	13.82	16.89	14.82
	$\theta < 80^\circ$	median	1.81	4.18	5.96	7.02	13.14	17.31	22.83
		IQR	0.77	1.87	3.06	8.23	13.70	16.76	14.60
	$\theta < 70^\circ$	median	1.88	4.31	6.21	7.85	14.50	19.72	25.83
		IQR	0.90	2.02	3.95	9.67	15.26	18.64	15.72
Weighted	$\theta < 90^\circ$	median	1.67	4.20	5.37	7.29	10.59	13.18	16.85
		IQR	1.31	2.23	1.69	5.93	10.53	12.68	11.05
	$\theta < 89^\circ$	median	1.67	4.21	5.38	7.26	10.56	13.19	16.87
		IQR	1.31	2.23	1.70	5.95	10.58	12.75	11.07
	$\theta < 80^\circ$	median	1.65	4.16	5.27	6.91	10.15	12.61	16.17
		IQR	1.32	2.26	1.54	5.44	9.90	12.03	10.19
	$\theta < 70^\circ$	median	1.74	4.14	5.80	6.84	10.91	14.29	18.25
		IQR	1.01	1.93	2.78	6.94	11.89	14.83	13.01
Iterative Weighted	$\theta < 90^\circ$	median	1.62	4.03	5.21	7.62	10.91	13.67	17.38
		IQR	1.46	2.54	1.78	5.73	10.61	12.84	11.43
	$\theta < 89^\circ$	median	1.62	4.03	5.19	7.59	10.88	13.66	17.37
		IQR	1.46	2.53	1.78	5.73	10.63	12.87	11.43
	$\theta < 80^\circ$	median	1.60	4.01	5.13	7.34	10.48	13.01	16.56
		IQR	1.49	2.56	1.68	5.22	9.92	12.04	10.57
	$\theta < 70^\circ$	median	1.58	3.96	5.06	7.11	10.64	13.35	16.57
		IQR	1.28	2.42	1.82	5.32	9.73	12.18	10.63

Table 5.3: Visual odometry errors (metres) as a function of the distance travelled for the Hyperion sequence using the Euclidean ground plane constraint, variable frame-rate, and the spherical, weighted and iterative weighted DLT's. The blue entries are the smallest median values for the angular thresholds. IQR is the inter-quartile range.

results than the spherical DLT. However, all have very similar accuracy. Note also that the accuracy of the visual odometry estimates using the spherical DLT for angular thresholds  $\theta < 90^\circ$  and  $\theta < 89^\circ$  are comparable to the accuracy using the standard DLT with the smallest angular threshold of  $\theta < 70^\circ$ . These results suggest that using the most basic angular threshold  $\theta < 90^\circ$  (i.e. points can lie anywhere in the ground plane), the spherical, weighted and iterative weighted DLT's are more suited than the standard DLT for estimating the homography between frames using the Euclidean ground plane constraint and fixed frame-rate for the Hyperion sequence.

It is of interest to note here that there are minimal differences in the results found

using the weighted DLT and the iterative weighted DLT. This is not surprising as the magnitude of the camera translation  $\mathbf{t}$  between the views used to compute the camera egomotion is very small using the fixed frame-rate. As a result, the magnitude of  $H\eta_i$  is dependent mainly on the rotational component of the homography  $H$  such that  $\|H\eta_i\| \approx \text{constant} \forall \eta_i$ ; this is the condition that the iterative weighted DLT attempts to achieve.

**Variable frame-rate:** As was the case with the fixed frame-rate, when compared to the results obtained using the standard DLT, the accuracy of the visual odometry estimates obtained using the spherical, weighted and iterative weighted DLT's are less influenced by the angular threshold used. In general, both the weighted and iterative weighted DLT's show some improvement over the spherical DLT. Unlike the fixed frame-rate, the magnitude of the translation  $\mathbf{t}$  between the frames used to estimate the camera egomotion is greater resulting in an increased variability in the magnitude of  $H\eta_i$  for each  $\eta_i$ . This suggests that the iterative weighted DLT would give improved performance over the weighted DLT. This improvement in performance is observed in the results, particularly for the smallest angular threshold of  $\theta = 70^\circ$ .

#### 5.2.6.4 Iterative Refinement

After obtaining a linear estimate of the homography  $H$  between frames from which an estimate of the camera egomotion is obtained, the accuracy of egomotion estimate can be improved by minimising some new cost function (different to the cost function used to find the linear estimate) using a non-linear optimisation. A standard cost function to be minimised is the *transfer* error  $\varepsilon$  between images defined as [95]:

$$\varepsilon = \sum_{i=1}^n d(\mathbf{u}'_i, \mathbf{u}'_i(H\eta_i))^2 + d(\mathbf{u}_i(H^{-1}\eta'_i), \mathbf{u}_i)^2, \quad (5.37)$$

where  $n$  is the number of correspondences. The parameters  $\mathbf{u}_i$  and  $\mathbf{u}'_i$  are the coordinates of the pSIFT keypoints in the first and second stereographic image respectively. The parameter  $\mathbf{u}'_i(H\eta_i)$  is the coordinate, in the second stereographic image, of the point  $H\eta_i$  mapped to this image. Likewise,  $\mathbf{u}_i(H^{-1}\eta'_i)$  is the coordinate, in the first stereographic image, of the point  $H^{-1}\eta'_i$  mapped to this image.  $d(\mathbf{u}'_i, \mathbf{u}'_i(H\eta_i))$  is the Euclidean distance between the points  $\mathbf{u}'_i$  and  $\mathbf{u}'_i(H\eta_i)$  measured on the stereographic image plane, and  $d(\mathbf{u}_i(H^{-1}\eta'_i), \mathbf{u}_i)$  is the Euclidean distance between the points  $\mathbf{u}_i(H^{-1}\eta'_i)$  and  $\mathbf{u}_i$  measured on the stereographic image plane.



A *geometric* cost function can also be used for the Euclidean ground plane constraint based on the method used by Maimone, Cheng and Matthies [145]. In contrast to the transfer error which is defined on the image plane, the geometric error is defined with respect to the position of the world points  $\mathbf{X}, \mathbf{X}'$  in Euclidean space. Although Maimone, Cheng and Matthies use their method to estimate 6 degree of freedom camera motion using stereo vision, the method can be adapted for used with the Euclidean ground plane constraint. As all world points are constrained to lie in the ground plane, the camera motion between frames can be restored to 2 degrees of freedom where

$$R_z(\phi) = \begin{bmatrix} \cos \phi & -\sin \phi \\ \sin \phi & \cos \phi \end{bmatrix}, \quad \mathbf{t} = \begin{bmatrix} t_x \\ t_y \end{bmatrix}. \quad (5.38)$$

Using the two-dimensional coordinates  $\mathbf{X}_i = (X_i, Y_i)^T$  and  $\mathbf{X}'_i = (X'_i, Y'_i)^T$  of the pair of corresponding keypoints obtained from equation 5.17, the geometric error for the pair of corresponding keypoints is

$$\boldsymbol{\varepsilon}_i = \mathbf{X}'_i - (R_z \mathbf{X}_i + \mathbf{t}). \quad (5.39)$$

For the set of all correspondences  $\mathbf{X} \leftrightarrow \mathbf{X}'$ , the maximum likelihood estimate of the camera motion defined by the variables  $\phi, t_x$  and  $t_y$  minimises the sum

$$\sum_i (\boldsymbol{\varepsilon}_i^T W_i \boldsymbol{\varepsilon}_i), \quad (5.40)$$

where  $W_i$  is the inverse covariance matrix for the error  $\boldsymbol{\varepsilon}_i$  defined as

$$W_i = (R_z \Sigma_{\mathbf{X}'_i} R_z^T + \Sigma_{\mathbf{X}_i})^{-1}. \quad (5.41)$$

The parameters  $\Sigma_{\mathbf{X}'_i}$  and  $\Sigma_{\mathbf{X}_i}$  are the  $2 \times 2$  covariance matrices of the points  $\mathbf{X}_i$  and  $\mathbf{X}'_i$  respectively. Given the coordinates  $\mathbf{x}_i$  and  $\mathbf{x}'_i$  on the stereographic image (relative to the principal point) of two corresponding pSIFT keypoints, the covariance matrix for each is obtained from equation 5.2.6.5 — the uncertainty in the positions of all keypoints on the stereographic image are assumed to be equal.

### 5.2.6.5 Experiments: Iterative Refinement

The experiments in this section compare the accuracy of the optimised visual odometry estimates for the Hyperion sequence using the Euclidean ground plane constraint and the transfer and geometric cost functions. For the fixed and variable frame rates, an initial estimate of the Homography  $H$  between frames was found using the iterative

weighted DLT from with the variables  $\phi, t_x$  and  $t_y$  were recovered. The threshold angle used was  $\theta < 89^\circ$ . Using this estimate of  $H$  obtained with the iterative weighted DLT, any correspondences with a transfer error  $\varepsilon_i > 4$  were considered outliers and removed. Using these remaining correspondences, the variables  $\phi, t_x$  and  $t_y$  were optimised using a Matlab implementation of Levenberg-Marquardt and each of the transfer and geometric cost functions defined in equations 5.37 and 5.40. Optimising the variables  $\phi, t_x$  and  $t_y$  using the transfer cost function ensures that the homography contains no affine transform. The parameters  $\phi, t_x$  and  $t_y$ , and the height  $h_p$  of the camera from the ground plane are used to find the homography in equation 5.16 for each iteration of the optimisation. For each cost function, optimisation is terminated if the tolerance on the rotation or translation values changes by less than  $10e^{-6}$ , or if the number of iterations exceeds 200.

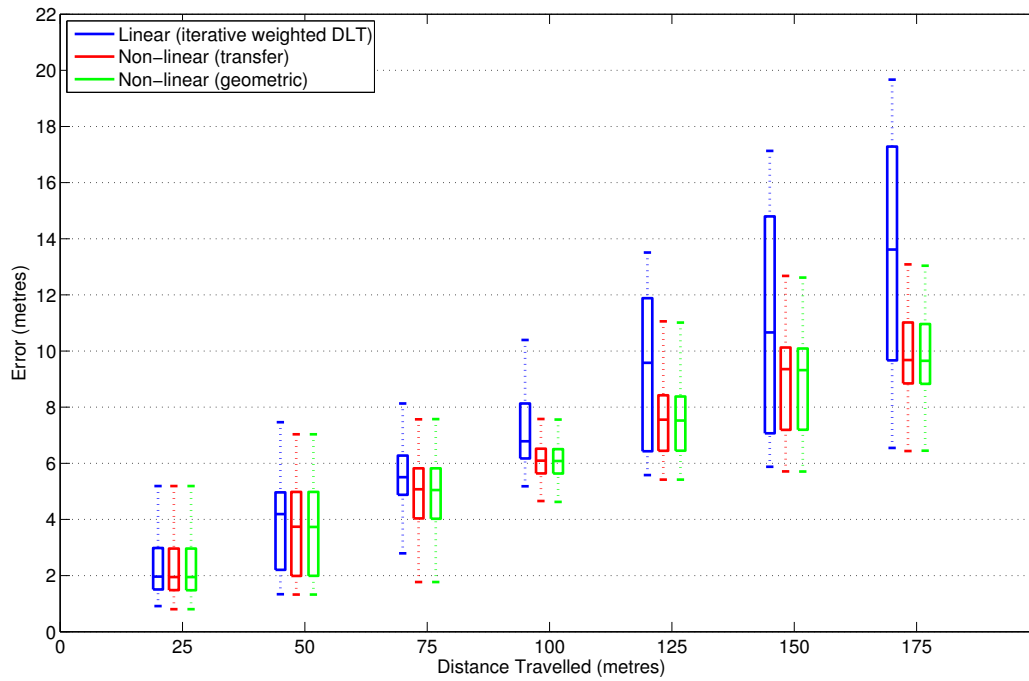
## Results

The results are shown in figure 5.9 accumulated over a total of 20 trials. This is a separate set of 20 trials to those used in the previous experiments. The visual odometry errors for the initial linear estimates of the camera egomotion obtained using the iterative weighted DLT have been included for comparison. A summary of the results is given in table 5.4. Figure 5.10 illustrates for one of the trials the visual odometry estimates versus GPS ground truth for the linear estimated (iterative weighted DLT), and each of the non-linear methods using the transfer error and geometric error.

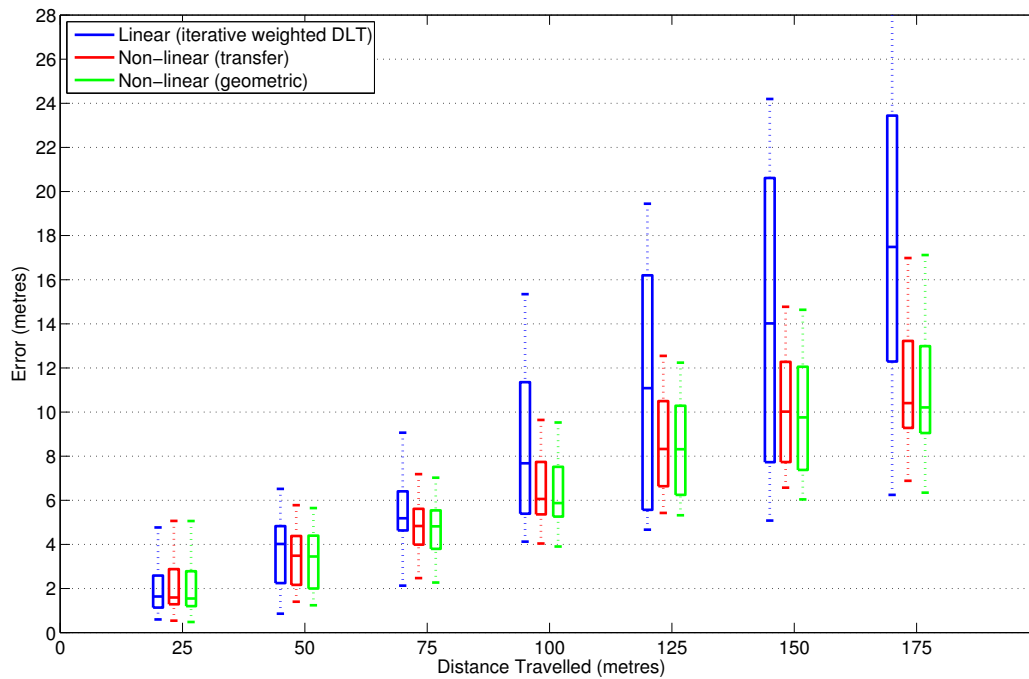
## Discussion and Conclusions

The results for each frame-rate indicate improvements in accuracy over the linear estimates for both non-linear schemes (transfer and geometric cost functions). For the fixed frame-rate, accuracy of the results using the transfer error are marginally better than those for the geometric error, however, the opposite is true for the variable frame-rate.

The similarities in the accuracy using the two cost modes can be explained by the fact that each performs a similar operation. Each attempts to find the maximum likelihood estimate of the camera egomotion which accounts for the uncertainty of keypoint positions in the stereographic image. The transfer cost function accounts for this uncertainty by defining the error to be minimised on the stereographic image. The geometric cost function defines the error to be minimised in Euclidean with the covariance matrices used to account for the uncertainty in keypoint positions in the stereographic image. These covariance matrices were obtained from equation

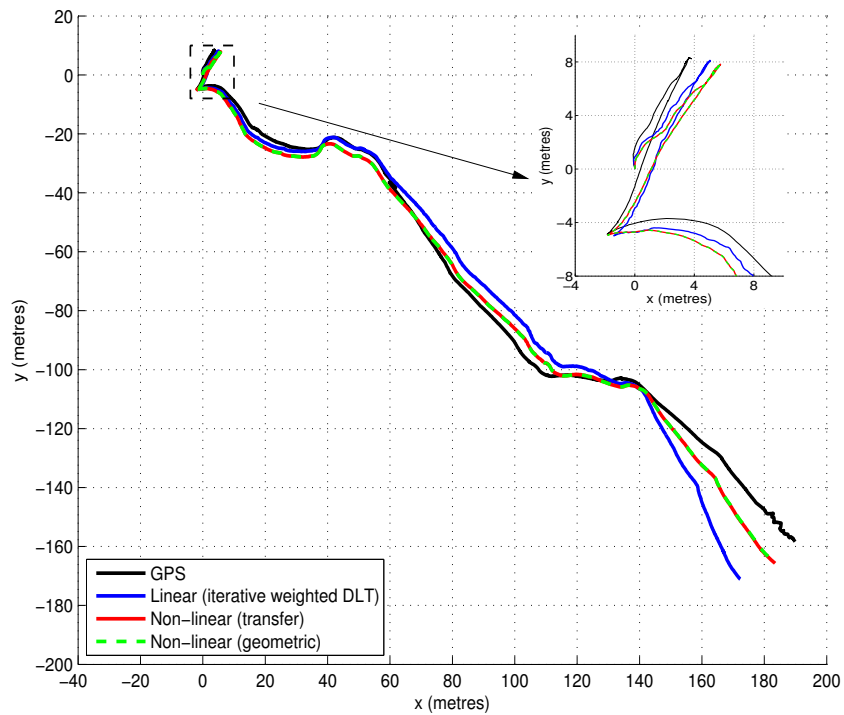


(a) Fixed frame-rate

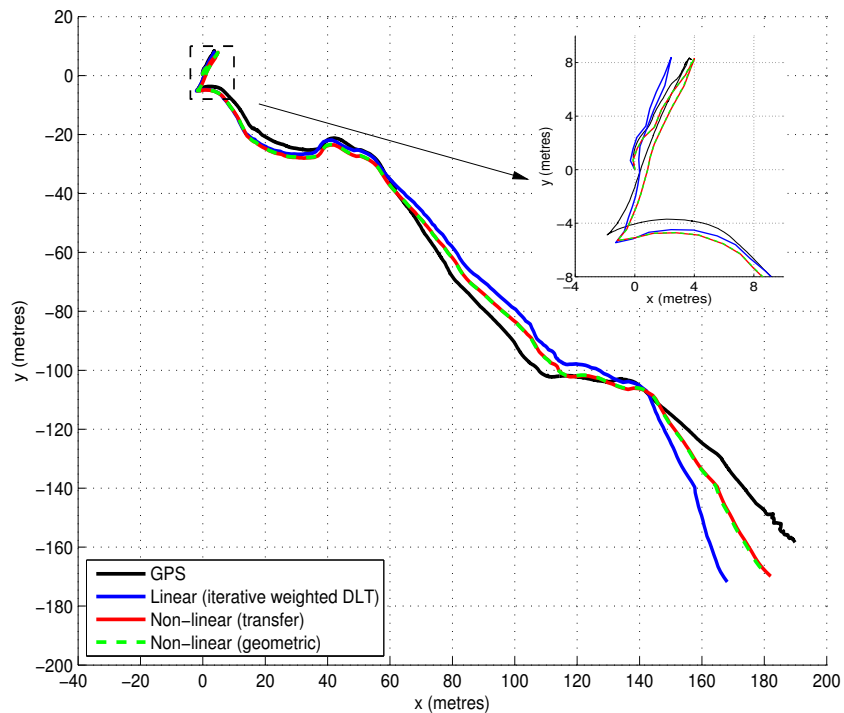


(b) Variable frame-rate

Figure 5.9: Box plot of the visual odometry error as a function of the distance travelled for the Hyperion sequence using a linear estimate of the camera motion followed by iterative refinement for the Euclidean ground plane constraint. Results are for a separate set of 20 trials to those used previously to find the results in figures 5.6a and 5.6b. The visual odometry errors using the initial linear estimate have been included for comparison.



(a) Fixed frame-rate



(b) Variable frame-rate

Figure 5.10: Plot of the visual odometry estimates for the Euclidean ground plane constraint versus GPS ground truth. The linear iterative weighted DLT was used to obtain an initial estimate of the camera egomotion between frames. This estimate was then optimised using the transfer and geometric cost functions. The paths have been manually aligned as no orientation information of the vehicle was available.

Mode	Statistic	Distance Travelled (metres)						
		25	50	75	100	125	150	175
Linear (iterative weighted DLT)	median	1.97	4.19	5.50	6.78	9.58	10.66	13.62
	IQR	1.47	2.76	1.40	1.96	5.45	7.73	7.62
Non-Linear (transfer)	median	1.95	3.74	5.08	6.09	7.55	9.36	9.68
	IQR	1.48	2.99	1.79	0.88	1.98	2.94	2.17
Non-Linear (geometric)	median	1.95	3.73	5.05	6.08	7.52	9.32	9.65
	IQR	1.48	2.99	1.79	0.87	1.93	2.90	2.13

(a) Fixed frame-rate.

Mode	Statistic	Distance Travelled (metres)						
		25	50	75	100	125	150	175
Linear (iterative weighted DLT)	median	1.64	4.02	5.18	7.68	11.08	14.02	17.49
	IQR	1.45	2.58	1.78	5.97	10.63	12.89	11.15
Non-Linear (transfer)	median	1.59	3.49	4.83	6.06	8.32	10.02	10.40
	IQR	1.59	2.22	1.63	2.37	3.86	4.54	3.94
Non-Linear (geometric)	median	1.54	3.45	4.82	5.87	8.32	9.76	10.21
	IQR	1.58	2.40	1.74	2.25	4.04	4.68	3.94

(b) Variable frame-rate.

Table 5.4: Error (metres) as a function of the distance travelled for the Hyperion sequence using the Euclidean ground plane constraint. An initial estimate of the camera egomotion was obtained using the iterative weighted DLT. This accuracy of this estimate was then optimised using the transfer and geometric cost functions. The blue entries are the smallest median values for all methods. IQR is the inter-quartile range.

The results suggest that either the transfer or geometric cost functions are suitable for optimising the estimate of the camera egomotion. For each iteration of optimisation, the transfer error is more computationally efficient to compute than the geometric error. This makes it potentially the more practical method to use.

### 5.2.7 Ground Plane Visual Odometry: Triggs' Method

In many practical applications the precise normal to the ground plane  $\mathbf{n}_p$  may be unknown. When this is the case, the homography between two frames is of the generalised form given in equation 5.6 which has eight degrees of freedom and can be solved up to arbitrary scale factor. The experiments in this section find the visual odometry estimates by solving for the generalised homography between frames. The components of the rotation  $R$ , translation  $\mathbf{t}$  and the normal  $\mathbf{n}$  are extracted from the homography  $H$  using the method devised by Triggs [226]. The visual odometry estimates obtained using this method will be referred to as the Triggs ground plane constraint, which is the same terminology used by Scaramuzza [198, 200].

The generalised homography can be solved by constructing the set of simultaneous

linear equations  $A\mathbf{h} = \mathbf{0}$ , where each keypoint correspondence contributes two rows to the matrix  $A$  — these are the first two rows in equation 5.12. A solution for the entries  $\mathbf{h}$  of the homography  $H$  can be obtained from a minimum of four non-collinear correspondences from the singular value decomposition  $USV = \text{svd}(A)$  of  $A$ .  $\mathbf{h}$  is the column vector of  $V$  corresponding to the smallest non-zero singular value. When researching methods of camera calibration using planar scenes [226], Triggs formulated a method for extracting, up to a two-fold ambiguity, the rotation  $R$ , translation  $\mathbf{t}/h_p$  and normal  $\mathbf{n}_p$  from the generalised homography  $H$  defined equation 5.6, where the known height  $h_p$  of the camera from the ground plane can be used to find  $\mathbf{t}$ . These are found from the singular value decomposition of  $H$ , and the full details can be found in [226]. The two-fold ambiguity is resolved by selecting the solution whose normal  $n_p$  is nearest to the approximately known value of  $n_p = (0, 0, 1)^T$ .

As just discussed, the homography  $H$  can be solved from the matrix  $A$  which uses the homogeneous coordinates  $\mathbf{x}, \mathbf{x}'$ . The coordinates  $\mathbf{X}, \mathbf{X}'$  can not be used as the normal to the ground plane needs to be known precisely to derive them from the coordinates of the keypoints in the stereographic image. Given the coordinates  $\eta, \eta'$  of the corresponding keypoints on the sphere, the ‘standard’ way to construct the matrix is to use the normalised homogeneous coordinates  $\check{\mathbf{x}}_i = (\eta(x)_i/\eta(z)_i, \eta(y)_i/\eta(z)_i, 1)$ ,  $\check{\mathbf{x}}'_i = (\eta(x)'_i/\eta(z)'_i, \eta(y)'_i/\eta(z)'_i, 1)$  and apply the normalisation described in [95] to improve numerical stability when solving for  $\mathbf{h}$ . This method was used by Scaramuzza and Siegwart [200] to estimate camera egomotion using the Triggs ground plane constraint with keypoints detected in wide-angle catadioptric images. The solution obtained using this method will be referred to as the *standard DLT*.

As was the case for the Euclidean ground plane constraint, the spherical, weighted and iterative weighted DLT’s can be used to find a solution for  $\mathbf{h}$ . The only difference is that the matrix  $A$  is of the form given in equation 5.34 and not 5.35 (i.e. there are no redundant terms in  $\mathbf{h}$ ). The weighting factors given in equation are used by the weighted and iterative weighted DLT’s. The solution for  $\mathbf{h}$  using the spherical, weighted and iterative weighted DLT’s is again obtained from the singular value decomposition of  $\mathbf{A}$ . No data normalisation similar to that described in [95] is used by the spherical, weighted and iterative weighted DLT’s.

As a final note, the estimate of the camera egomotion returned using the Triggs ground plane constraint is defined in the camera coordinate frame of reference. The corresponding change in angle  $\phi$  and translation on the ground plane can be found by projecting  $R$  and  $\mathbf{t}$  to the ground plane using  $\mathbf{n}_p$ . This is done in the following experiments to find the motion of the camera in the ground plane.

### 5.2.7.1 Experiments: Linear Estimate

For each of the fixed and variable frame-rates, the visual odometry estimates for the Hyperion sequence were obtained using the Triggs ground plane constraint for 20 separate trials. These were the same trials used in the experiments in section 5.2.6.3. These visual odometry estimates were found using each of the standard, spherical, weighted and iterative weighted DLT's and the same angular thresholds  $\theta$  used in the previous experiments.

#### Results

The results for the fixed frame rate are displayed in figure 5.11 and summarised in table 5.5. The results for the fixed frame rate are displayed in figure 5.12 and are summarised in tables 5.5 and 5.6.

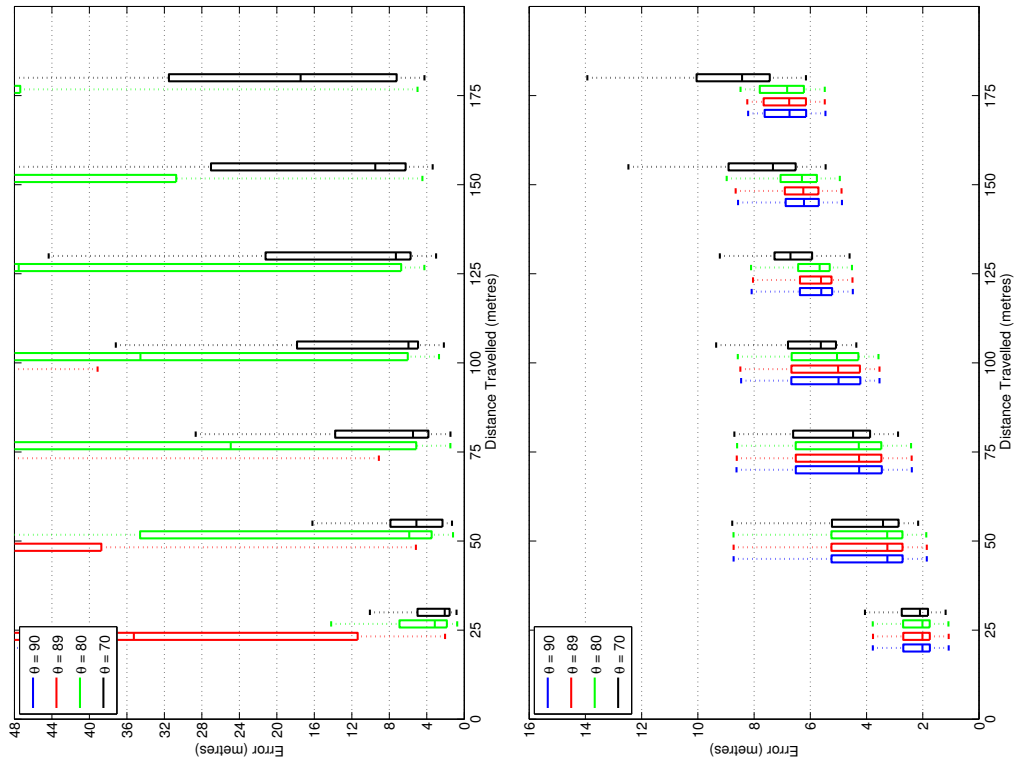
#### Discussion and Conclusions

As for the Euclidean ground plane constraint, the accuracy of the visual odometry results found using the standard DLT for the fixed and variable frame-rates is sensitive to the angular threshold used. In contrast, the accuracy of the visual odometry estimates using the spherical, weighted and iterative weighted DLT's are less influenced by the angular threshold used.

For both the fixed and variable frame-rates, in general the weighted DLT gives improved accuracy over the spherical DLT. With respect to the iterative weighted DLT, for both frame-rates improvements in the accuracy compared to the weighted DLT are observed, although these are more substantial for the variable frame-rate. As argued previously, the accuracy of the results found using the iterative weighted DLT are only marginally better than those found using the weighted DLT for the fixed frame-rate as that the magnitude of the translation between frames is small. As a result, the magnitude of  $H\eta_i$  is near constant for all  $\eta_i$  which is the condition that the iterative weighted DLT aims to achieve.

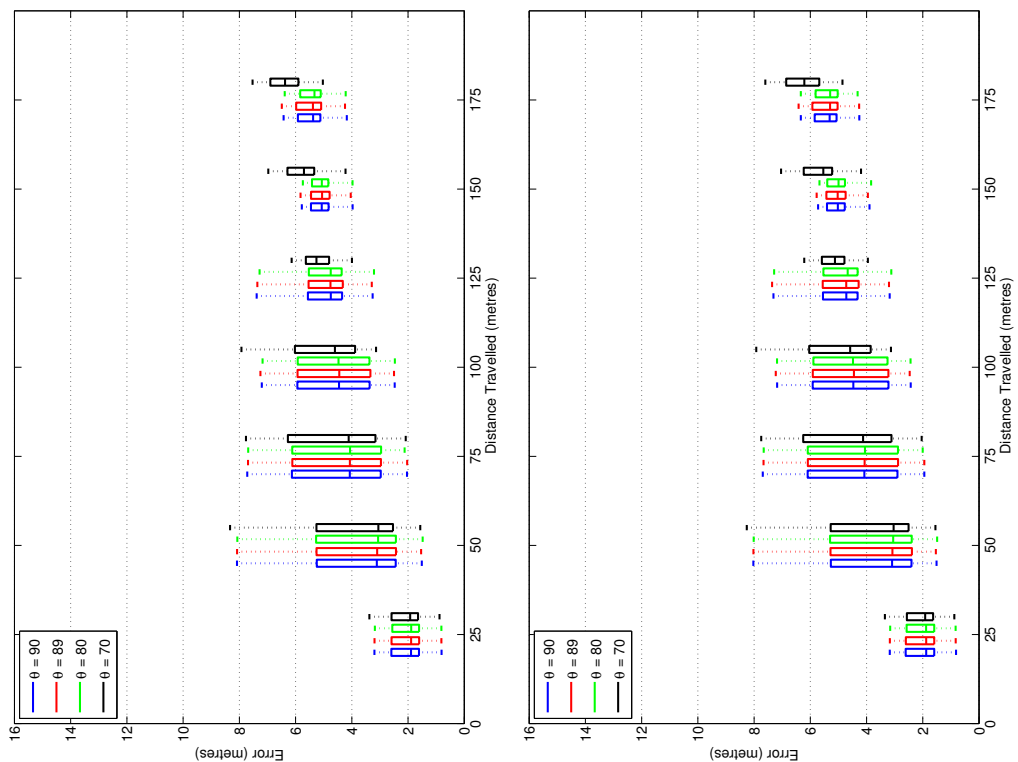
For the fixed frame-rate, the iterative weighted DLT gives improved performance over the standard DLT for all threshold values. The same is true for the variable frame-rate except for case where the threshold is set to  $\theta = 80^\circ$  using the standard DLT. However, as shown by the results in figures 5.11 and 5.12, the iterative weighted DLT is more robust as the performance remains more consistent for all threshold angles  $\theta$ .





(a) Standard DLT.

(b) spherical DLT.



(c) Weighted DLT.

(d) Iterative Weighted DLT.

Figure 5.11: Box plots of the visual odometry errors using a fixed frame-rate (Hyperion, Triggs ground plane constraint). The results were obtained using the same 20 trials as the results in figure 5.6a.

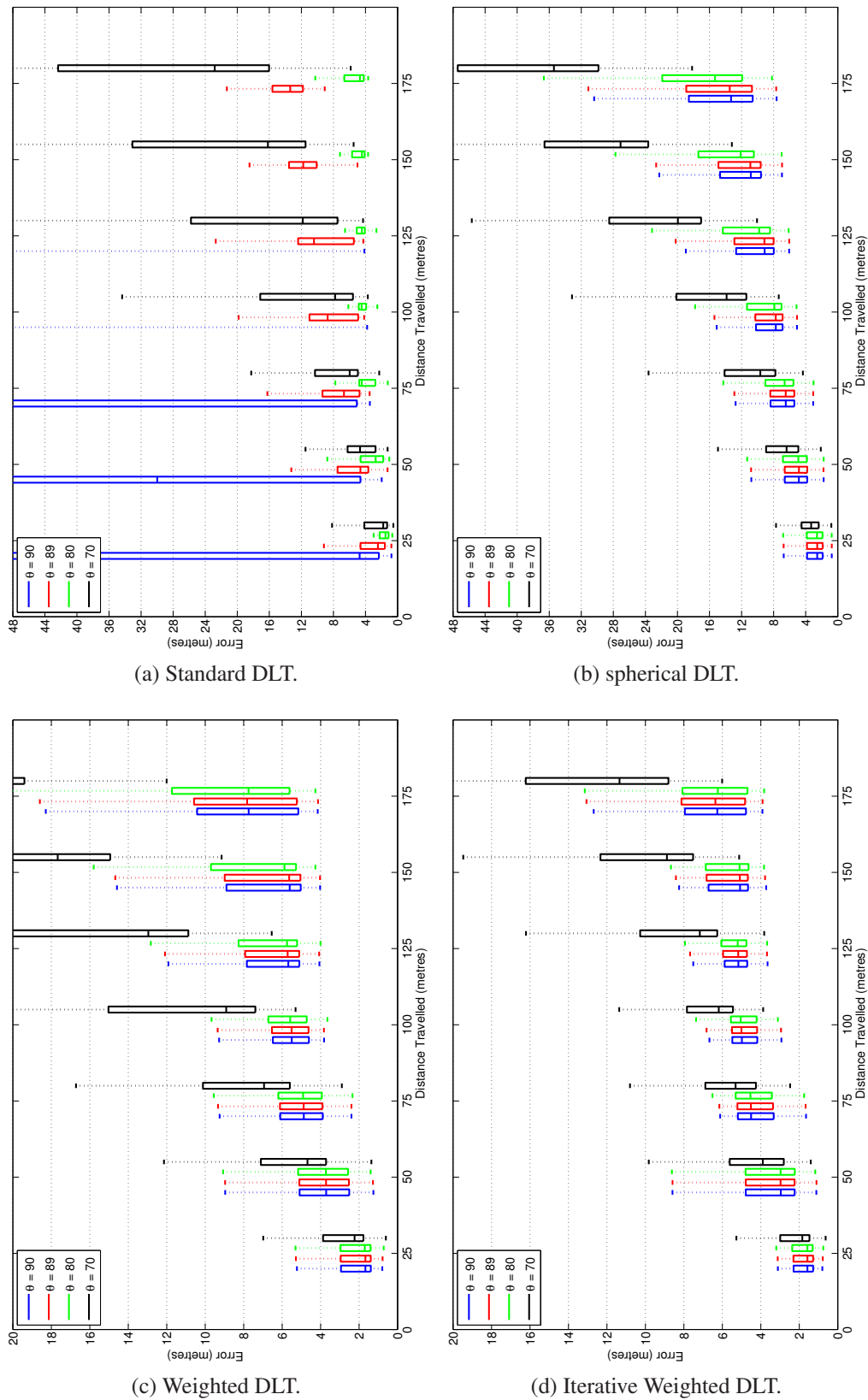


Figure 5.12: Box plots of the visual odometry errors using a variable frame-rate (Hyperion, Triggs ground plane constraint). The results were obtained using the same 20 trials as the results in figure 5.6b.

DLT Mode	Threshold	Statistic	Distance Travelled (metres)						
			25	50	75	100	125	150	175
Standard	$\theta < 90^\circ$	median	$\geq 999$	$\geq 999$	$\geq 999$	$\geq 999$	$\geq 999$	$\geq 999$	$\geq 999$
		IQR	$\geq 999$	$\geq 999$	$\geq 999$	$\geq 999$	$\geq 999$	$\geq 999$	$\geq 999$
	$\theta < 89^\circ$	median	35.26	67.11	110.78	157.09	186.41	213.75	238.84
		IQR	58.91	77.87	99.32	113.28	104.72	96.46	81.17
	$\theta < 80^\circ$	median	3.15	5.88	24.92	34.55	47.54	57.72	69.21
		IQR	5.05	31.11	56.96	64.07	67.84	50.21	46.82
	$\theta < 70^\circ$	median	<b>2.09</b>	<b>5.11</b>	<b>5.47</b>	<b>5.95</b>	<b>7.30</b>	<b>9.49</b>	<b>17.46</b>
		IQR	3.43	5.56	9.94	12.92	15.48	20.76	24.30
Spherical	$\theta < 90^\circ$	median	<b>2.01</b>	<b>3.26</b>	<b>4.26</b>	<b>5.00</b>	<b>5.61</b>	<b>6.22</b>	<b>6.74</b>
		IQR	0.95	2.53	3.06	2.45	1.15	1.18	1.48
	$\theta < 89^\circ$	median	<b>2.01</b>	<b>3.26</b>	4.27	5.01	5.62	6.25	6.75
		IQR	0.94	2.53	3.04	2.44	1.12	1.20	1.51
	$\theta < 80^\circ$	median	<b>2.01</b>	<b>3.26</b>	4.28	5.05	5.67	6.30	6.82
		IQR	0.95	2.52	3.04	2.37	1.12	1.29	1.57
	$\theta < 70^\circ$	median	2.10	3.42	4.48	5.62	6.70	7.33	8.43
		IQR	0.93	2.37	2.74	1.71	1.34	2.39	2.60
Weighted	$\theta < 90^\circ$	median	<b>1.89</b>	3.11	4.07	4.46	<b>4.75</b>	5.07	5.38
		IQR	0.98	2.81	3.16	2.57	1.22	0.63	0.80
	$\theta < 89^\circ$	median	<b>1.89</b>	3.10	4.07	<b>4.44</b>	4.76	<b>5.06</b>	5.38
		IQR	0.98	2.83	3.15	2.59	1.22	0.66	0.89
	$\theta < 80^\circ$	median	<b>1.89</b>	3.07	<b>4.06</b>	4.47	<b>4.75</b>	<b>5.06</b>	<b>5.33</b>
		IQR	0.95	2.84	3.16	2.55	1.17	0.58	0.73
	$\theta < 70^\circ$	median	1.92	<b>3.06</b>	4.11	4.60	5.26	5.70	6.37
		IQR	0.94	2.72	3.12	2.14	0.83	0.95	1.00
Iterative Weighted	$\theta < 90^\circ$	median	<b>1.88</b>	3.09	4.08	4.47	4.72	5.02	5.31
		IQR	1.01	2.87	3.19	2.69	1.23	0.63	0.78
	$\theta < 89^\circ$	median	<b>1.88</b>	3.08	4.07	<b>4.45</b>	4.72	5.03	5.30
		IQR	1.01	2.90	3.21	2.69	1.27	0.68	0.90
	$\theta < 80^\circ$	median	<b>1.88</b>	3.05	<b>4.06</b>	4.48	<b>4.67</b>	<b>5.00</b>	<b>5.29</b>
		IQR	0.98	2.90	3.21	2.64	1.22	0.63	0.79
	$\theta < 70^\circ$	median	1.91	<b>3.04</b>	4.12	4.58	5.12	5.54	6.21
		IQR	0.93	2.77	3.14	2.19	0.81	1.01	1.18

Table 5.5: Visual odometry errors using fixed frame-rate (Hyperion, Triggs ground plane constraint) for the standard, spherical, weighted and iterative weighted DLT's (see figure 5.11). All values have units of metres. The blue entries indicate the smallest median value for each DLT across all angular thresholds. IQR is the inter-quartile range.

### 5.2.7.2 Iterative Refinement

After obtaining a linear solution for the homography  $H$ , the accuracy of  $H$  can potentially be improved using a non-linear optimisation. The transfer error defined in equation 5.37 can again be used as the cost function to be minimised. It is possible, given the normal to the plane  $\mathbf{n}_p$  and the estimate of the inter-frame rotation  $R$ , to resolve the position of points  $\mathbf{X}, \mathbf{X}'$  and their associated covariance matrices  $\sigma_{\mathbf{X}}, \sigma_{\mathbf{X}'}$  directly from the keypoint positions on image. However, these would need to be recalculated

DLT Mode	Threshold	Statistic	Distance Travelled (metres)						
			25	50	75	100	125	150	175
Standard	$\theta < 90^\circ$	median	5.12	27.88	120.00	492.48	684.47	753.11	769.85
		IQR	60.20	159.17	611.14	848.11	$\geq 999$	$\geq 999$	$\geq 999$
	$\theta < 89^\circ$	median	2.43	4.78	7.27	9.00	11.30	11.76	12.31
		IQR	3.08	3.18	4.06	6.66	6.05	4.93	5.18
	$\theta < 80^\circ$	median	1.55	2.91	4.02	4.02	4.09	4.65	5.25
		IQR	0.85	2.68	1.98	0.95	0.73	1.40	2.21
	$\theta < 70^\circ$	median	1.75	4.57	5.87	7.82	11.45	15.34	22.27
		IQR	2.12	3.35	4.42	9.50	14.84	18.47	21.91
Spherical	$\theta < 90^\circ$	median	2.38	4.21	5.85	6.76	8.41	9.91	12.36
		IQR	1.67	2.59	2.66	2.75	3.92	4.33	5.23
	$\theta < 89^\circ$	median	2.38	4.22	5.85	6.77	8.45	9.93	12.46
		IQR	1.68	2.63	2.71	2.80	4.09	4.50	5.44
	$\theta < 80^\circ$	median	2.40	4.30	5.98	6.98	8.97	10.84	14.18
		IQR	1.67	2.81	2.84	3.41	5.20	6.03	7.25
	$\theta < 70^\circ$	median	2.99	5.78	8.75	12.86	18.32	24.80	33.67
		IQR	1.91	3.41	4.39	7.23	9.71	11.71	14.26
Weighted	$\theta < 90^\circ$	median	1.70	3.18	4.36	4.87	5.28	6.20	7.15
		IQR	1.31	2.37	2.04	1.09	2.19	3.37	3.88
	$\theta < 89^\circ$	median	1.71	3.19	4.38	4.89	5.33	6.24	7.20
		IQR	1.32	2.38	2.03	1.09	2.27	3.45	3.95
	$\theta < 80^\circ$	median	1.70	3.21	4.50	4.88	5.24	5.99	7.43
		IQR	1.33	2.44	2.13	1.22	2.76	3.71	4.02
	$\theta < 70^\circ$	median	2.09	4.24	6.34	8.41	11.92	16.23	23.13
		IQR	1.84	3.23	3.22	6.75	9.11	10.81	13.79
Iterative Weighted	$\theta < 90^\circ$	median	1.60	2.96	3.91	4.32	4.80	5.67	6.51
		IQR	0.86	2.54	2.09	0.99	1.06	2.23	2.48
	$\theta < 89^\circ$	median	1.60	2.97	3.92	4.36	4.85	5.73	6.57
		IQR	0.86	2.53	2.09	0.97	1.11	2.29	2.52
	$\theta < 80^\circ$	median	1.59	2.92	3.95	4.43	4.84	5.70	6.49
		IQR	0.85	2.52	2.04	0.94	1.18	2.36	2.73
	$\theta < 70^\circ$	median	1.77	3.50	4.94	5.30	6.63	8.24	10.51
		IQR	1.22	2.61	2.30	2.24	3.48	4.09	5.74

Table 5.6: Visual odometry errors using variable frame-rate (Hyperion, Triggs ground plane constraint) for the standard, spherical, weighted and iterative weighted DLT's (see figure 5.12). All values have units of metres. The blue entries indicate the smallest median value for each DLT across all angular thresholds. IQR is the inter-quartile range.

for each iteration of the optimisation using the estimate of  $H$  obtained in the previous iteration. As a result, the computational expense using the geometric error far exceeds that of using the transfer error. Considering also that the accuracy of the results was very similar for transfer and geometric cost functions for the Euclidean ground plane constraint, only the transfer error is considered in this section.

### 5.2.7.3 Experiments: Iterative Refinement

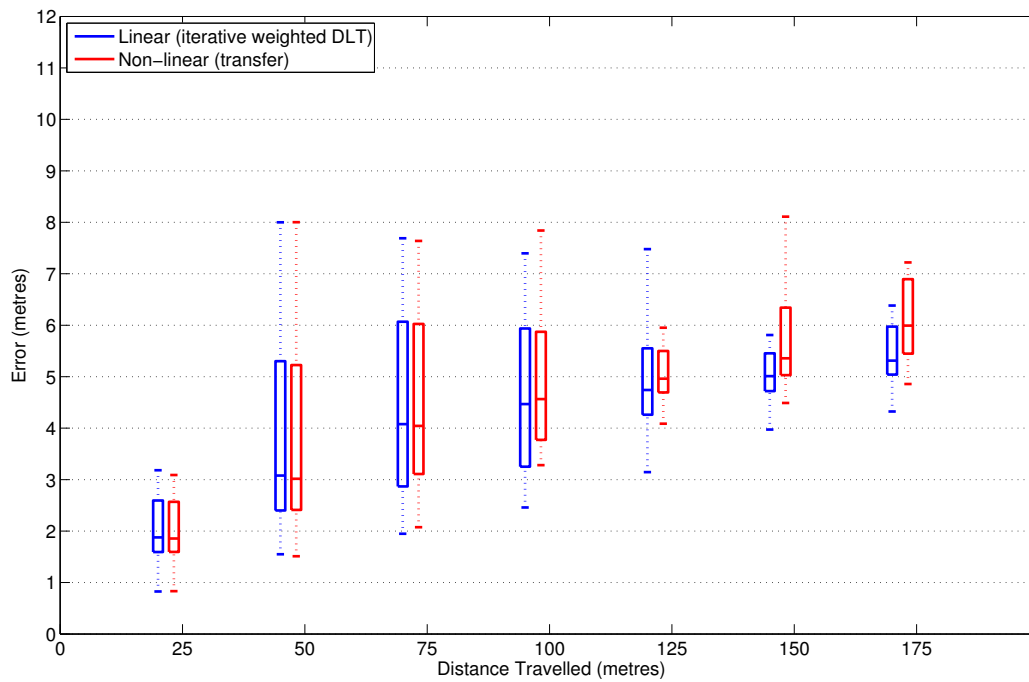
Using either the fixed or variable frame-rate, an initial estimate of the homography  $H$  was found using the iterative weighted DLT with angular threshold  $\theta < 89^\circ$ . The normal to the plane  $n_p$  and rotation were recovered from this estimate and used to remove any of the correspondences that do not project to a points in the ground plane. This estimate of  $H$  was then used to find the transfer error  $\sigma_i$  defined in equation 5.37 for each of the remaining correspondences. Any correspondences with an error  $\varepsilon_i > 4$  were considered outliers and were removed. The individual elements of the homography  $H$  were then optimised, which is the method recommended in [95], by minimising the transfer error cost function in equation 5.37. A Matlab implementation of Levenberg-Marquardt was used to perform this optimisation. The optimisation terminates if the tolerance of all the elements of  $H$  change by less than  $10e^{-6}$ , or if a number of iterations exceeds 200.

#### Results

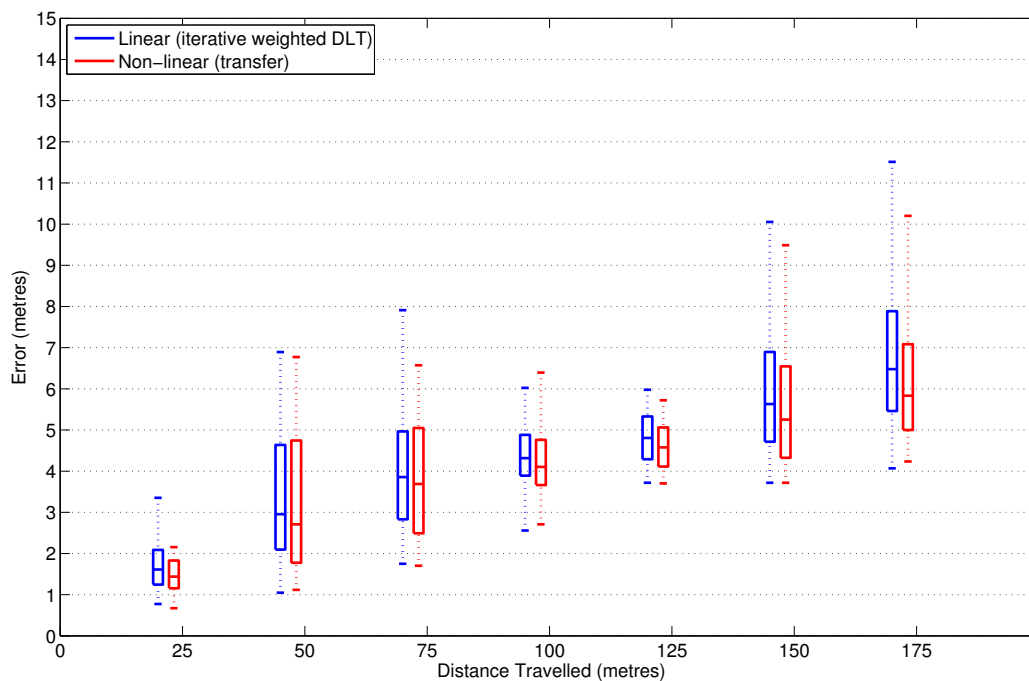
Results were obtained using the same 20 trials used previously in section 5.2.6.5. They are presented in figure 5.13 and summarised in table 5.7. Figure 5.14 shows, for one of the fixed frame-rate and variable frame-rate trials, the estimates of the visual odometry found using simply the initial iterative weighted DLT estimate, and the non-linear (optimised) estimate, versus GPS ground truth.

#### Discussion and Conclusions

For the fixed frame-rate, the results indicate some improvements in the accuracy of the visual odometry estimates over small distances only. The fact that this is not true for the longer segments is an unexpected result. One possible explanation is that there may still be outliers present in some of the frame to frame correspondences during the non-linear optimisation which have a greater effect on the accuracy of the homography estimate than during the linear estimate with the iterative weighted DLT. Considering that a single inaccurate measurement of the homography will have a greater effect on the long range accuracy than the short range accuracy, then this may be the most likely explanation. Unlike the results for the fixed frame-rate, there are improvements in the accuracy of the visual odometry estimates for all distances using the non-linear optimisation over the linear estimate.

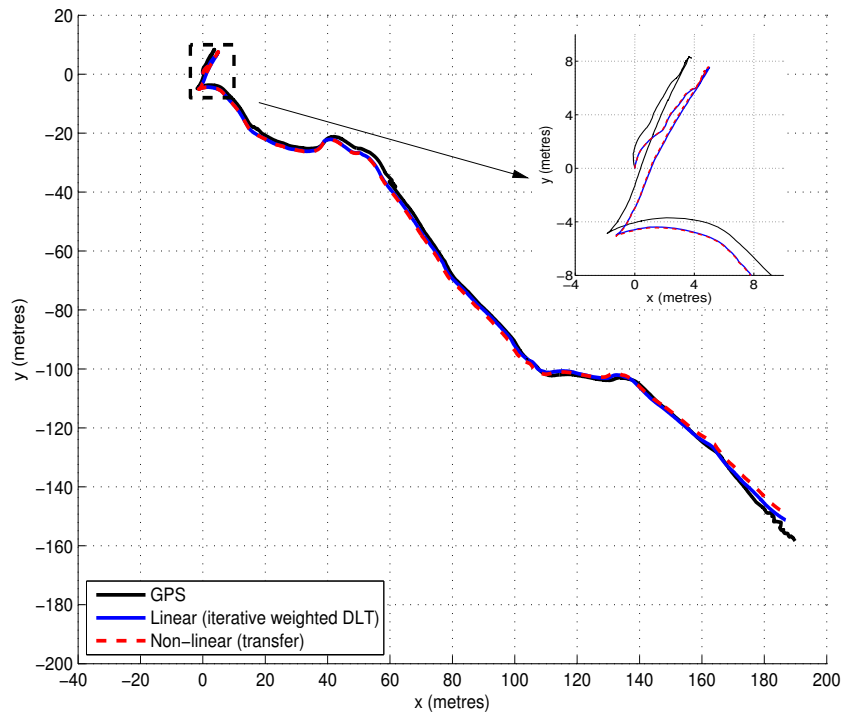


(a) Fixed frame-rate

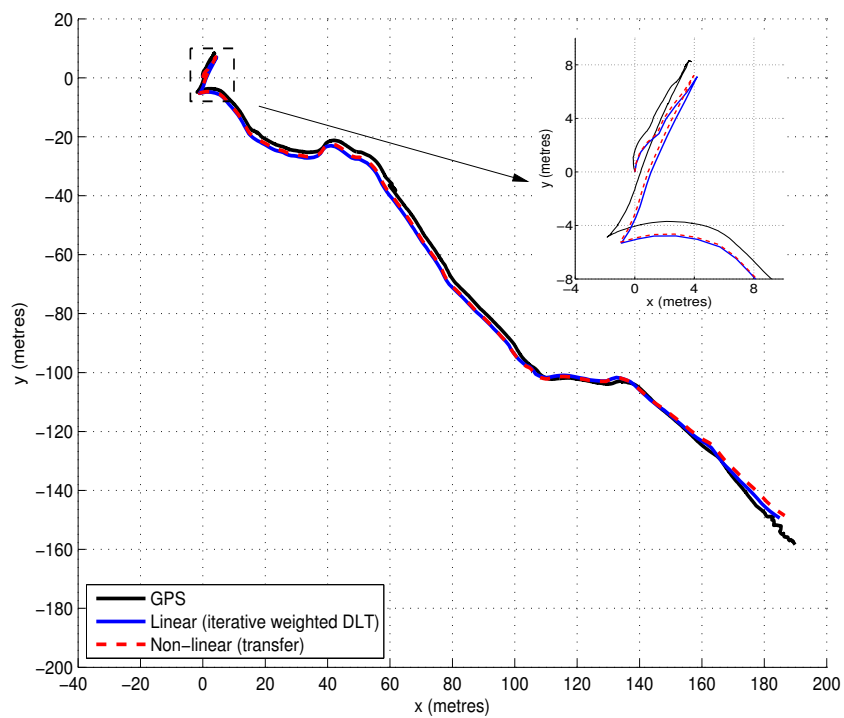


(b) Variable frame-rate

Figure 5.13: Box plots of the visual odometry errors (Hyperions, Triggs ground plane constraint). An initial estimate of the Homography between frames was found using the iterative weighted DLT and threshold  $\theta < 89^\circ$ . This estimate was then optimised using the transfer error cost function. The results shown are for a separate set of 20 trials used previously in figures 5.6a and 5.6b.



(a) Fixed frame-rate



(b) Variable frame-rate

Figure 5.14: Plot of the visual odometry estimates for the Triggs ground plane constraint using the linear iterative weighted DLT and the non-linear optimisations (transfer error) versus GPS ground truth for one of the fixed and variable frame-rate trials. The paths have been manually aligned.



Mode	Statistic	Distance Travelled (metres)						
		25	50	75	100	125	150	175
Linear (iterative weighted DLT)	median	1.88	3.08	4.08	4.47	4.74	5.01	5.31
	IQR	1.00	2.90	3.20	2.69	1.29	0.73	0.93
Non-Linear (transfer)	median	1.85	3.02	4.04	4.56	4.96	5.36	5.99
	IQR	0.97	2.82	2.92	2.10	0.81	1.31	1.45

(a) Fixed frame-rate.

Mode	Statistic	Distance Travelled (metres)						
		25	50	75	100	125	150	175
Linear (iterative weighted DLT)	median	1.61	2.96	3.86	4.32	4.81	5.63	6.48
	IQR	0.84	2.54	2.14	0.99	1.04	2.18	2.43
Non-Linear (transfer)	median	1.44	2.71	3.69	4.10	4.58	5.25	5.83
	IQR	0.67	2.97	2.56	1.10	0.95	2.22	2.09

(b) Variable frame-rate.

Table 5.7: visual odometry errors (Hyperions, Triggs ground plane constraint). An initial estimate of the Homography between frames was found using the iterative weighted DLT and threshold  $\theta < 89^\circ$ . This estimate was then optimised using the transfer error cost function. All errors have units of metres. IQR is the inter-quartile range.

## 5.2.8 Discussion and Conclusions: Ground Plane Visual Odometry

### 5.2.8.1 Direct Linear Transforms (DLT's)

The results in sections 5.2.6 and 5.2.7 showed that for both the Euclidean and Triggs ground plane constraints respectively, the initial visual odometry estimates found using the standard DLT's were highly sensitive to the angular threshold  $\theta$  placed on the cameras effective field of view. Two variants of the standard DLT were proposed termed the weighted and the iterative weighted DLT. The goal of each was to obtain a linear estimate for the homography  $H$  which accounted for the uncertainty of keypoint location during detection on the image plane.

For both the fixed and variable frame-rates, the visual odometry estimates for the Euclidean and Triggs ground plane constraints found using each of the weighted and iterative weighted DLT's were in general more accurate than the estimates found using the standard DLT's. The weighted and iterative weighted DLT's were also less sensitive to the angular threshold  $\theta$  on the cameras effective field of view. For the fixed frame-rate, the accuracy of the visual odometry estimates found using the iterative weighted DLT was not significantly better than that found using the weighted DLT for both the Euclidean and Triggs ground plane constraints. The reasons for these observations

were discussed previously in sections 5.2.6 and 5.2.7. For the variable frame-rate, although there was minimal difference between the accuracy of the visual odometry estimates for the Euclidean ground plane constraint found using the the weighted and iterative weighted DLT's, significant improvements using the latter were observed for the Triggs ground plane constraint.

Overall, the results suggest that for the particular downward facing camera configuration used in the Hyperion sequence, when estimating the camera egomotion using either the Euclidean or Triggs ground plane constraint, both the weighted and iterative weighted DLT's are a suitable alternative to the standard DLT's.

### 5.2.8.2 Euclidean versus Triggs

From the results presented, the visual odometry estimates obtained using the Triggs ground plane constraint were more accurate than those obtained using the Euclidean ground plane constraint. For the Euclidean ground plane constraint, it was assumed that the normal to the plane was  $\mathbf{n}_p = (0, 0, 1)^T$  (the cameras principal axis was orthogonal to the ground plane). Any variations from this assumption have the potential to limit the accuracy of the frame to frame egomotion estimates and hence the visual odometry estimates. A number of factors, all relevant to the Hyperion sequence, for which  $\mathbf{n}_p \neq (0, 0, 1)^T$  include: inaccurate alignment of the camera with respect to the ground plane, camera vibration, and movement of the robot over rocks and uneven terrain.

The Triggs ground plane constraint can potentially account for these factors. It is of interest to summarise here the results of Scaramuzza [198] who compared the relative performance of the Euclidean and Triggs ground plane constraints for egomotion estimates using synthetic omnidirectional (wide-angle) image data. A brief summary of his observations is:

- In the presence of image noise, the Euclidean results were in general better than those for the Triggs algorithm for perfect camera alignment ( $\mathbf{n}_p = (0, 0, 1)^T$ ). The opposite was true for non-perfect camera alignment (camera pitch = camera roll =  $1^\circ$ ).
- As the variance in the distribution of the keypoint correspondences along the y axis decreases (for example, all correspondences being located on one side of the image), the accuracy of the egomotion estimates using the Euclidean constraint was better than for the Triggs constraint for perfect camera alignment.

Furthermore, in the case where all keypoint correspondences approach a degenerate configuration, such as being collinear (lying on a single conic on the image plane), the results using Triggs gave extremely poor performance. This poor performance for Triggs with near degenerate configuration was found to exist also for the case of non-perfect camera alignment, (camera pitch = camera roll =  $1^\circ$ ). However, as the variance in the distribution of the points increased, the results for Triggs were better than for the Euclidean case.

Although the results for the Hyperion sequence do not consider the effect of image noise and distribution of the keypoint correspondences explicitly, the distribution of the keypoints was in most instances far from reaching a degenerate configuration — pSIFT was able to detect and find corresponding keypoints in most regions of the image. Inaccuracies in egomotion estimates are therefore attributed primarily to image noise and variations in the camera alignment with respect to the ground plane. Based on the results of Scaramuzza [198], for both these conditions the Triggs method will in general give improved estimates compared to the Euclidean method. The results observed in the experiments for the Hyperion sequence appear to support these observations.

To conclude, although the Triggs ground constraint requires solving for a homography with more degrees of freedom than that using the Euclidean ground constraint, the results show that it provides more accurate visual odometry estimates for the hyperion sequence for each of the fixed and variable frame-rates. This is most likely due to the fact that it can account for variations in the normal to the plane  $\mathbf{n}_p$  and, as suggested by Scaramuzza, is less sensitive to image noise. If for example the distribution of the keypoint correspondences between two frames began approaching a near degenerate configuration, the *switching* scheme of Scaramuzza [198] could be used where the Euclidean ground plane constraint is used in preference to the Triggs ground plane constraint. The accuracy of both the Euclidean and Triggs ground plane constraints are still limited by the fact that the ground plane constraint is only an assumption/constraint made. In reality the world points in the Hyperion sequence would not be perfectly coplanar as the majority of the keypoints detected are rocks that are elevated above the ground plane. There would also be natural undulations of the ground. However, accurate visual odometry results were still able to be found using the ground assumption/constraint.

Cost Function	Frame-rate	Statistic	Distance Travelled (metres)						
			25	50	75	100	125	150	175
Transfer	Fixed	median	1.57	3.32	4.82	5.61	7.39	8.91	9.19
		IQR	1.31	2.86	1.67	1.52	3.38	3.66	2.84
	Variable	median	1.59	3.49	4.83	6.06	8.32	10.02	10.40
		IQR	1.59	2.22	1.63	2.37	3.86	4.54	3.94
Geometric	Fixed	median	1.57	3.31	4.82	5.59	7.36	8.87	9.15
		IQR	1.31	2.86	1.67	1.48	3.32	3.60	2.81
	Variable	median	1.54	3.45	4.82	5.87	8.32	9.76	10.21
		IQR	1.58	2.40	1.74	2.25	4.04	4.68	3.94

(a) Euclidean ground plane constraint.

Cost Function	Frame-rate	Statistic	Distance Travelled (metres)						
			25	50	75	100	125	150	175
Transfer	Fixed	median	1.43	2.81	3.67	4.21	4.52	5.10	5.55
		IQR	0.83	2.96	2.88	1.67	0.64	1.75	1.69
	Variable	median	1.44	2.71	3.69	4.10	4.58	5.25	5.83
		IQR	0.67	2.97	2.56	1.10	0.95	2.22	2.09

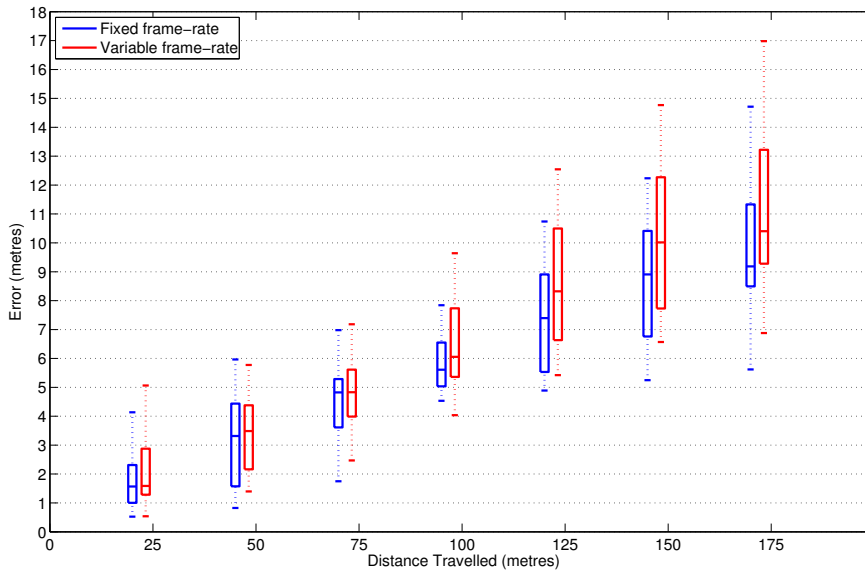
(b) Triggs ground plane constraint.

Table 5.8: Visual odometry errors (Hyperion, Euclidean and Triggs ground plane constraints) for fixed and variable frame-rates. The results are shown only for the non-linear estimates — the initial linear estimates were obtained using the iterative weighted DLT with angular threshold  $\theta < 89^\circ$ . The blue entries indicate the smallest median error for each distance and cost function. IQR is the inter-quartile range.

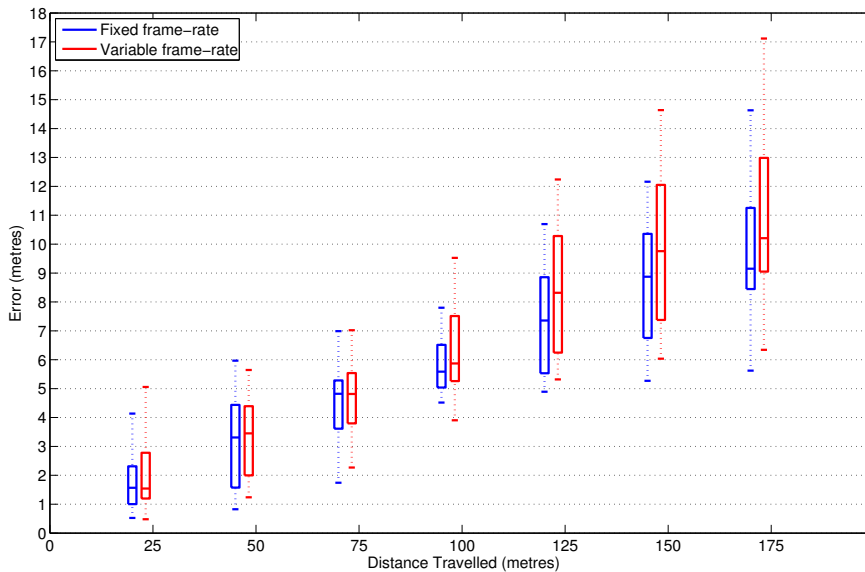
### 5.2.8.3 Fixed versus Variable Frame-rates

It was suggested previously that increasing the change in pose between the frames used to compute camera egomotion has the potential to improve the accuracy of visual odometry estimates. The variable frame rate scheme described in section 5.2.2 was used to do this. The comparison of the relative accuracy of the visual odometry estimates found using the fixed and variable frame-rates are shown in figures 5.15 and 5.16 for the Euclidean ground plane constraint (non-linear) and the Triggs ground plane constraint (non-linear) respectively. These same results are summarised in table 5.8. The same segments (position of start and end points) used to find the errors for the variable frame-rate have were also used to find the errors for the fixed frame-rate in an attempt to prevent any potential bias in the results. The results have been accumulated for the same 20 trials used to find the results in figures 5.9 and 5.13.

Overall, the results indicate that when using the Euclidean ground plane constraint, for each cost function the accuracy of the visual odometry estimate found using the fixed-frame rate are more accurate than those found using the variable frame-rate. The relative accuracy of the estimates for the fixed and variable frames rate are much more similar for the Triggs ground plane constraint, however, overall the accuracy of the



(a) Transfer Error



(b) Geometric Error

Figure 5.15: Box plot of the visual odometry errors (Hyperion, Euclidean ground plane constraint) for the fixed and variable frame-rates. The results are shown only for the non-linear estimates — the initial linear estimates were obtained using the iterative weighted DLT with angular threshold  $\theta < 89^\circ$ .

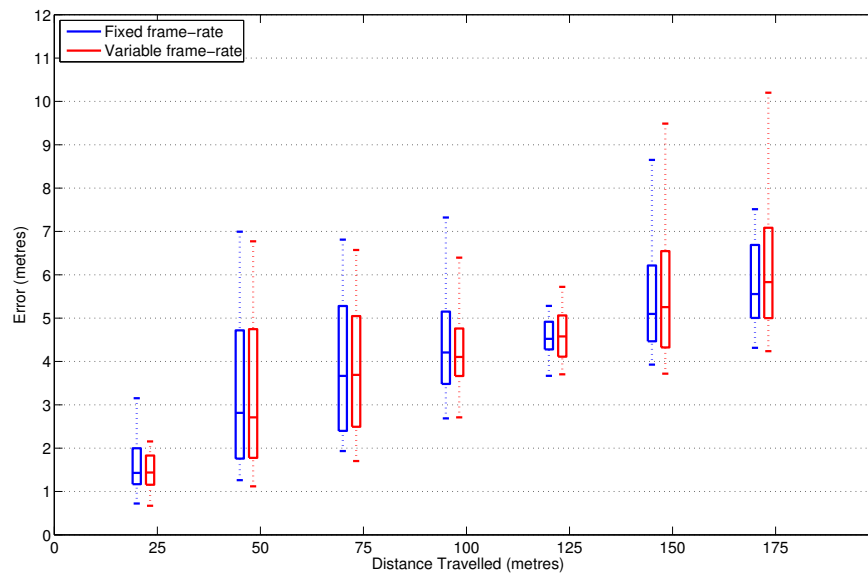


Figure 5.16: Box plot of the visual odometry errors (Hyperion, Triggs ground plane constraint) for the fixed and variable frame-rates. The results are shown only for the non-linear estimates — the initial linear estimates were obtained using the iterative weighted DLT with angular threshold  $\theta < 89^\circ$ .

estimate found using the fixed frame-rate is marginally better than that found using the variable frame-rate, especially as the length of the segment (distance travelled) increases.

The variable frame-rate algorithm increases the change in pose between the frames used to compute the camera egomotion, and this has the advantage of improving the signal to noise ratio in the change in keypoint positions in the image. However, fewer correspondences are used to compute the egomotion using the fixed frame-rate. The accuracy of the egomotion estimates may therefore be more sensitive to outliers, and may explain why the accuracy of the visual odometry estimates using the fixed frame-rate are better than those using the variable frame-rate. However, there may be advantages using the variable frame-rate for the case of *generalised* visual odometry which makes no assumptions regarding the camera motion or the position of the scene points (i.e.. being coplanar). This will be investigated in section 5.2.9.

### 5.2.9 Generalised (Unconstrained) Visual Odometry

If no information regarding the position of the world points associated with keypoint correspondences is known, for example being coplanar, then a *generalised* means for estimating visual odometry needs to be used. The term generalised is defined to mean that no assumptions/constraints are made regarding the position of the world points

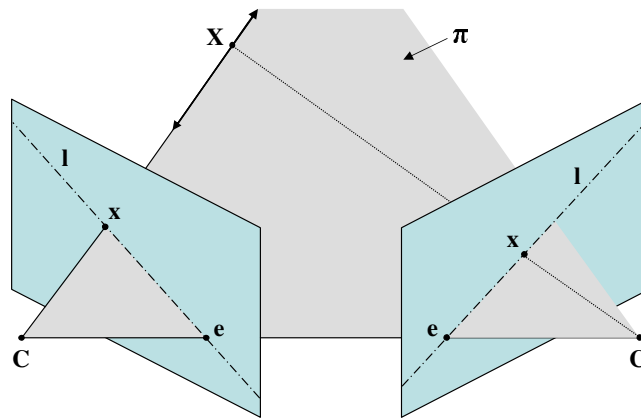


Figure 5.17: The epipolar geometry between two cameras centred at points  $C$  and  $C'$ .

(except that they remain stationary between views). Although constraints can be placed on the camera motion, the experiments in this section assume that the camera can undergo full six degree of freedom motion.

### 5.2.9.1 Epipolar Geometry and the Essential Matrix

Referring to figure 5.17, let  $\mathbf{x}$  and  $\mathbf{x}'$  be a pair of corresponding keypoints, associated with the world point  $\mathbf{X}$ , detected in different perspective images captured by two cameras centred at  $C$  and  $C'$ . The relationship between the homogeneous coordinates of the keypoints  $\mathbf{x} = (x, y, 1)^T$  and  $\mathbf{x}' = (x', y', 1)^T$  is defined by the epipolar geometry between the views.

For a monocular camera, the position of the world point  $\mathbf{X}$  on the line passing through  $C$  and  $\mathbf{x}$  can not be derived from the coordinate of the point  $\mathbf{x}$ . However,  $\mathbf{X}$  must be in front of the camera. If the camera moves to a new position  $C'$  under the action of some rotation  $R$  and translation  $\mathbf{t}$ , the line connecting the two camera centres intersects the image planes of camera 1 and 2 at the epipolar points  $\mathbf{e}$  and  $\mathbf{e}'$  respectively. The points  $C, C'$  and  $\mathbf{x}$  define a unique plane  $\pi$  — the epipolar plane for the point  $\mathbf{x}$ . Since the depth of the point  $\mathbf{X}$  is unknown, it could lie anywhere on the line  $l'$  in image 2. The line  $l'$  is the epipolar line for the point  $\mathbf{x}$ , and is defined as the intersection of the epipolar plane and the image plane of camera 2. It may be said then that under any generalised motion of the camera, any point  $\mathbf{x}$  in image 1 maps to its epipolar line  $l'$  in image 2:  $\mathbf{x} \mapsto l'$ . Conversely, a point  $\mathbf{x}'$  in image 2 maps the the epipolar line  $l$  in image 1:  $\mathbf{x}' \mapsto l$ . Given a set of corresponding keypoints in two images, the epipolar lines  $l'$  for all keypoints  $\mathbf{x}$  in image 1 will intersect at the epipolar point  $\mathbf{e}'$ , and the epipolar lines  $l$  for all keypoints  $\mathbf{x}'$  in image 2 will intersect at the

epipolar point  $\mathbf{e}$ . Notice that the positions of the epipolar points  $\mathbf{e}, \mathbf{e}'$  in images 1 and 2 define the direction of the camera translation from viewpoint 1 to 2 and viewpoint 2 to 1 respectively.

As detailed in [95], using the principal of point transfer via a plane, the points  $\mathbf{x}$  and  $\mathbf{x}'$  are projectively equivalent and related by a 2D homography  $H_{\Pi}$ :

$$\mathbf{x}'_i = H_{\Pi} \mathbf{x}_i, \quad \forall \mathbf{x}_i, \mathbf{x}'_i, \quad (5.42)$$

where  $\mathbf{x}_i$  and  $\mathbf{x}'_i$  are a corresponding pair of keypoints in the set of all corresponding keypoints between the images.

Referring to figure 5.17, the epipolar line  $\mathbf{l}'$  is defined as the vector cross product  $\mathbf{l}' = \mathbf{e}' \times \mathbf{x}' = [\mathbf{e}']_{\times} \mathbf{x}'$ . Here, the notation used in [95] is used where for any vector  $\mathbf{A} = (a_1, a_2, a_3)^T$ ,

$$[\mathbf{A}]_{\times} = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix}. \quad (5.43)$$

Substituting equation 5.11 into the relationship  $\mathbf{l}' = [\mathbf{e}']_{\times} \mathbf{x}'$  gives

$$\mathbf{l}' = [\mathbf{e}']_{\times} H_{\Pi} \mathbf{x}, \quad (5.44)$$

$$\mathbf{l}' = \mathbf{F} \mathbf{x}, \quad (5.45)$$

where  $\mathbf{F}$  is the  $3 \times 3$  *fundamental* matrix. The fundamental matrix also describes the relationship  $\mathbf{l} = \mathbf{F}^{-1} \mathbf{x}'$  and relates points  $\mathbf{x}$  and  $\mathbf{x}'$  by [95]:

$$\mathbf{x}'_i{}^T \mathbf{F} \mathbf{x}_i = 0, \quad \forall \mathbf{x}_i, \mathbf{x}'_i \quad (5.46)$$

assuming only that the position of world points  $\mathbf{X}_i$  remain fixed between views. Again,  $\mathbf{x}_i$  and  $\mathbf{x}'_i$  are a corresponding pair of keypoints in the set of all corresponding keypoints between the images.

If the calibrated coordinates of the keypoints are known, for example the spherical coordinates  $\eta$  and  $\eta'$ , then the condition in equation 5.46 can be written as

$$\eta'_i{}^T E \eta_i = 0, \quad \forall \eta_i, \eta'_i, \quad (5.47)$$

where  $E$  is the essential matrix<sup>1</sup>. Equation 5.48 can also be written using the homoge-

<sup>1</sup>If  $\check{\mathbf{x}} = n\eta$ , where  $n$  is some scalar and  $\eta$  is the coordinate of the point  $\mathbf{x}$  on the unit sphere, then for a perspective camera there is some matrix  $K_1$  for which  $K_1 \check{\mathbf{x}} = \mathbf{x}$ . Additionally, there is some matrix



neous coordinates  $\check{\mathbf{x}} = m\eta$  and  $\check{\mathbf{x}}' = n\eta'$ , where  $m$  and  $n$  are any non-zero scalar values, as

$$\check{\mathbf{x}}_i'^T E \check{\mathbf{x}}_i = 0, \quad \forall \check{\mathbf{x}}_i, \check{\mathbf{x}}_i'. \quad (5.48)$$

Assuming that the first camera matrix is  $P_1 = [I_{3 \times 3} | \mathbf{0}]$ , where  $\check{\mathbf{x}} = P\mathbf{X}$ , then for a change in rotation  $R$  and translation  $\mathbf{t}$  between views, the second camera matrix is  $P_2 = [R | \mathbf{t}]$ , where  $\check{\mathbf{x}}' = P_2\mathbf{X}$ . This rotation  $R$  and translation  $\mathbf{t}$  are the same as those defined previously in equation 5.1 (pg. 268). The essential matrix  $E$  is a function of this rotation  $R$  and translation  $\mathbf{t}$  (up to an unknown scale ambiguity) [95]:

$$E = [\mathbf{t}]_{\times} R. \quad (5.49)$$

If camera 1 is positioned at the origin of the world coordinate frame, then  $\mathbf{C} = (0, 0, 0)^T$ . The position  $\mathbf{C}'$  of the second camera is  $\mathbf{C}' = -R^T \mathbf{t}$ .

As previously discussed in section 5.2.1, the five-point algorithm of Nistér [180] and RANSAC [70] are used to find the essential matrix between views given a set of calibrated keypoint correspondences  $\eta \leftrightarrow \eta'$ . The details of this process are given in the same section. The rotation  $R$  and translation  $\mathbf{t}$  are recovered from the essential matrix up to a four-fold ambiguity using the method described in [95] — there are two possible solution each for the rotation  $R$  and unit baseline translation  $\mathbf{t}$ . Only a unit baseline translation can be found as the true magnitude of the translation can not be found without knowing the coordinates of the world points. To resolve this four-fold ambiguity, the scene reconstruction algorithm described in the same text [95] is used. The correct solution is the one which finds, for a corresponding pair of keypoints in the upper hemisphere of the view sphere (i.e. in front of the camera), a reconstructed world point in front of both cameras. Once the ambiguity is resolved, the position of the world points  $X$  are found for a unit baseline translation [95]. The initial estimate of the camera egomotion obtained from the essential matrix can be considered as a linear estimate.

### 5.2.9.2 Linear estimates: fixed versus variable frame-rate

Before outlining in detail the full visual odometry algorithm, the visual odometry estimates for the Hyperion sequence with the fixed and variable frame-rates will be compared using the linear estimate of the camera egomotion derived from the essential

---

$K_2$  where  $K_2 \check{\mathbf{x}}' = \mathbf{x}'$ . Equation 5.46 can then be rewritten as  $\check{\mathbf{x}}' K_2^T F K_1 \check{\mathbf{x}} = 0$ . The essential matrix  $E$  is related to the fundamental matrix  $F$  by  $E = K_2^T F K_1$ .

matrix  $E$ . Importantly, no threshold on the effective field of view of the camera is required as keypoints are not constrained to be coplanar. For simplicity, to resolve the correct magnitude of the translation  $\mathbf{t}$ , the GPS ground truth data is used as a virtual encoder. Since GPS has limited accuracy, the estimate of the magnitude of the translation obtained from the GPS measurements would likely be less accurate than that obtained by a traditional wheel encoder.

## Results

For each of the fixed and variable frame-rates, the visual odometry estimates for two separate trials are shown in figures 5.18 and 5.19 respectively.

## Discussion

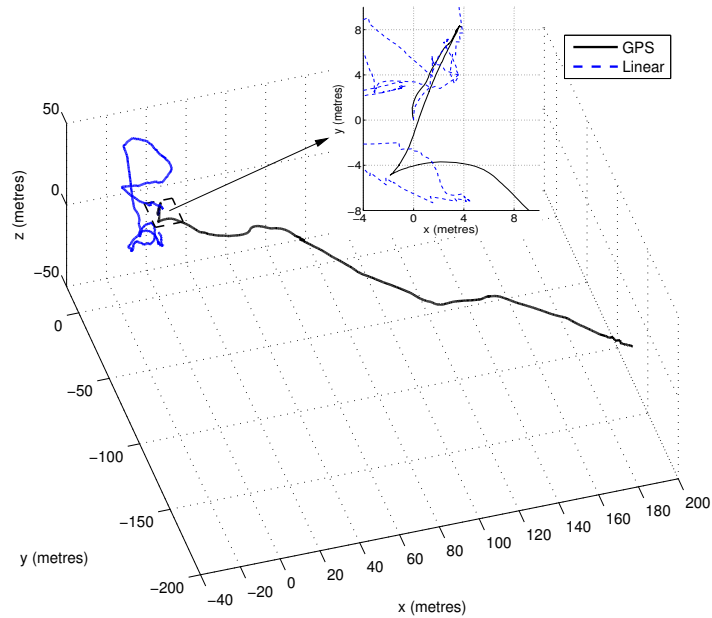
From simple inspection of the results, it is evident that visual odometry estimates using the variable frame-rate are superior to those using the fixed frame-rate. Although there is the potential to improve the egomotion estimates using a non-linear optimisation, which will be discussed in section 5.2.9.3, the inability to find a good initial estimate of the camera egomotion using the fixed frame-rate means that the optimisation could converge on local minima. Since the results for the variable frame-rate are superior to those for the fixed frame-rate, for the remaining experiments only the variable frame-rate data will be used.

Before proceeding, it is important to recall that accurate visual odometry estimates were able to be found using the Euclidean and Triggs ground plane constraints for the fixed frame-rate. This highlights the advantage of enforcing a ground plane constraint when valid (or approximately valid in the case of the Hyperion sequence).

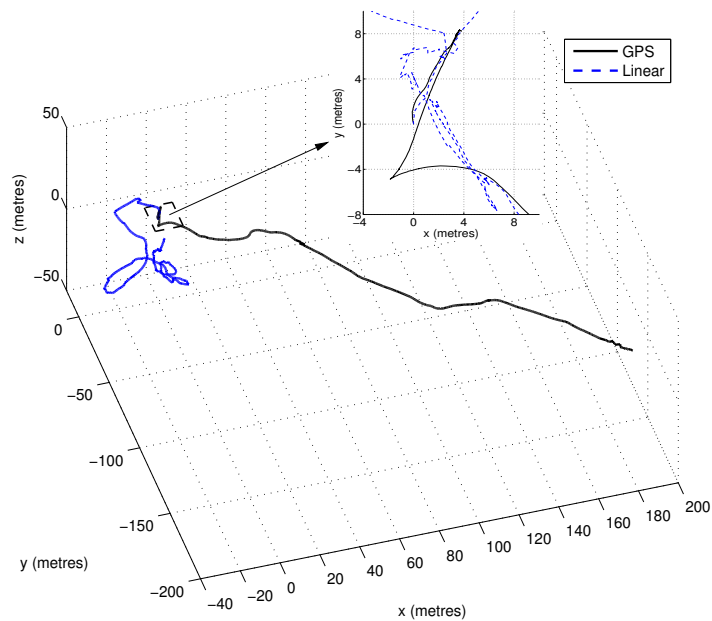
### 5.2.9.3 Visual Odometry Algorithm

The method for obtaining a generalised monocular visual odometry estimate is outlined here. The individual steps relating to iterative refinement and resolving the scale ambiguity in the magnitude of the translation will be discussed later. The steps in the method are:

1. Given a set of keypoint correspondences  $\eta \leftrightarrow \eta'$  between frame, obtain an estimate of the essential matrix using Nistér's five-point algorithm [180] and RANSAC [70].
2. Find the estimate of the second camera matrix  $P_2 = [R|\mathbf{t}]$  from the essential matrix, where  $\|\mathbf{t}\| = 1$  – the first camera matrix is  $P_1 = [I_{3 \times 3}|\mathbf{0}]$ . This requires

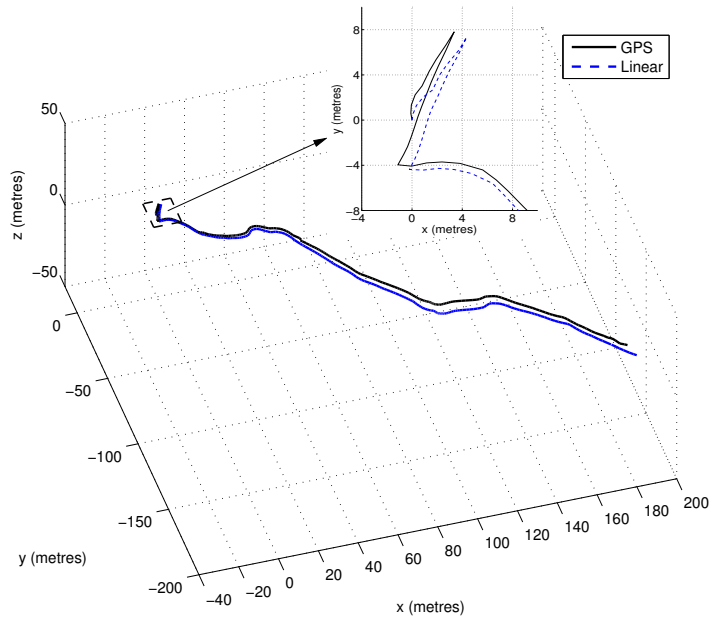


(a) Trial 1.

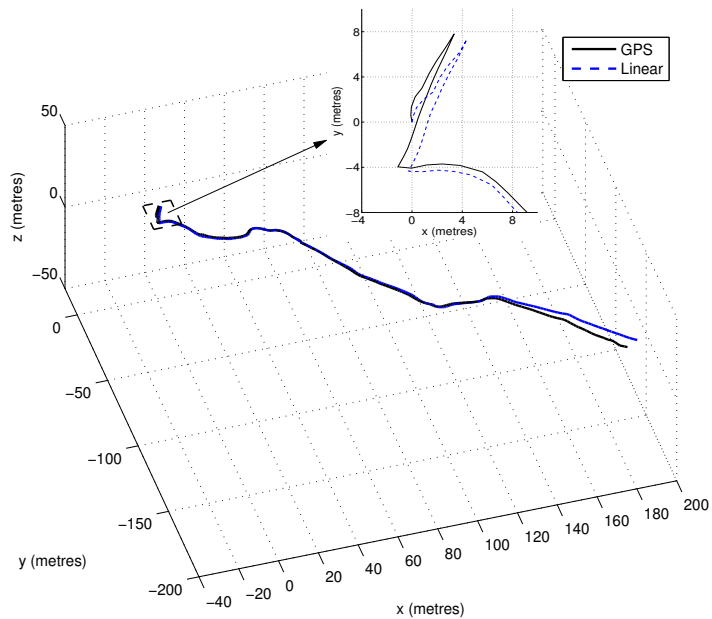


(b) Trial 2.

Figure 5.18: Visual odometry estimates versus GPS ground truth for the fixed frame-rate. The egomotion was resolved from the essential matrix using the five-point algorithm and RANSAC. The magnitude of the baseline translation was resolved from the GPS ground truth values.



(a) Trial 1



(b) Trial 2

Figure 5.19: Visual odometry estimates versus GPS ground truth for the variable frame-rate. The egomotion was resolved from the essential matrix using the five-point algorithm and RANSAC. The magnitude of the baseline translation was resolved from the GPS ground truth values.

resolving the four-fold ambiguity in the estimate [95].

3. Resolve the structure of scene points for a unit baseline translation  $\mathbf{t}$  [95].
4. Optimise the values of the second camera matrix  $P_2$  and the positions of the world points  $\mathbf{X}$ .
5. If first set of frame to frame correspondences, go to step 1.
6. Find the keypoints which appeared in the previous set of frame to frame correspondences and the Euclidean position of their associated world points in the current camera frame of reference.
7. Resolve the magnitude of the baseline translation.
8. Rescale the magnitude of the camera matrix translation and world points.
9. Repeat from step 1.

Using this method, the estimate of both the visual odometry and the position of all the world points are found. If desired, these world points could be used to construct a map of the environment.

It is interesting to note here that this method differs from the monocular scheme proposed by Nistér [181]. Using Nistér's method, the camera egomotion is resolved using only the correspondences for which the position of the reconstructed world points have been found in previous frames. Using the method outlined above, these points are used only to resolve the magnitude of the translation. This is flexible in the sense that another sensor such as wheel odometry could be used to resolve this magnitude. This method is selected as Nistér's method essentially discards many of the keypoint correspondences in the image when solving for the camera egomotion, particularly for the variable frame-rate. Since one of the advantages of wide-angle vision is the increased field of view, it is of benefit to exploit all possible correspondences throughout the image to resolve the motion. This is similar in some respects to the method of Tardif et al [218] who also used all keypoint correspondences to obtain an initial estimate for the camera egomotion, however, they retain only the initial estimate for the rotation and use the position of the existing world points from previous frames to find both the direction and magnitude of the camera translation.

### **Iterative Refinement**

The accuracy of the egomotion estimates and the position of the world points are improved using a non-linear iterative refinement. This requires optimising the parameters

of the second camera matrix  $P_2$  and the homogeneous coordinates of the world points  $\mathbf{X}$ . Given an initial estimate of the second camera matrix  $P_2 = [R|\mathbf{t}]$  obtained from the essential matrix, where  $\|\mathbf{t}\| = 1$ , and the position of the world points recovered for this initial estimate, the optimisation minimises the sum of the squared reprojection errors

$$\varepsilon = \sum_i d(\mathbf{u}_i, \mathbf{u}(P_1 \mathbf{X}_i))^2 + d(\mathbf{u}'_i, \mathbf{u}'(P_2 \mathbf{X}_i))^2, \quad (5.50)$$

where  $P_1 = [I|\mathbf{0}]$ .  $\mathbf{u}_i$  and  $\mathbf{u}'_i$  are the coordinates of a corresponding pair of keypoints in the first and second stereographic images.  $\mathbf{u}(P_1 \mathbf{X}_i)$  is the coordinate, in the first stereographic image, of the point  $P_1 \mathbf{X}_i$  mapped to this image. Likewise,  $\mathbf{u}'(P_2 \mathbf{X}_i)$  is the coordinate, in the second stereographic image, of the point  $P_2 \mathbf{X}_i$  mapped to this image. The positions on the stereographic image plane are used as pSIFT keypoints are detected in these stereographic images. The variable  $d(\mathbf{u}_i, \mathbf{u}(P_1 \mathbf{X}_i))$  is the Euclidean distance between the points  $\mathbf{u}_i$  and  $\mathbf{u}(P_1 \mathbf{X}_i)$  measured on the stereographic image, and  $d(\mathbf{u}'_i, \mathbf{u}'(P_2 \mathbf{X}_i))$  is the Euclidean distance between the points  $\mathbf{u}'_i$  and  $\mathbf{u}'(P_2 \mathbf{X}_i)$  measured on the stereographic image. The combined reprojection error  $d(\mathbf{u}_i, \mathbf{u}(P_1 \mathbf{X}_i)) + d(\mathbf{u}'_i, \mathbf{u}'(P_2 \mathbf{X}_i))$  for each keypoint correspondence is calculated before implementing the optimisation. Any correspondence with a combined reprojection error greater than 4 pixels is removed.

For the purposes of implementing the optimisation, the parameters for the second camera matrix used are

$$P_2 = [q_0, q_1, q_2, q_3, t_x, t_y, t_z]^T, \quad (5.51)$$

where  $\mathbf{q} = (q_0, q_1, q_2, q_3)^T$  are the quaternion values derived from the second camera matrix rotation  $R$ . Each world point is defined by its homogeneous coordinate  $\mathbf{X}_i = [x_{w_i}, y_{w_i}, z_{w_i}, w_{w_i}]^T$ , and optimisation takes place over all the variables

$$\mathbf{v} = [q_0, q_1, q_2, q_3, t_x, t_y, t_z, x_{w_i}, y_{w_i}, z_{w_i}, w_{w_i}]^T, \quad i \in \{1, 2, \dots, n\} \quad (5.52)$$

where  $n$  is the number of keypoint correspondences. A Matlab implementation of Levenberg-Marquardt is used for the optimisation in the following experiments.

### Resolving the scale ambiguity (magnitude of baseline translation)

For monocular cameras, there is no means for resolving the magnitude of the baseline translation  $\mathbf{t}$  without knowledge of the scene (or input from other sensors). However, given the position of the world points found from previous frames, then the scale can be resolved relative to these. It is possible to then find a visual odometry estimate with a single *global* scale ambiguity — typically the first frame in the sequence

has a unit baseline translation, so the magnitude of translation for all other frames are obtained relative to this.

To resolve the magnitude of the baseline translation, step 7, a subset of correspondences for the current frame are used. These are the correspondences for which the position of the world points  $\mathbf{X}$  were resolved in a previous cycle (i.e. from a previous set of frame to frame correspondences). The coordinates  $\mathbf{X}'$  of these world points are found in the current camera frame of reference, and the distance to each point  $l' = \|\mathbf{X}'\|$  found. After finding the positions  $\mathbf{X}$  of these same points using the current set of correspondences and the optimisation method described, the distance to each of these points  $l = \|\mathbf{X}\|$  is found. The magnitude of the baseline translation  $b$  is then found by minimising the error

$$\epsilon = \sum_i \frac{l'_i - b l_i}{l_i l'_i}. \quad (5.53)$$

The denominator applies an inverse distance weighting which is necessary as world points at a greater distance from the camera will typically have a greater uncertainty (covariance) in their position than those nearer to the camera.

#### 5.2.9.4 Experiments: Hyperion Sequence

The generalised non-linear visual odometry algorithm described was used to estimate the visual odometry for the Hyperion sequence using the variable frame-rate method. Figure 5.20 shows the plot of the estimated path against the GPS ground truth data. For comparison, a plot of the linear estimate for the same trial has also been included — the linear estimate was obtained by omitting step (4). The paths have been manually aligned and a suitable global scale factor selected. The results in the figure show the  $x, y$  translation,  $x, z$  translation and both the pitch and roll in the current camera frame of reference. Note that no  $z$ -component for the GPS data was available for comparison.

The results show that the generalised visual odometry estimate obtained using the variable frame-rate method provides a good estimate of both the camera motion and relative scale between egomotion estimates. However, the large variations in pitch angle at  $x \approx 120m$  suggests that the accuracy could further be improved by incorporating some camera motion model and/or means of Kalman filtering for example which has been used with success by Corke et al [48] over a segment of the same image sequence. Their method, which used a fixed frame-rate, failed after travelling approximately 25 metres.

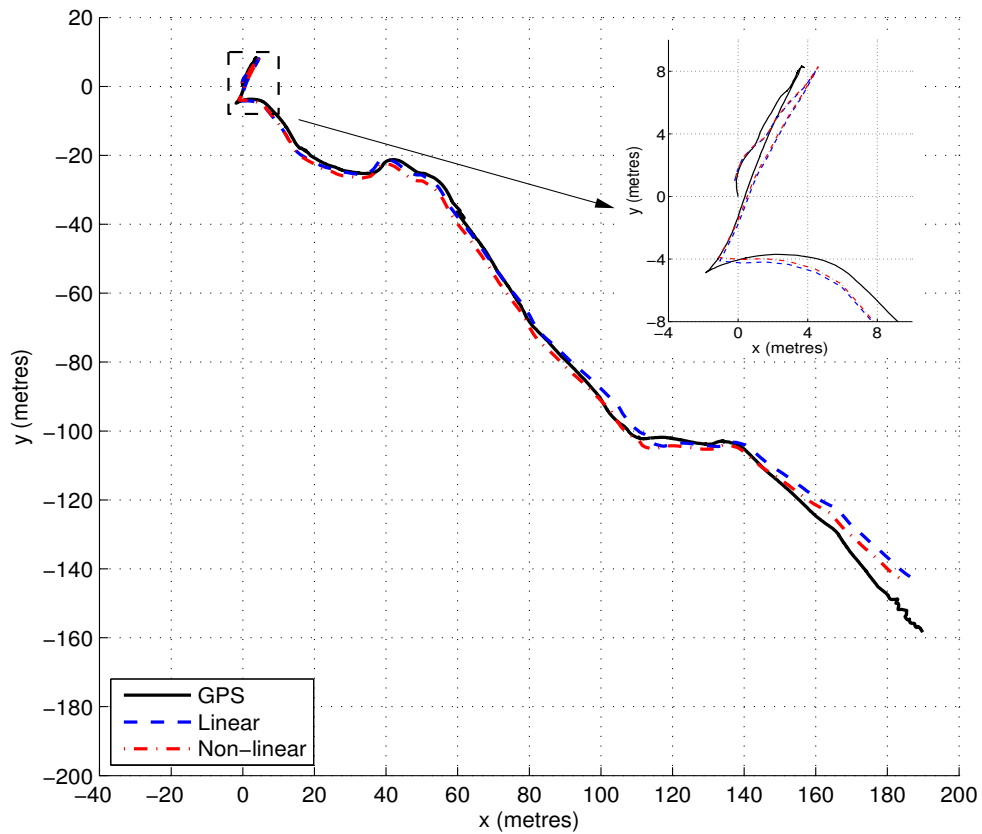
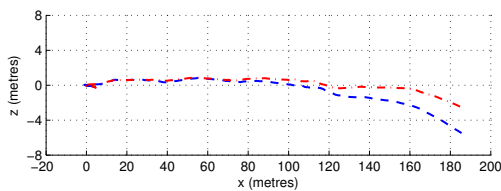
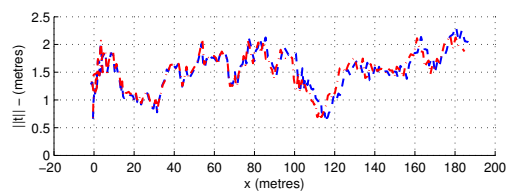
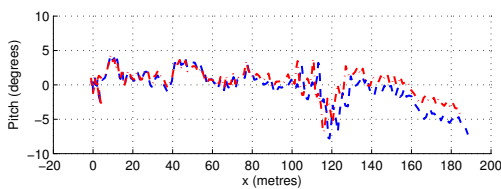
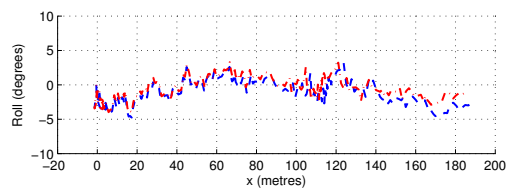
(a)  $x, y$  translation.(b)  $x, z$  translation(c) Magnitude of translation  $t$  vs  $x$  translation.(d) Pitch angle vs  $x$  translation.(e) Roll angle vs  $x$  translation.

Figure 5.20: Visual odometry results for the Hyperion sequence using no constraints on the vehicle motion or position of scene points. The pitch and roll angles are in the camera frame of reference and not the global frame.



### 5.2.9.5 Experiments: Fisheye Sequence

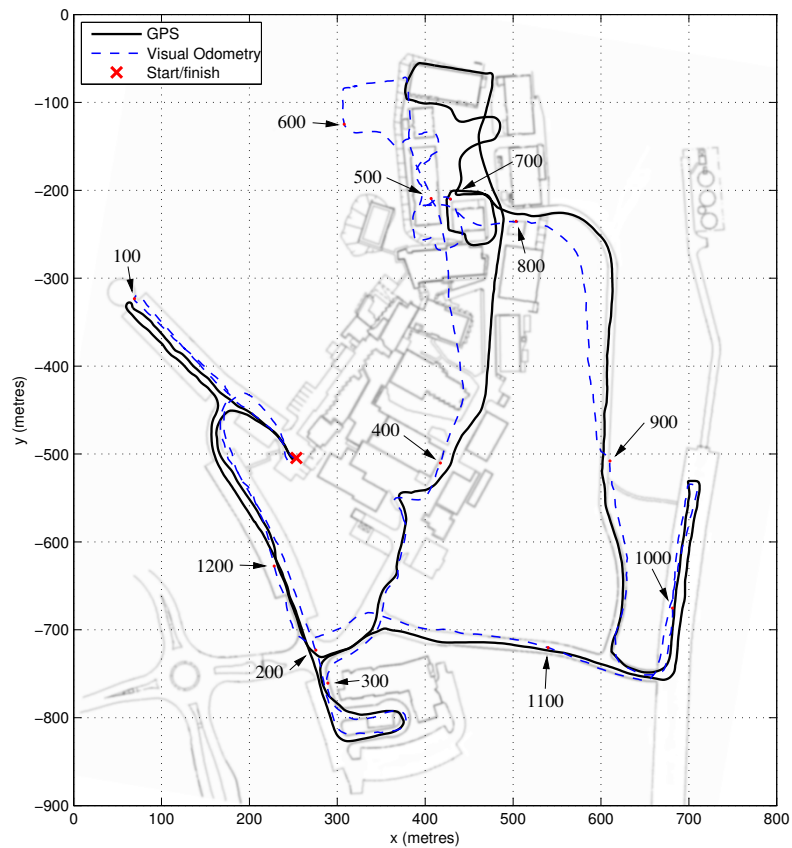
The visual odometry estimates for the 4.4 km fisheye sequence were also found using the generalised non-linear visual odometry algorithm described. The results are shown in figure 5.21. Unlike the Hyperion data set, the ratio of the distance to world points associated with each keypoint to the magnitude of the baseline translation was large. As a result, the ability to maintain an accurate overall scale between the magnitude of translation for successive egomotion estimates was limited. Therefore, the results presented in figure 5.21 use a simulated odometer to resolve the magnitude of the inter-frame translation using difference of GPS positions. This is the same method that was used to resolve the magnitude of translation for the linear estimates in figures 5.18 and 5.19 and again will be less accurate than real wheel odometry which was not available since the camera was hand held. Similarly to the Hyperion sequence, no z-component of the GPS data was available for comparison. The operating environment does contain variations in elevation, however, these can not be readily obtained.

Although the magnitude of the camera translation could not be reliably obtained using vision alone, the results still indicate that a suitable estimate for the camera egomotion between views (rotation and direction of translation) could be found. The variations in the paths, particularly in the upper half of the figure, are the results of some inaccurate estimates of the egomotion near reference frame 400 (i.e. the 400th estimate of the egomotion). Interestingly, this rotational misalignment appears to correct itself near reference frame 900 which is a coincidence only.

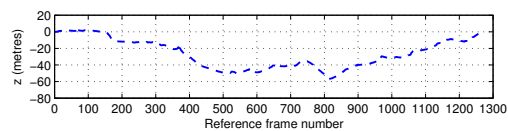
### 5.2.10 Conclusions

The pSIFT keypoint detector was used in this section to estimate the visual odometry of two outdoor wide-angle images sequences. Overall, accurate visual odometry estimates were able to be found using various constraints on the position of world points and camera motion.

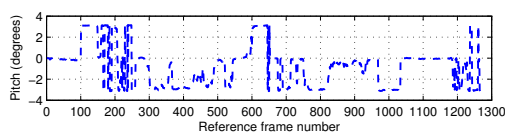
Visual odometry estimates for the Hyperion sequence (downward facing equian-gular catadioptric camera) were found using the Euclidean and Triggs ground plane constraints. For both cases, two suitable variations of the standard Direct Linear Transform (DLT) were proposed termed the weighted DLT and iterative weighted DLT. Both were shown in general to provide more accurate linear estimates of the camera egomotion whose results were far more consistent for different angular thresholds  $\theta$  on the camera's effective field of view. Suitable non-linear methods used to improve the accu-



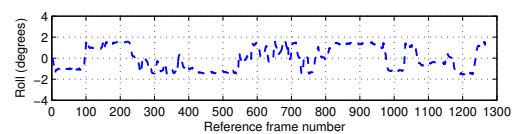
(a)  $x, y$  translation. The numerical annotations indicate reference frame numbers.



(b)  $z$  translation



(c) Pitch angle



(d) Roll angle

Figure 5.21: Visual odometry results for the Fisheye sequence using no constraints on the vehicle motion or position of scene points. The pitch and roll angles are in the camera frame of reference and not the global frame. The GPS ground truth results have been used as a virtual encoder to resolve the scale ambiguity in the magnitude of translation. The annotations on the figure are reference frame numbers — a reference number of 100 for example is the 100th iteration (egomotion estimate) of the variable frame-rate method.

racy of these estimates were then considered which minimise different cost functions. For the Euclidean ground plane constraint, the transfer and geometric cost functions were used, and both were found to give comparable performance for the fixed and variable frame-rates. In regards to the relative accuracy of the visual odometry estimates obtained using the Euclidean and Triggs ground plane constraints, improved accuracy was found using Triggs as it relaxes the, unreasonable, assumption that the camera's principal axis is always orthogonal to the ground plane. Although no advantage was observed using the variable frame-rate over the fixed frame-rate with respect to the accuracy of the visual odometry estimates, both were able to find reasonably accurate estimates, particularly using the Triggs ground plane constraint. The accuracy could potentially be further improved using the visual compass algorithm of Scaramuzza and Siegwart [200]. Their algorithm converts the wide-angle images used to estimate the camera egomotion to cylindrical panoramic images and then attempts to refine the orientation estimate by aligning these images in the space of appearance.

A generalised visual odometry algorithm was presented which made no assumptions regarding the camera motion or the position of the scene points. For the Hyperion sequence, the advantages of using the variable versus fixed frame-rate were illustrated using a linear estimate for the camera egomotion. For this same sequence, the monocular visual odometry algorithm was able to provide an accurate estimate for the camera motion where an accurate relative scale between the magnitudes of the successive estimates of the camera translations were maintained. For the fisheye sequence, although a reliable scale factor between the magnitude of translation estimates could not be retained, the visual odometry algorithm was in most instances able to find a suitable estimate of the camera motion. There were some deviations in the path as a results of integration some inaccurate estimates for the rotation, however, considering the length of the path was in excess of 4.4km the results are promising. To reliably resolve the magnitude of the camera translation between frames, aspects of the visual odometry algorithm of Tardif et al [218] or fusion of vision with inertial measurements or wheel odometry could potentially be used.

### 5.3 Visual place recognition

In this section the potential application of pSIFT to visual place recognition is demonstrated using the fisheye image sequence. As previously discussed, wide-angle cameras are suited for visual place recognition in large scale outdoor environments as they can obtain an image of the same scene from very different viewpoints. The "Video Google"

system of Sivic and Zisserman [210] is used to compare the similarity of images taken over the fisheye image sequence. This similarity between images can be used as an initial guide for loop closure detection, that is, recognising that the camera has returned to a previously visited place. The ability to detect previously visited locations is a key part of visual SLAM systems and is used to correct for drift in the estimate of camera location.

The Video Google system uses a visual ‘bag of words’ method to compare the similarity of two images, and has been used as the basis of image retrieval [49] algorithms and loop closure detection [99, 69] algorithms. The system is inspired by text retrieval methods. The visual words represent the visual vocabulary of keypoint descriptors, where each keypoint detected in an image is assigned to a visual word based on its descriptor values. Each image is then described by a visual word vector  $V_d$ , which is a weighted occurrence histogram of the visual words in the image. This cosine angle between the visual word vectors of any two images is used to measure their similarity. Importantly, the Video Google system is an appearance based method used to assess similarity between images, which means that the similarity is dependent solely on the keypoints detected in the images. A review of alternate methods used for loop closure detection and experimental comparisons can be found in [238]. Although higher level appearance based methods have since been developed, including a series of seminal works by Cummins et al [50, 51, 52], the Video Google system is well suited for demonstrating the ability of pSIFT to detect and describe accurately the same keypoints in different wide-angle images taken at very different viewpoints.

### 5.3.1 Establishing the visual-words

The visual vocabulary was learned offline using a separate fisheye image sequence operating in the same environment. pSIFT keypoints were detected in all the images, and their descriptors were used to find a visual vocabulary of 10,000 words using  $k$ -means clustering. The seed points (means) were initialised by randomly selecting 10,000 descriptors for the set of all keypoints. The distance between a descriptor and a mean was measured using Euclidean distance during clustering.

### 5.3.2 Word reliability

The Video Google systems tracks keypoints across multiple frames to determine stable keypoints, and the visual word vector for each image is constructed using only these

stable keypoints. Using this method improves the ability of the algorithm to correctly identify similar images. The problem with applying this to outdoor image sequence is that the ability to track keypoints is dependent on many factors, such as the change in camera pose between images. It is proposed that a word *reliability* metric can be used to achieve a similar result without the need to track keypoints across multiple frames, where this word reliability is learned offline.

A visual word is considered *reliable* if it satisfies two conditions. Firstly, it describes some salient region in the image. In this context salient is defined to mean any non-repeatable region or pattern in the image. Secondly, it is able to describe the same region in the environment robustly with respect to small projective transformations or viewpoint change. To determine word reliability, the same training sequence used to build the visual vocabulary is used. pSIFT keypoints are detected in each image, and each keypoint is assigned to a visual word. For successive images in the sequence, the pSIFT keypoint correspondences are found using the ambiguity metric for matching. Correct correspondences are then found using RANSAC and the five-point algorithm, as described in section 5.2.1. If two corresponding keypoints have been assigned the same visual word, the a correct match of the word has been. This word then satisfies the two constraints.

Let  $d_{i_n}$  be the number of times word  $i$  has been found in image  $n$  and  $m_{i_{n,n+1}}$  be the number of correct matches of word  $i$  between images  $n$  and  $n + 1$ . Then for a total of  $N > 2$  training images the following parameters are found:

$$Nd_i = \frac{d_{i_1}}{2} + \sum_{n=2}^{N-1} d_{i_n} + \frac{d_{i_N}}{2}, \quad (5.54)$$

$$Nm_i = \sum_{n=1}^{N-1} m_{i_{n,n+1}}, \quad (5.55)$$

from which the reliability  $R_i$  of the word  $i$  is evaluated as

$$R_i = \frac{Nm_i}{Nd_i}, \quad 0 \leq R_i \leq 1. \quad (5.56)$$

### 5.3.3 Visual Word Vector

Each image is represented by a visual word vector  $V_d$  whose length is equal to the number of words  $k$  in the vocabulary. Using the same definition as used in [210], for a

database of  $N$  images each element in the vector  $V_d$  for a given image is

$$V_{d_i} = (n_{id}/n_d) \log_{10}(N/n_i), \quad i \in \{1, 2, \dots, k\}. \quad (5.57)$$

Here,  $(n_{id}/n_d)$  is the term frequency  $tf$  which is the ratio of the frequency of word  $i$  in the current document  $n_{id}$  to the total number of words in the document  $n_d$ . The weighting  $\log_{10}(N/n_i)$  is the inverse document frequency  $idf$ , where  $n_i$  is number of images in the database (image sequence) which contain the word  $i$ . The purpose of the inverse document frequency is to reduce the weighting of visual words which appear in many images. The word reliability metric is then applied to find a modified visual word vector  $V'_{d_i}$ :

$$V'_{d_i} = R_i^2 (n_{id}/n_d) \log_{10}(N/n_i), \quad i \in \{1, 2, \dots, k\}. \quad (5.58)$$

A squared reliability metric was selected based on empirical observations. The elements of  $V$  and  $V'$  corresponding to the top 5% and bottom 10% of the most recurring visual words in the image sequence are considered as being stop words [210] and are removed. The elements of  $V'$  corresponding to the 10% of the of the visual words with the lowest reliability are also removed.

As an example, figure 5.22 illustrates three visual words from the fisheye training set. The inverse document frequency and reliability values are shown in the captions for each. Note that the inverse document frequency values have been normalised such that the maximum range is from zero to one. Notice that although the inverse document frequency for the first two words is higher than the third, their reliability scores are far smaller. That is, these first two words were found through training to be unreliable when attempting to match across images.

### 5.3.4 Image Similarity

For each image in the fisheye sequence, the pSIFT keypoints were detected. Each keypoint was then assigned to a visual word using its descriptor and the visual vocabulary, where the Euclidean distance between the descriptor is used to measure the distance between the descriptor and each word in the vocabulary. Although not used, an efficient vocabulary tree algorithm designed for this purpose has been developed by Nistér and Stewénus [183]. The visual word vectors  $V_d$  and  $V'_d$  for each image were then computed. Two similarity matrices for the fisheye sequence were then found by measuring the similarity between all possible image pairs in the sequence using the

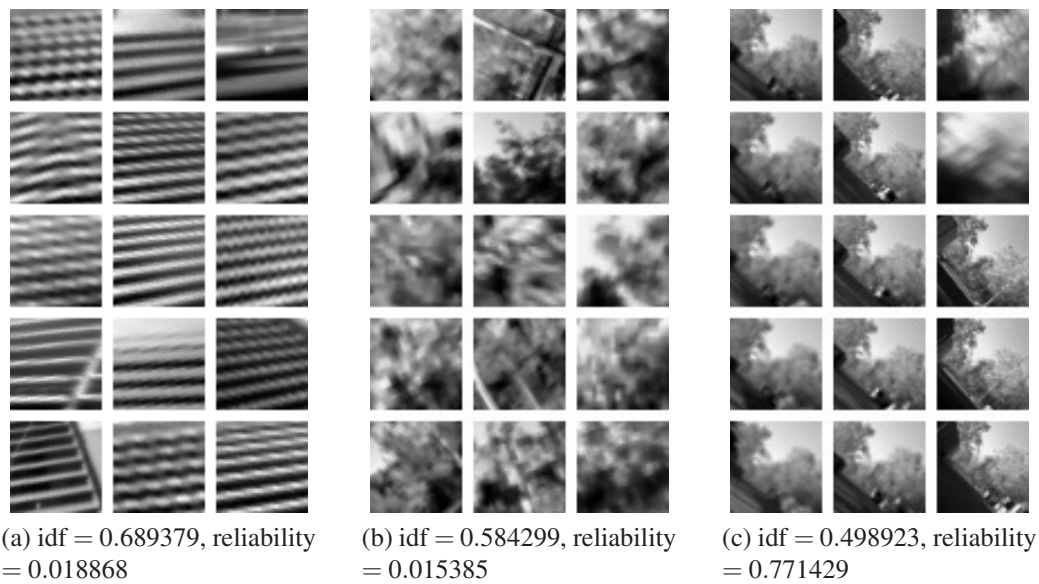


Figure 5.22: Example visual words in the vocabulary and their associated inverse document frequency (idf) and word reliability values. Note that the inverse document frequency values have been normalised to be in the range 0 to 1.

visual word vectors  $V_d$  and  $V'_d$ . This similarity is measured as the cosine angle between the visual word vectors.

### 5.3.5 Results and Discussion

The similarity matrices for the fisheye sequence in figure 5.1 found using the visual word vectors  $V_d$  and  $V'_d$  are given in figure .

The result of thresholding the similarity score for the similarity matrix found using  $V'$  is shown in figure 5.24. The dark off diagonal blobs indicate potential loop closure events, and the lines in figure 5.25a indicate these potential loop closure events on the map of the operating environment. Although this result alone would not be used as the only basis for deciding if the loop should be closed, it illustrates the ability of pSIFT to detect and describe the same keypoints in wide-angle images subject to large changes in camera pose. Figure 5.25b illustrates on the the image pairs corresponding to a potential loop closure detection with a similarity score of 0.79. The lines in figure 5.25c show the visual word correspondences in this image pair. To illustrate the advantages of wide-angle vision for loop closure detection, the images in Fig.5.25b illustrate by the dashed line the equivalent view that a perspective camera with a horizontal angle of view of  $60^\circ$  would obtain.



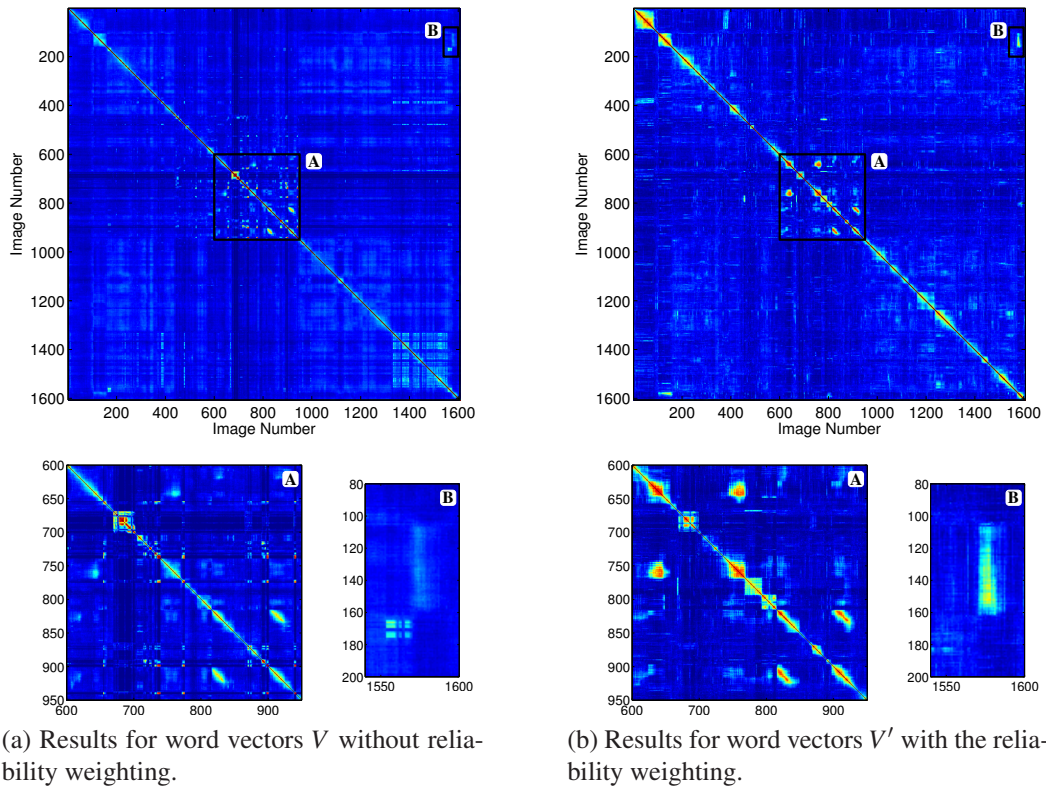


Figure 5.23: Similarity matrices for the fisheye sequence using visual word vectors with and without the reliability weighting.

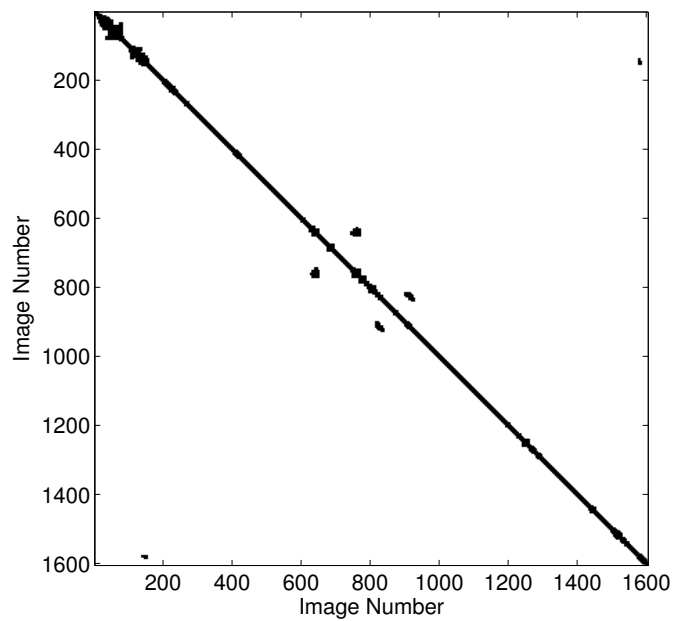
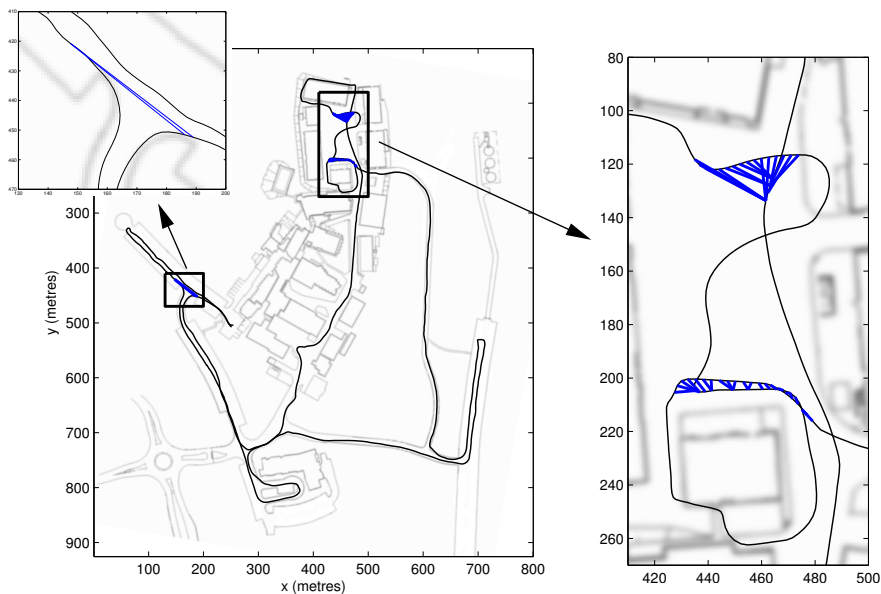
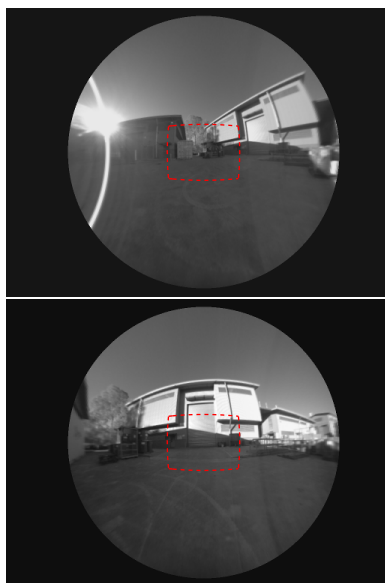


Figure 5.24: Thresholding of the cosine similarity score using the visual word vectors  $V'$  for a value of 0.6.

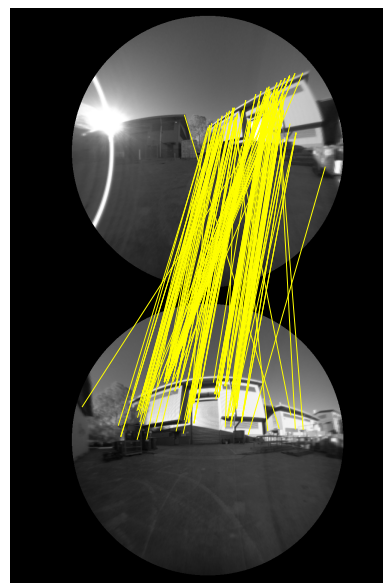




(a) Positions of loop closure detection.



(b) Query image (top) and retrieved image (bottom) (Cosine similarity = 0.79).



(c) Visual word correspondences

Figure 5.25: The location of the potential visual loop closure events. The lines indicate the matching images on the map. For the potential loop closure events in the upper right region of (a), an example pair of images is shown in (b) where the visual word correspondences are shown in (c). Note that (b) includes an overlay of the equivalent view that a typical perspective camera with a horizontal angle of view of  $60^\circ$  would obtain.

## 5.4 Conclusions

The new pSIFT keypoint detector was applied to common vision-based localisation tasks in this chapter including visual odometry and visual place recognition. Various constraints were used to estimate the visual odometry for two wide-angle images sequences, and the effect of frame-rate selection on the accuracy of these estimates were compared. A fixed frame-rate was used with estimates camera egomotion between successive frames in the sequences, and variable frame-rate was used which automatically selects the frames used based on a minimum number of keypoints that can be tracked between the frames. The algorithm used for the variable frame-rate method was based on that of Mouragnon et al [170]. Visual place recognition results were also found using a fisheye image sequence which included loop closure events.

Visual odometry estimates were first found for the Hyperion sequence using a ground plane constraint, which means the world points associated with all pSIFT keypoint correspondences were assumed to be coplanar. A Euclidean ground plane constraint was first used where the camera's principal axis was assumed to be orthogonal to the ground plane. The visual odometry estimates were then found using the Triggs ground plane constraint which relaxes this constraint regarding the precise alignment of the camera's principal axis with respect to the ground plane. It was observed that the standard Direct Linear Transform (DLT) used to estimate the camera egomotion for each of these constraints was sensitive to the effective field of view of the camera. Two modified versions of the standard DLT were formulated, and they vary from the standard DLT in their choice of keypoint coordinates used. The coordinates used are the weighted spherical coordinates of keypoint correspondences, where this weighting is a function of the uncertainty of keypoint positions on the sphere relative to the uncertainty in their positions found during detection. Both were shown to be far less sensitive to the effective camera field of view than the standard DLT and were able to find accurate linear estimates of camera egomotion. Two cost function were then used to optimise the initial linear estimate based on the transfer error and geometric error. Both were shown to produce similar results and improve the accuracy of the visual odometry estimates, in particular the Triggs ground plane constraint. The relative accuracy of the visual odometry estimates found for each of the Euclidean and Triggs ground plane constraints were found to be similar using a fixed frame-rate and a variable frame-rate.

The visual odometry estimates were then found for the Hyperion and fisheye image sequences using a generalised algorithm which made no assumptions regarding the

position of scene points or camera motion. The ability to find an accurate visual odometry estimate using a fixed frame-rate and a variable frame-rate were compared for the Hyperion sequence, and the results indicated that the variable frame-rate method was superior. This results gives strong evidence to show that for a generalised visual odometry algorithm, that accuracy of the egomotion estimate between frames improves as the change in pose between frames increases. The ability of pSIFT to detect and correctly match keypoints between successive images in the sequence enabled this change in pose between the automatically selected frames using the variable frame-rate algorithm to be large. For the Hyperion sequence, the generalised visual odometry algorithm was able to find an accurate estimate of the true vehicle path using the variable frame-rate, and was able to reliably resolve the magnitude of the camera translation using only vision. The results for the fisheye sequence using the variable frame-rate again showed reasonably accurate results, especially considering that the length of the transit exceeded 4 kilometres. However, the ability to reliably resolve the magnitude of the camera translation between frames for this sequence was limited. The GPS ground truth data was therefore used as a virtual encoder for this purpose. However, the estimate of the camera rotation and direction of translation between frames was accurately estimated using only vision.

Finally, visual place recognition results were presented for the fisheye sequence using the “Video Google” system proposed by Sivic and Zisserman in [210] with the addition of a word reliability metric. This system is an appearance only method which finds the similarity between any two images using only the keypoints detected in the images. The results found indicated that correct loop closure events could be identified, that is, previously visited places in the transit could be recognised. This results showed that the pSIFT keypoint detector was able to detect, and reliably describe, the same keypoints in different wide-angle images separated by a large change in pose.

# Chapter 6

## Conclusions

As discussed in the introduction in chapter 1, the large field of view of wide-angle cameras makes them ideal for vision based localisation tasks. However, these tasks typically require keypoints to be detected and matched between images separated by a large change in camera pose, that is, keypoints must be detected and matched across wide-baselines. Most existing methods used for this purpose have been designed for narrow field of view perspective cameras.

This principal question addressed in this thesis is:

*‘Can a method of wide-baseline keypoint detection and matching be found suited for use with any wide-angle camera for vision-based localisation, including visual odometry and visual place recognition?’*

In answering this question, a comprehensive review of wide-angle image formation was presented. The important outcome of this review is that most wide-angle cameras can be modelled as central projection, and this means that the distorted images they produce can be back projected to effectively undistorted functions on the sphere. As discussed by Daniilidis et al [56], image processing algorithms formulated as operations on the sphere are ideal for central projection wide-angle images as they are invariant to the radial distortion in the image. This work by Daniilidis et al was one of the inspirations for the work in this thesis.

Before attempting to develop a method of wide-baseline keypoint detection and matching for wide-angle images, a review of the state of the art was conducted. It was found that to successfully detect and match keypoints across wide-baselines, it was necessary to detect and describe keypoints in a way that was invariant to projective changes between the images. Scale-invariant keypoint detecting algorithms were found

to be ideally suited for this, especially those based on the scale-space framework. The Scale Invariant Feature Transform (SIFT) was identified as one of the best performing keypoint detectors through a review of comparative works. However, the limitation of applying SIFT directly to wide-angle images is the inability to account for image distortion. It was proposed that SIFT could be reformulated as an image processing algorithm on the sphere.

Two variants of SIFT were developed, termed spherical SIFT (sSIFT) and parabolic SIFT (pSIFT), that were suited for wide-angle cameras. The methods of keypoint detection and description used by SIFT were reformulated as operations on the sphere. Both used the solution of the heat diffusion equation on the sphere solved by Bülow [31] to find the scale-space representations of a wide-angle image. The solution is the convolution of the image mapped to the sphere with the spherical Gaussian. sSIFT and pSIFT differ in the method they used to implement this convolution: sSIFT in the spherical Fourier domain, and pSIFT as an approximate operation on the stereographic image plane. The abilities of sSIFT and pSIFT to reliably detect, describe and match keypoints between image pairs separated by a large change in pose were compared to that of SIFT (operating directly on wide-angle images and rectified perspective images) through extensive and systematic experiments. Overall, pSIFT was found to perform consistently well and better than SIFT.

To answer the research question, pSIFT was applied to vision-based localisation tasks. Accurate visual odometry estimates were able to be found for real outdoor wide-angle image sequences using various constraints on the position of world points and camera motion. Visual place recognition results were also presented, and pSIFT was shown to be able to reliably recognise previously visited locations. Visual odometry and place recognition are fundamental primitives in many vision-base localisation frameworks.

In conclusion, a method of keypoint detection and description was able to be developed that is suited for wide-baseline keypoint detection and matching with wide-angle images. It was able to perform better than its equivalent algorithm designed for perspective images (SIFT) in systematic experiments, and was shown to be suited for vision-based localisation tasks such as visual odometry and visual place recognition.

## 6.1 Answers to questions posed

In answering the principal question, a number of additional questions were posed in chapter 1. The thesis was structured on these research questions, and they are addressed here:

1. *What types of wide-angle cameras exist, what are the methods of modelling the distortion in the image, and how can they be calibrated to resolve the camera's intrinsic parameters?*

A review of wide-angle cameras having a field of view near or in excess of a full hemisphere was presented in chapter 2, and they include catadioptric and fisheye cameras. A review of the parametric camera models used to describe image formation with catadioptric and fisheye cameras was also presented in the same chapter along with the methods used to calibrate wide-angle cameras. Catadioptric cameras use a camera and reflective surface to obtain a wide-field of view image, and they are typically designed to follow some predefined model of image formation. On the other hand, fisheye camera models are typically selected empirically — a review of the common fisheye camera models was presented. Auto-calibration, full-range, and plumb line calibration algorithms can all be used to find the camera intrinsic parameters. In most cases, wide-angle cameras are considered to have a single effective viewpoint, which means that the images they produce can be mapped to the unit view sphere centred at the single effective viewpoint of the camera. These images can also be converted to geometrically correct perspective images.

2. *What existing methods of wide-baseline keypoint detection are suitable for vision-based localisation?*

This question was addressed in chapter 3. Scale-invariant, and scale and affine invariant keypoint detectors are used for wide-baseline keypoint detection and matching. They can detect and describe the same keypoints in two different images in a manner invariant to scale change between the images and small projective deformations. The scale-invariant keypoint detectors are most frequently used for vision based localisation. The Scale Invariant Feature Transform (SIFT) [142] is the most frequently used of the keypoint detectors for vision based localisation applications, although more recently the Speeded-Up Robust Features algorithm [17] has gained popularity due to its increase speed over SIFT.

3. *What are the limitations of applying these existing methods to wide-angle images*

*and how, if possible, can they be adapted to suit wide-angle images?*

Towards the end of chapter 3, the limitations of applying existing algorithms such as SIFT directly to wide-angle images was identified as their inability to account for the radial distortion in the image. One could always convert the wide-angle image to a perspective image before applying these algorithms, however, this conversion introduces severe interpolation artifacts. Perspective projection is also unable to produce an image with a field of view in excess of a full hemisphere. Another alternate that is frequently used is to convert a wide-angle image to a log-polar or cylindrical panoramic image and apply existing algorithms designed for use with perspective images to these. However, these log-polar and cylindrical panoramic images are not perspective, so applying algorithms designed for use with perspective images to these is not ideal.

4. *Can an alternative to existing wide-baseline keypoint detection and matching algorithms be developed which is more suitable for use with wide-angle images?*

The sSIFT and pSIFT keypoint detectors were developed in chapter 4. They are each variants of SIFT designed for wide-angle cameras, and they are able to detect and describe keypoints in a way that is invariant to wide-angle image distortion. The ability of sSIFT and pSIFT to match keypoints in images separated by large change in camera pose (wide-baseline separation) was compared to that of SIFT operating directly on wide-angle and rectified perspective views in experiments. Overall, pSIFT was found in the experiments conducted to be the best solution for wide-baseline keypoint detection and matching in wide-angle images.

5. *Assuming a suitable alternative can be found, can it be used to obtain accurate visual odometry estimates for a mobile robot, and if so, what are the effects of increasing the change in pose between views with respect to accuracy?*

Chapter 5 presented visual odometry results using the new pSIFT keypoint detector with two wide-angle images sequences. Using various constraints on the position of world points and camera motion, accurate visual odometry estimates were found for each image sequence. A variable frame-rate algorithm based on that of Mouragnon et al [170] was used to automatically select the frames used to compute camera egomotion. The relative accuracy of these estimates were compared to those using a fixed frame-rate. When constraints were placed on the position of the world points (ground plane constraint), there was found to be only minor variations in the results. For generalised visual odometry where no assumptions/constraints are made regarding the position of the world points or



camera motion, the variable frame-rate method was found to be superior to the fixed frame-rate method. This result highlighted the importance of increasing the change in pose between the frames used to compute the camera egomotion, and why wide-baseline keypoint detection and matching algorithms are used in many vision odometry systems.

6. *Assuming again that a suitable alternative can be found, can it be used to obtain robust visual place recognition using wide-angle cameras?*

Visual place recognition results were presented for an outdoor fisheye image sequence in chapter 5 using pSIFT. The place recognition algorithm used was the appearance based “Video Google” system of Sivic and Zisserman [210] with the addition of a novel word reliability metric. The results indicated that potential place recognition (loop-closure) events could correctly be identified. This showed that pSIFT was able to correctly detect and describe the same keypoints in different wide-angle images taken at very different viewpoints.

## 6.2 Contributions of the Thesis

The contributions of the thesis are summarised as:

- A review of the advantages of using wide-angle cameras for visual odometry and visual place recognition was presented. Wide-angle cameras are able to find more accurate estimates of camera egomotion when compared to narrow field of view cameras, and they are able to capture images of the same scene separated by a large change in camera pose.
- A review of image formation with wide-angle cameras and camera calibration was given, including a detailed review of the camera models used to describe image formation.
- The development of a modified ‘plumb line’ calibration algorithm suited for central projection wide-angle cameras was developed. The algorithm can be used to calibrate for a camera’s intrinsic parameters using any model of image formation. A robust grid point detection algorithm was also developed which operates in parallel with calibration.
- Review of the state of the art in keypoint detection and description, and their applications to wide-baseline matching. Scale-invariant keypoint detection and



description algorithms such as SIFT were identified as being suited for this application to wide-baseline matching. However, most of these methods are designed for use with perspective cameras, and the limitations of applying them directly to wide-angle images was identified as their inability to account for camera distortion. The limitation of applying them to either rectified perspective or (log-polar or cylindrical) panoramic images was also identified. This limitation is the severe interpolation artifacts that are introduced during these conversions. Furthermore, panoramic images are not perspective, so applying algorithms designed for perspective images to them is not ideal.

- Development of sSIFT and pSIFT, two methods of scale-invariant keypoint detection suited for wide-angle cameras that can be used for wide-baseline keypoint detection and matching with wide-angle images. Both are variants of SIFT that are reformulated as image processing algorithms on the sphere. Both define scale-space for wide-angle images as the convolution of the image mapped to the sphere with the spherical Gaussian. The methods of keypoint detection and description used by sSIFT and pSIFT are invariant to the radial distortion in the image.
- A method to estimate the bandwidth of a wide-angle image from the camera intrinsic parameters was introduced. This estimate is used to find the minimum sample rate used by sSIFT to convolve the image with the spherical Gaussian in the spherical Fourier domain. An anti-aliasing interpolation filter was also designed that can be used to minimise aliasing when the required sample rate exceeds the maximum computationally feasible value.
- An approximation spherical diffusion operation used to find the scale-space representations of wide-angle images. This approximate diffusion operation is implemented as an efficient convolution operation on the stereographic image plane. Although the scale-space images found using this approximation have a non-uniform scale, they were used successfully by pSIFT for scale-invariant keypoint detection in wide-angle images.
- Two coordinate weighting schemes (one iterative) were formulated for egomotion estimation with pSIFT keypoints and the Direct Linear Transform (DLT) using a ground plane constraint. They were compared to the 'standard' coordinate selections used for egomotion estimation using the DLT, for example normalised homogeneous coordinates. They were found to find egomotion estimates with either comparable or better accuracy and whose estimates were less influenced by the effective field of view of the camera.

- An investigation of the effect of frame-rate selection on the accuracy of visual odometry was conducted through experiments. For generalised visual odometry, a variable frame-rate method, which attempts to increase the change in pose between frames used to compute camera egomotion, was found to produce superior results to those found using a fixed frame-rate.
- A word reliability metric was developed and incorporated into the appearance based “Video Google” system used for visual place recognition. Experimental results were presented using pSIFT and this algorithm with an outdoor fisheye image sequence. The results showed that potential loop closure events could be correctly identified.

## 6.3 Further directions

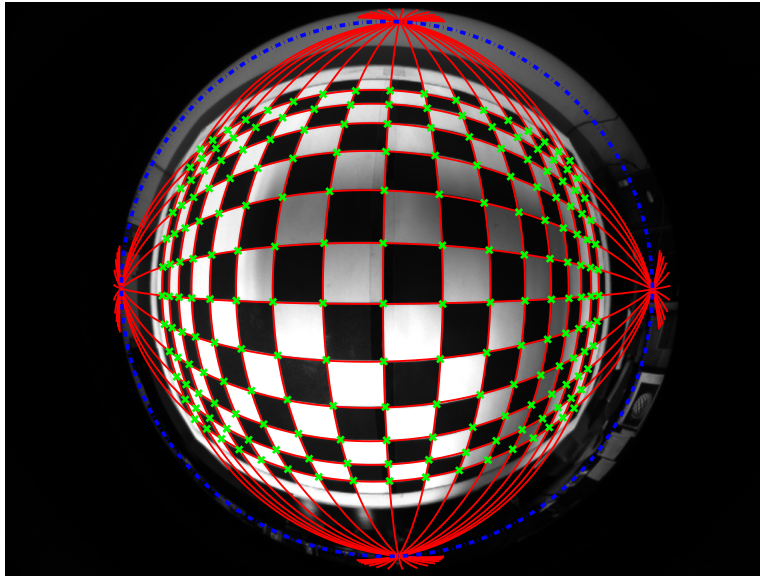
There were a number of further directions of research that were not explored in the thesis due to time constraints, and they include:

- The keypoint detectors developed in the thesis reformulated SIFT as an image processing algorithm on the sphere making them suited for wide-angle cameras. It would be of interest to reformulate other existing methods designed for use with perspective images, such as MSER and SURF, as image processing algorithms on the sphere. The relative performance of these algorithms tailored for use with wide-angle cameras could then be compared.
- pSIFT exploits the conformal nature of stereographic projection to approximate spherical diffusion efficiently on the stereographic image plane. A range of conformal mappings other than stereographic projection warrants further investigation, where each could be used to map a wide-angle image to a ‘conformal image plane’. This investigation would compare the relative advantages and disadvantages of each with respect to approximating spherical diffusion as a convolution operation on these conformal image planes.
- Re-evaluate the sSIFT keypoint detector with different hardware. In all the experiments conducted, the required sample rate for each wide-angle camera exceeded the maximum computationally feasible value. It would be of interest to re-evaluate sSIFT using hardware capable of using a sample rate greater than that used in the experiments.

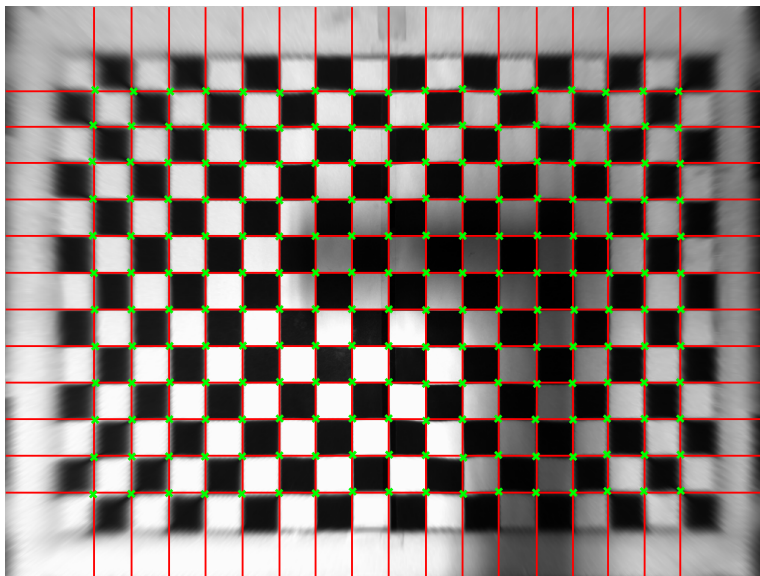
- The sSIFT keypoint detector uses s2kit to find the spherical harmonic functions, which requires sampling the image values on an equiangular  $\theta, \phi$  grid. Alternate methods, such as the technique developed by Chung et al in [41], have been developed for this purpose, and they do not require sampling the image on a  $\theta, \phi$  grid. These alternate methods should be explored in more detail and potentially implemented.
- The sSIFT keypoint detector maps the scale-space images on the sphere (sampled on an equiangular  $\theta, \phi$  grid) back to the original wide-angle image plane. This mapping introduces some form of interpolation artifacts. It may be more suitable to detect the keypoints in the scale-space images sampled on the  $\theta, \phi$  grid, thus preventing the introduction of interpolation artifacts.
- The keypoint detection stage of sSIFT and pSIFT map the greyscale intensity values within a keypoint's support region to fixed sized patch from which the descriptor is evaluated — this mapping requires an interpolation of the greyscale intensity function. Other methods used to evaluate the keypoints descriptors need to be explored which avoid this need for interpolation and the artifacts it can produce.
- There has recently been significant advancements in visual SLAM algorithms that can operate in large scale outdoor environments. It would be of interest to evaluate the performance of pSIFT in a visual SLAM framework.

# **Appendix A**

## **Calibration Results for Fisheye Images**

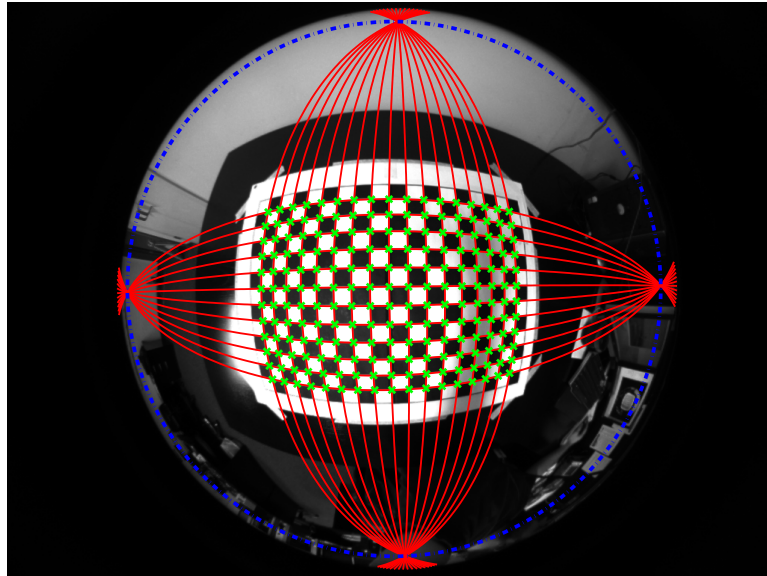


(a) Calibration results depicted on the original image plane ( $1024 \times 768$  pixels).

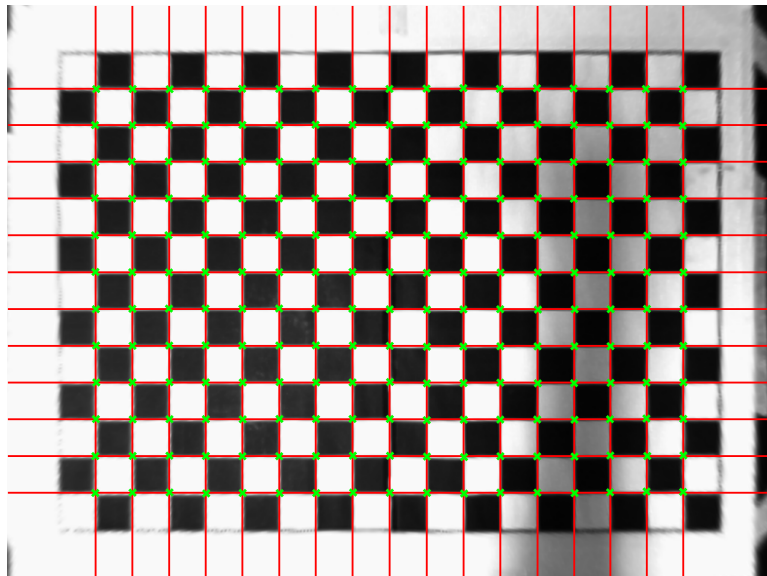


(b) Calibration results depicted on the orthonormal perspective plane ( $1024 \times 768$  pixels).

Figure A.1: Calibration results for image 1. The green crosses show the position of the grid points found using the grid point detection algorithm. The red lines illustrate the fitted great circle on the sphere, and the blue line the front-parallel horizon of the plane containing the planar checkerboard pattern.

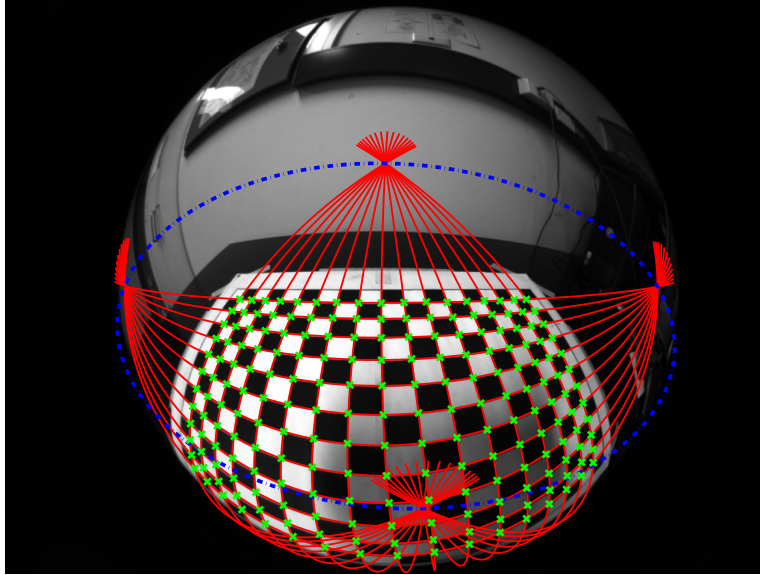


(a) Calibration results depicted on the original image plane ( $1024 \times 768$  pixels).

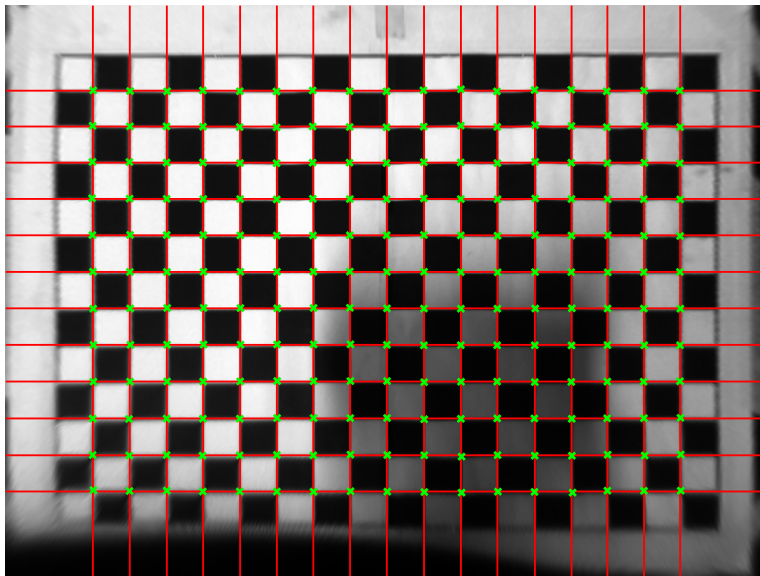


(b) Calibration results depicted on the orthonormal perspective plane ( $1024 \times 768$  pixels).

Figure A.2: Calibration results for image 2. The green crosses show the position of the grid points found using the grid point detection algorithm. The red lines illustrate the fitted great circle on the sphere, and the blue line the front-parallel horizon of the plane containing the planar checkerboard pattern.



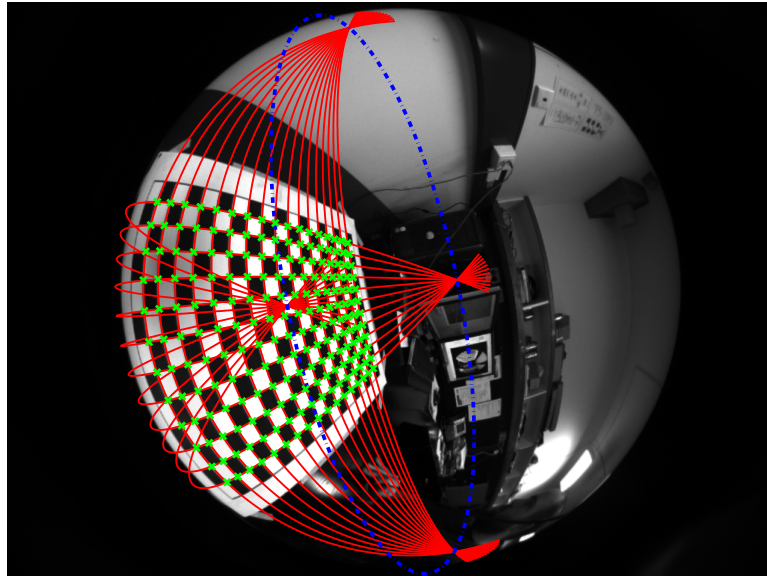
(a) Calibration results depicted on the original image plane ( $1024 \times 768$  pixels).



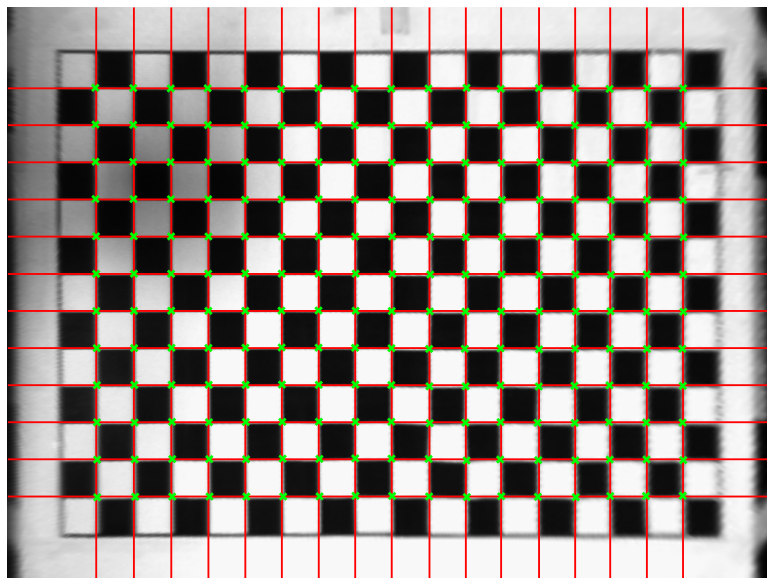
(b) Calibration results depicted on the orthonormal perspective plane ( $1024 \times 768$  pixels).

Figure A.3: Calibration results for image 3. The green crosses show the position of the grid points found using the grid point detection algorithm. The red lines illustrate the fitted great circle on the sphere, and the blue line the front-parallel horizon of the plane containing the planar checkerboard pattern.





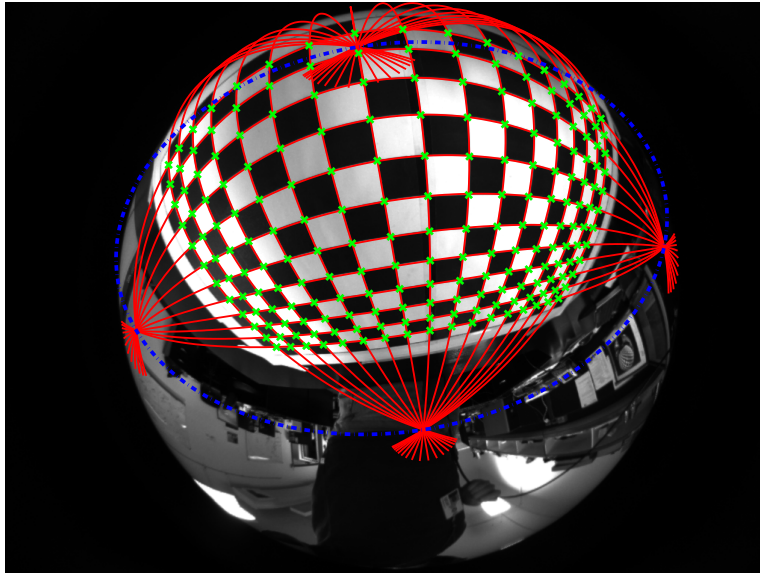
(a) Calibration results depicted on the original image plane ( $1024 \times 768$  pixels).



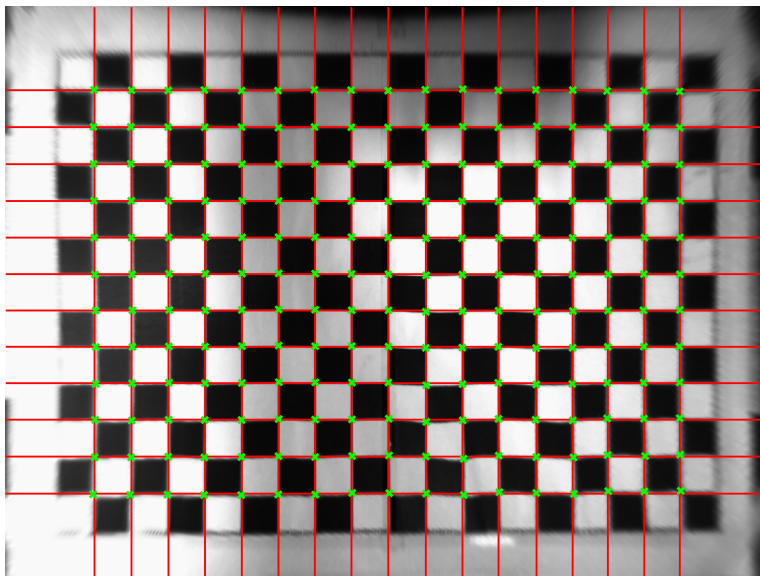
(b) Calibration results depicted on the orthonormal perspective plane ( $1024 \times 768$  pixels).

Figure A.4: Calibration results for image 4. The green crosses show the position of the grid points found using the grid point detection algorithm. The red lines illustrate the fitted great circle on the sphere, and the blue line the front-parallel horizon of the plane containing the planar checkerboard pattern.



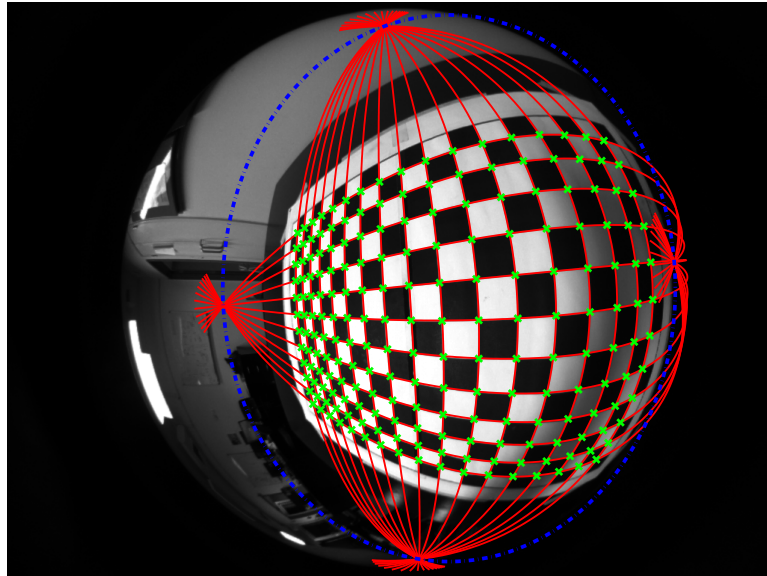


(a) Calibration results depicted on the original image plane ( $1024 \times 768$  pixels).

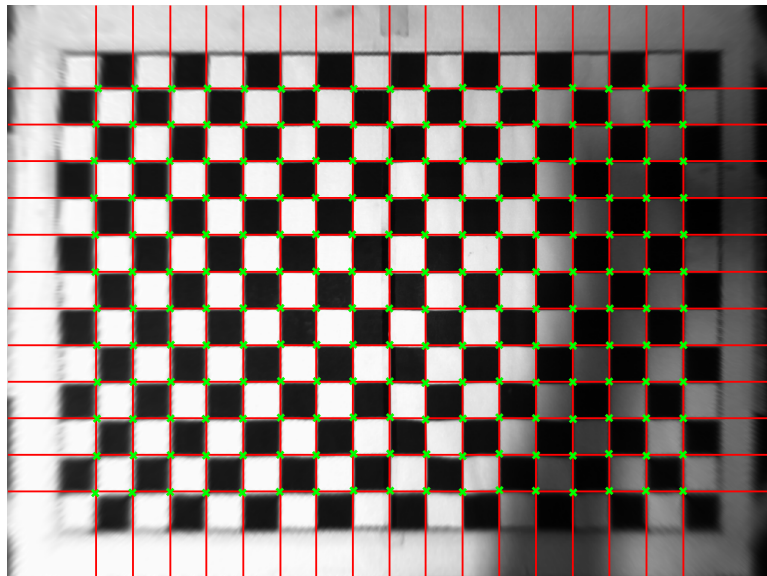


(b) Calibration results depicted on the orthonormal perspective plane ( $1024 \times 768$  pixels).

Figure A.5: Calibration results for image 5. The green crosses show the position of the grid points found using the grid point detection algorithm. The red lines illustrate the fitted great circle on the sphere, and the blue line the front-parallel horizon of the plane containing the planar checkerboard pattern.

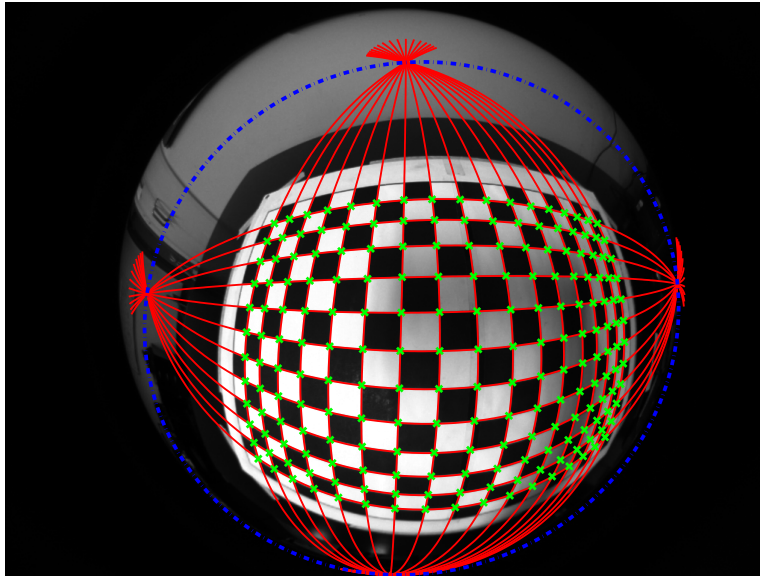


(a) Calibration results depicted on the original image plane ( $1024 \times 768$  pixels).

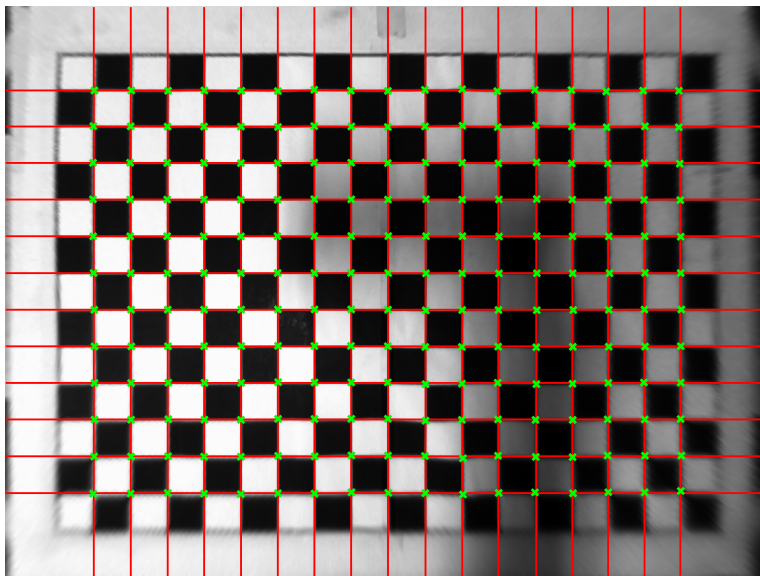


(b) Calibration results depicted on the orthonormal perspective plane ( $1024 \times 768$  pixels).

Figure A.6: Calibration results for image 6. The green crosses show the position of the grid points found using the grid point detection algorithm. The red lines illustrate the fitted great circle on the sphere, and the blue line the front-parallel horizon of the plane containing the planar checkerboard pattern.

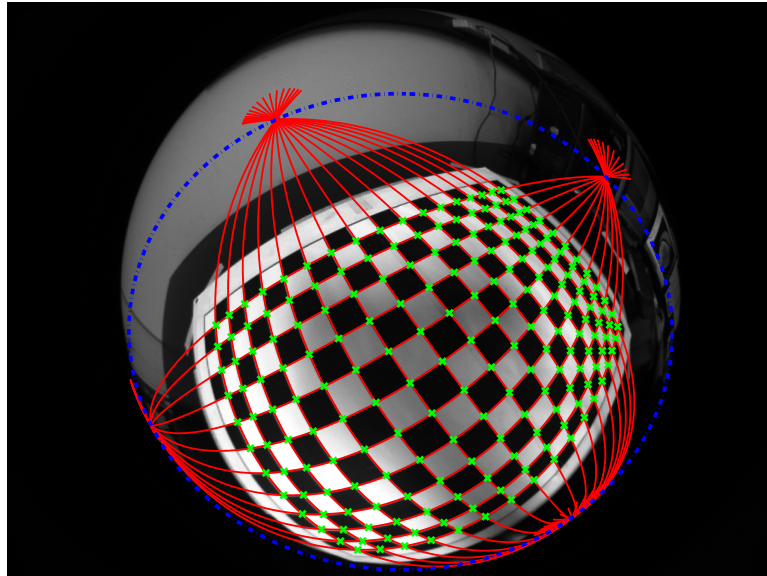


(a) Calibration results depicted on the original image plane ( $1024 \times 768$  pixels).

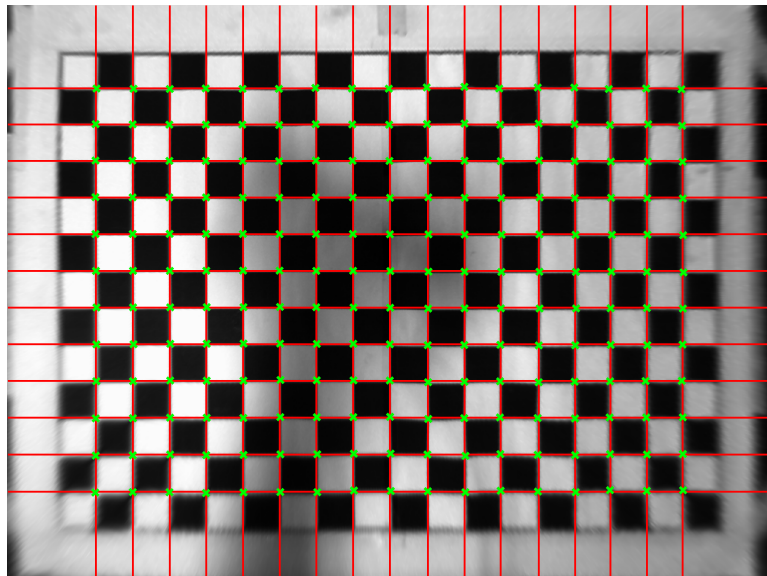


(b) Calibration results depicted on the orthonormal perspective plane ( $1024 \times 768$  pixels).

Figure A.7: Calibration results for image 7. The green crosses show the position of the grid points found using the grid point detection algorithm. The red lines illustrate the fitted great circle on the sphere, and the blue line the front-parallel horizon of the plane containing the planar checkerboard pattern.

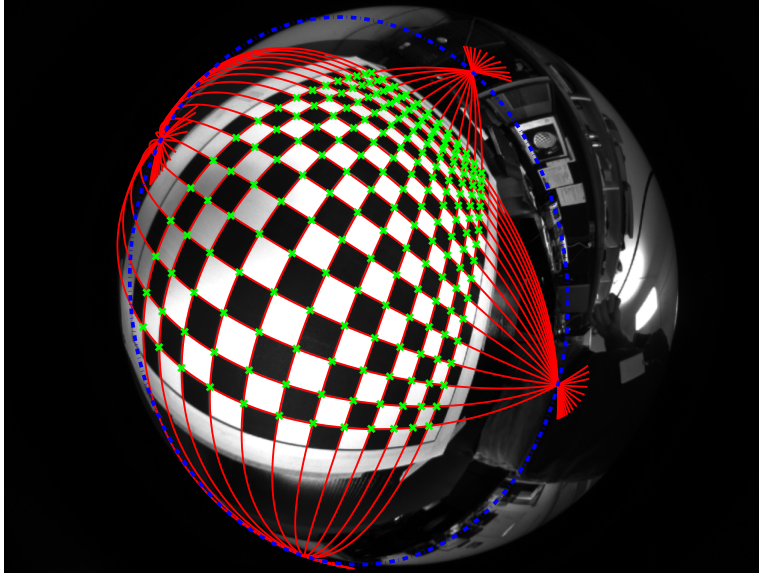


(a) Calibration results depicted on the original image plane ( $1024 \times 768$  pixels).

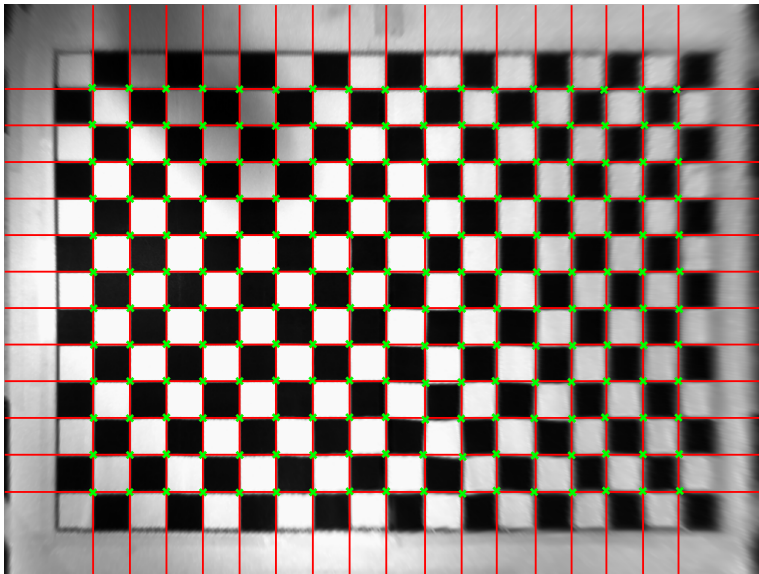


(b) Calibration results depicted on the orthonormal perspective plane ( $1024 \times 768$  pixels).

Figure A.8: Calibration results for image 8. The green crosses show the position of the grid points found using the grid point detection algorithm. The red lines illustrate the fitted great circle on the sphere, and the blue line the front-parallel horizon of the plane containing the planar checkerboard pattern.



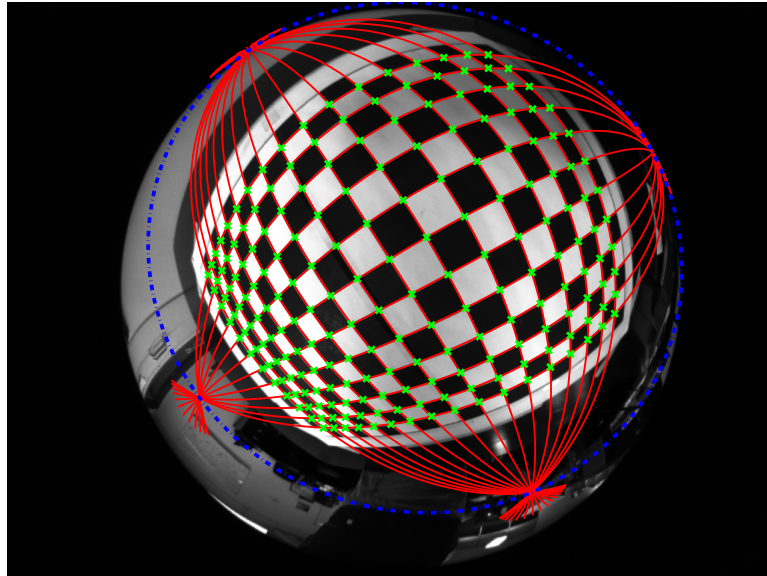
(a) Calibration results depicted on the original image plane ( $1024 \times 768$  pixels).



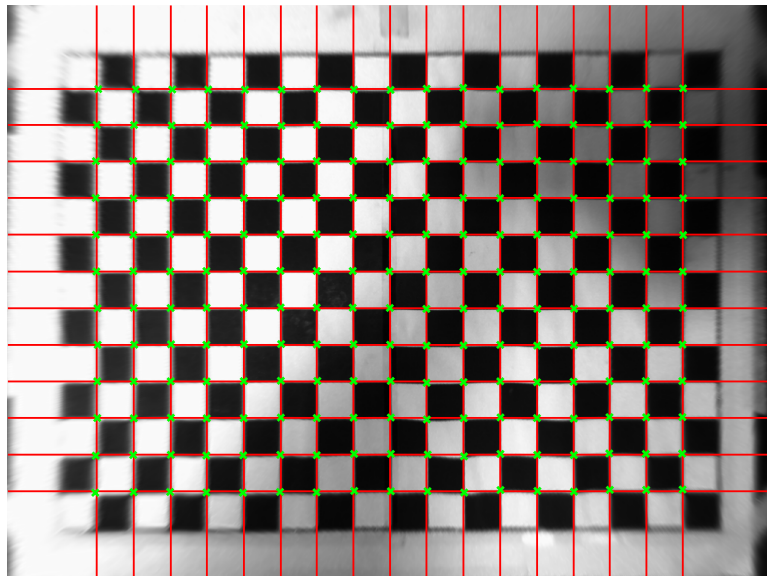
(b) Calibration results depicted on the orthonormal perspective plane ( $1024 \times 768$  pixels).

Figure A.9: Calibration results for image 9. The green crosses show the position of the grid points found using the grid point detection algorithm. The red lines illustrate the fitted great circle on the sphere, and the blue line the front-parallel horizon of the plane containing the planar checkerboard pattern.





(a) Calibration results depicted on the original image plane ( $1024 \times 768$  pixels).



(b) Calibration results depicted on the orthonormal perspective plane ( $1024 \times 768$  pixels).

Figure A.10: Calibration results for image 10. The green crosses show the position of the grid points found using the grid point detection algorithm. The red lines illustrate the fitted great circle on the sphere, and the blue line the front-parallel horizon of the plane containing the planar checkerboard pattern.



## Appendix B

# Spherical Harmonic Expansion of the Spherical Dirac Function

The spherical Dirac function  $\delta_{\mathbb{S}^2}$  is defined as

$$f(\mathbf{n}) = \int_{\eta \in \mathbb{S}^2} f(\eta) \delta_{\mathbb{S}^2}(\theta, \phi) d\eta, \quad f \in L^2(\mathbb{S}^2), \quad (\text{B.1})$$

where  $\theta \in [0, \pi)$  is an angle of colatitude,  $\phi \in [0, 2\pi)$  is an angle of longitude,  $\mathbf{n} = (0, 0, 1)^T$  is the north pole, and  $d\eta = \sin(\theta) d\theta d\phi$ . Any square integratable function  $f$  on the sphere can be expanded into spherical harmonics as

$$f = \sum_{l \in \mathbb{N}} \sum_{|m| \leq l} \hat{f}_l^m Y_l^m, \quad \hat{f}_l^m = \int_{\mathbb{S}^2} f(\eta) \overline{Y_l^m}(\eta) d\eta, \quad (\text{B.2})$$

where  $\overline{Y_l^m}$  denotes the complex conjugate of the spherical harmonic functions  $Y_l^m(\theta, \phi)$ :

$$Y_l^m(\theta, \phi) = \sqrt{\frac{2l+1}{4\pi} \frac{(l-m)!}{(l+m)!}} P_l^m(\cos(\theta)) e^{im\phi}, \quad l \in \mathbb{N}, |m| \leq l, \quad (\text{B.3})$$

where  $P_l^m$  are the associated Legendre polynomials

$$P_l^m(x) = \frac{(-1)^m (1-x^2)^{\frac{m}{2}}}{2^l l!} \frac{d^{l+m}}{dx^{l+m}} (x^2 - 1)^l. \quad (\text{B.4})$$

The spectrum  $\hat{\delta}_{\mathbb{S}^2}$  of the spherical Dirac function  $\delta_{\mathbb{S}^2}$  is obtained as

$$(\hat{\delta}_{\mathbb{S}^2})_l^m = \int_{\mathbb{S}^2} \delta_{\mathbb{S}^2}(\eta) \overline{Y_l^m}(\theta, \phi) d\eta. \quad (\text{B.5})$$



From the definition of  $\delta_{\mathbb{S}^2}$  in equation B.1, equation B.5 becomes

$$(\hat{\delta}_{\mathbb{S}^2})_l^m = \bar{Y}_l^m(\mathbf{n}). \quad (\text{B.6})$$

There is an apparent singularity in trying to evaluate  $\bar{Y}_l^m$  at the north pole as the spherical harmonic function,  $Y_l^m(\theta, \phi)$ , is parameterised by spherical coordinates  $\theta, \phi$  —  $\phi$  is not defined at the pole. However, as the angle  $\theta = 0$  at the north pole, the associated Legendre polynomials  $P_l^m(\cos(\theta))$  evaluated for  $\theta = 0$  are

$$P_l^m(\cos(\theta = 0)) = \begin{cases} 1 & m = 0 \\ 0 & m \neq 0 \end{cases} \quad (\text{B.7})$$

It follows then that irrespective of the angle  $\phi$ ,

$$(\hat{\delta}_{\mathbb{S}^2})_l^m = \bar{Y}_l^m(\mathbf{n}) = \begin{cases} \sqrt{\frac{2l+1}{4\pi}} & m = 0 \\ 0 & m \neq 0 \end{cases} \quad (\text{B.8})$$

From equation B.2, the spherical harmonic expansion of  $\delta_{\mathbb{S}^2}$  is

$$\delta_{\mathbb{S}^2}(\boldsymbol{\eta}) = \sum_{l \in \mathbb{N}} (\hat{\delta}_{\mathbb{S}^2})_l^m Y_l^0(\boldsymbol{\eta}) \quad (\text{B.9})$$

$$= \sum_{l \in \mathbb{N}} \sqrt{\frac{2l+1}{4\pi}} Y_l^0(\boldsymbol{\eta}), \quad (\text{B.10})$$

which is the summation of only the zonal harmonic function  $Y_l^0$  as  $(\hat{\delta}_{\mathbb{S}^2})_l^m = 0 \forall m \neq 0$ .

## Appendix C

# Computation of Discrete First Order Derivatives and Hessian Matrix for Keypoint Interpolation

For a keypoint detected at position  $\mathbf{u} = (u, v)^T$  in the difference of Gaussian image  $\mathcal{D}_{\mathbb{S}^2}(\cdot; kt_i)$ , the quadratic interpolation scheme proposed by Brown and Lowe [29] is used to improve the accuracy of a keypoint's position and scale. The interpolated position  $\hat{\mathbf{x}}$  of the keypoint from the origin  $\mathbf{x} = (u, v, kt)^T$  is

$$\hat{\mathbf{x}} = -\frac{\partial^2 \mathcal{D}_{\mathbb{S}^2}}{\partial \mathbf{x}}^{-1} \frac{\partial \mathcal{D}_{\mathbb{S}^2}}{\partial \mathbf{x}}, \quad (\text{C.1})$$

where  $\frac{\partial \mathcal{D}_{\mathbb{S}^2}}{\partial \mathbf{x}}$  is the  $3 \times 1$  vector of first order partial derivatives computed at the point  $\mathcal{D}_{\mathbb{S}^2}(u, v; kt_i)$ , and  $\frac{\partial^2 \mathcal{D}_{\mathbb{S}^2}}{\partial \mathbf{x}^2}$  is the  $3 \times 3$  Hessian matrix computed at the point  $\mathcal{D}_{\mathbb{S}^2}(u, v; kt_i)$ .

The vector  $\frac{\partial \mathcal{D}_{\mathbb{S}^2}}{\partial \mathbf{x}}$  is obtained as

$$\frac{\partial \mathcal{D}_{\mathbb{S}^2}}{\partial \mathbf{x}} = \begin{bmatrix} \mathcal{D}_{\mathbb{S}^2}(u, v; kt_i)_u \\ \mathcal{D}_{\mathbb{S}^2}(u, v; kt_i)_v \\ \mathcal{D}_{\mathbb{S}^2}(u, v; kt_i)_{kt} \end{bmatrix} \quad (\text{C.2})$$

$$= \begin{bmatrix} \frac{1}{2} \{ \mathcal{D}_{\mathbb{S}^2}(u+1, v; kt_i) - \mathcal{D}_{\mathbb{S}^2}(u-1, v; kt_i) \} \\ \frac{1}{2} \{ \mathcal{D}_{\mathbb{S}^2}(u, v+1; kt_i) - \mathcal{D}_{\mathbb{S}^2}(u, v-1; kt_i) \} \\ \frac{1}{2} \{ \mathcal{D}_{\mathbb{S}^2}(u, v; kt_{i+1}) - \mathcal{D}_{\mathbb{S}^2}(u, v; kt_{i-1}) \} \end{bmatrix}. \quad (\text{C.3})$$

The Hessian matrix  $\frac{\partial^2 \mathcal{D}_{\mathbb{S}^2}}{\partial \mathbf{x}^2}$  is obtained as

$$\frac{\partial^2 \mathcal{D}_{\mathbb{S}^2}}{\partial \mathbf{x}^2} = \begin{bmatrix} \mathcal{D}_{\mathbb{S}^2}(u, v; kt_i)_{uu} & \mathcal{D}_{\mathbb{S}^2}(u, v; kt_i)_{uv} & \mathcal{D}_{\mathbb{S}^2}(u, v; kt_i)_{ukt} \\ \mathcal{D}_{\mathbb{S}^2}(u, v; kt_i)_{uv} & \mathcal{D}_{\mathbb{S}^2}(u, v; kt_i)_{vv} & \mathcal{D}_{\mathbb{S}^2}(u, v; kt_i)_{vkt} \\ \mathcal{D}_{\mathbb{S}^2}(u, v; kt_i)_{ukt} & \mathcal{D}_{\mathbb{S}^2}(u, v; kt_i)_{vkt} & \mathcal{D}_{\mathbb{S}^2}(u, v; kt_i)_{ktkt} \end{bmatrix}, \quad (\text{C.4})$$

where

$$\mathcal{D}_{\mathbb{S}^2}(u, v; kt_i)_{uu} = \mathcal{D}_{\mathbb{S}^2}(u+1, v; kt_i) + \mathcal{D}_{\mathbb{S}^2}(u-1, v; kt_i) - 2\mathcal{D}_{\mathbb{S}^2}(u, v; kt_i), \quad (\text{C.5})$$

$$\mathcal{D}_{\mathbb{S}^2}(u, v; kt_i)_{vv} = \mathcal{D}_{\mathbb{S}^2}(u, v+1; kt_i) + \mathcal{D}_{\mathbb{S}^2}(u, v-1; kt_i) - 2\mathcal{D}_{\mathbb{S}^2}(u, v; kt_i), \quad (\text{C.6})$$

$$\mathcal{D}_{\mathbb{S}^2}(u, v; kt_i)_{ktkt} = \mathcal{D}_{\mathbb{S}^2}(u, v; kt_{i+1}) + \mathcal{D}_{\mathbb{S}^2}(u, v; kt_{i-1}) - 2\mathcal{D}_{\mathbb{S}^2}(u, v; kt_i), \quad (\text{C.7})$$

$$\mathcal{D}_{\mathbb{S}^2}(u, v; kt_i)_{uv} = \frac{1}{4} [(\mathcal{D}_{\mathbb{S}^2}(u+1, v+1; kt_i) - \mathcal{D}_{\mathbb{S}^2}(u-1, v+1; kt_i)) - \quad (\text{C.8})$$

$$(\mathcal{D}_{\mathbb{S}^2}(u+1, v-1; kt_i) - \mathcal{D}_{\mathbb{S}^2}(u-1, v-1; kt_i))], \quad (\text{C.9})$$

$$\mathcal{D}_{\mathbb{S}^2}(u, v; kt_i)_{ukt} = \frac{1}{4} [(\mathcal{D}_{\mathbb{S}^2}(u+1, v; kt_{i+1}) - \mathcal{D}_{\mathbb{S}^2}(u-1, v; kt_{i+1})) - \quad (\text{C.10})$$

$$(\mathcal{D}_{\mathbb{S}^2}(u+1, v; kt_{i-1}) - \mathcal{D}_{\mathbb{S}^2}(u-1, v; kt_{i-1}))], \quad (\text{C.11})$$

$$\mathcal{D}_{\mathbb{S}^2}(u, v; kt_i)_{vkt} = \frac{1}{4} [(\mathcal{D}_{\mathbb{S}^2}(u, v+1; kt_{i+1}) - \mathcal{D}_{\mathbb{S}^2}(u, v-1; kt_{i+1})) - \quad (\text{C.12})$$

$$(\mathcal{D}_{\mathbb{S}^2}(u, v+1; kt_{i-1}) - \mathcal{D}_{\mathbb{S}^2}(u, v-1; kt_{i-1}))]. \quad (\text{C.13})$$

# Bibliography

- [1] Motilal Agrawal and Kurt Konolige. Real-time localization in outdoor environments using stereo vision and inexpensive gps. In *International Conference on Pattern Recognition (ICPR)*, 2006.
- [2] Motilal Agrawal and Kurt Konolige. Rough terrain visual odometry. In *International Conference on Advanced Robotics (ICAR)*, August 2007.
- [3] Henrik Andreasson, Tom Duckett, and Achim Lilienthal. Mini-SLAM: minimalistic visual SLAM in large-scale environments based on a new interpretation of image similarity. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 4096–4101, Roma, Italy, April 2007.
- [4] Adrien Angeli, Stéphane Doncieux, and David Filliat. Real-time visual loop-closure detection. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 1842–1847, Pasadena, USA, May 2008.
- [5] J. P. Antoine and P. Vandergheynst. Wavelets on the 2-sphere: A group-theoretical approach. *Applied and Computational Harmonic Analysis*, 7:262–291, 1999.
- [6] Jean Babaud, Andrew Witkin, Michel Baudin, and Richard Duda. Uniqueness of the gaussian kernel for scale-space filtering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8(1):26–33, January 1986.
- [7] Simon Baker and Shree Nayar. A theory of single-viewpoint catadioptric image formation. *International Journal of Computer Vision*, 35(2):175–196, 1999.
- [8] Hynek Bakstein and Tomáš Pajdla. Calibration of a fish eye lens with field of view larger than  $180^\circ$ . In *Proceedings of the CVWW*, pages 276–285, 2002.
- [9] Hynek Bakstein and Tomáš Pajdla. Panoramic mosaicing with 180 field of view lens. In *Proceedings of the third Workshop on Omnidirectional Vision*, 2002.

- [10] Jasmine Banks. *Reliability Analysis of Transform-Based Stereo Matching Techniques, and a New Matching Constraint*. PhD thesis, Queensland University of Technology, April 2000.
- [11] Joao Barreto. A unifying geometric representation for central projection systems. *Computer Vision and Image Understanding*, 103(3):208–217, Sep 2006.
- [12] Joao Barreto and Helder Araujo. Geometric properties of central catadioptric line images and their application in calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(8):1327–1333, August 2005.
- [13] Joao Barreto and Kostas Daniilidis. Unifying image plane liftings for central catadioptric and dioptric cameras. *Imaging Beyond the Pinhole Camera*, 2006.
- [14] Anup Basu and Sergio Licardie. Alternate models for fish-eye lenses. *Pattern Recognition Letters*, 10:433–441, 1995.
- [15] Jahannes Bauer, Miko Sünderhauf, and Peter Protzel. Comparing several implementations of two recently published feature detectors. In *International Conference on Intelligent and Autonomous Systems (IAV)*, Toulouse, France, 2007.
- [16] Adam Baumberg. Reliable feature matching across widely separated views. In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2000)*, 2000.
- [17] Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool. Speeded-up robust features (SURF). *Computer Vision and Image Understanding*, 110(3):346–359, June 2008.
- [18] Herbert Bay, Vittorio Ferrari, and Luc Van Gool. Wide-baseline stereo matching with line segments. In *Proceedings 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2005)*, pages 329–336, June 2005.
- [19] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. SURF: speeded up robust features. In *European Conference on Computer Vision*, 2006.
- [20] Shawn Becker and V. Michael Bove. Semiautomatic 3-D model extraction from uncalibrated 2-D camera views. In *Proc. of the SPIE Symposium on Electronic Imaging*, pages 447–461, 1995.

- [21] Erwan Bigorgne, Catherine Achard, and Jean Devars. A local color descriptor for efficient scene-object recognition. In *Proceedings of 11th International conference on Image Analysis and Processing*, pages 440–445, Palermo, September 2001.
- [22] O. Booij, B. Terwijn, Z. Zivkovic, and B. Kröse. Navigation using an appearance based topological map. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 3927–3932, Roma, Italy, April 2007.
- [23] M. Bosse, P. Newman, J. Leonard, and S. Teller. Simultaneous localisation and map building in large-scale cyclic environments using the atlas framework. *International Journal of Robotics Research*, 23(12):1113–1139, December 2004.
- [24] Michael Bosse and Robert Zlot. Map matching and data association for large-scale two-dimensional laser scan-based slam. *The International Journal of Robotics Research*, 27(6):667–691, 2008.
- [25] Christian Bräuer-Burchardt and Klaus Voss. A new algorithm to correct fish-eye- and strong wide-angle-lens-distortion from single images. In *Proceedings International Conference on Image Processing*, pages 225–228, October 2001.
- [26] Amy Briggs, Carrick Detweiler, Peter Mullen, and Daniel Scharstein. Scale-space features in 1D omnidirectional images. In *Omnivision*, 2004.
- [27] Duane.C. Brown. Close-range camera calibration. *Photogrammetric Engineering*, 1971.
- [28] M. Brown and D.G. Lowe. Recognising panoramas. In *International Conference on Computer Vision (ICCV 2003)*, Nice, France, October 2003.
- [29] Matthew Brown and David Lowe. Invariant features from interest point groups. In *Proceedings British Machine Vision Conference*, pages 656–665, Cardiff, Wales, September 2002.
- [30] Thomas Bülow. Multiscale image processing on the sphere. In Luc van Gool, editor, *Proc. 24th Symposium Pattern Recognition of the DAGM*, pages 609–617, 2002.
- [31] Thomas Bülow. Spherical diffusion for 3D surface smoothing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(12):1650–1654, December 2004.

- [32] Roland Bunschoten and Ben Kröse. Robust scene reconstruction from an omnidirectional vision system. *IEEE Transactions on Robotics and Automation*, 19(2):351–357, 2003.
- [33] Roland Bunschoten and Ben Kröse. Visual odometry from an omnidirectional vision system. In *Proceedings of the 2003 IEEE International Conference on Robotics and Automation*, pages 577–583, Taipei, Taiwan, September 2003.
- [34] Jason Campbell, Rahal Sukthankar, and Illah Nourbakhsh. Techniques for evaluating optical flow for visual odometry in extreme terrain. In *Proceedings of 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3704–3711, Sendai, Japan, September 2004.
- [35] Jason Campbell, Rahul Sukthankar, Illah Nourbakhsh, and Aroon Pahwa. A robust visual odometry and precipice detection system using consumer-grade monocular vision. In *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, pages 3421–3427, Barcelona, Spain, April 2005.
- [36] F.J Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(6):679–698, 1986.
- [37] J. S. Chahl and M. J. Srinivassan. Reflective surfaces for panoramic imaging. *Applied Optics*, 36(31):8275–8285, November 1997.
- [38] Chang Cheng, David Page, and Mongi Abidi. Object-based place recognition and loop closing with jigsaw puzzle image segmentation algorithm. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 557–562, Pasadena, USA, May 2008.
- [39] Chroust and Vincze. Fusion of vision and inertial data for motion and structure estimation. *Journal of Robotic Systems*, 21(2):73–83, 2004.
- [40] Moo K. Chung. Model building in two-sphere via gauss-weierstrass kernel smoothing and its application to cortical analysis, part i. Technical Report 1115, University of Wisconsin, Madison, Wisconsin, November 2005.
- [41] Moo K. Chung, Richard Hartley, Kim M. Dalton, and Richard J. Davison. Encoding cortical surface by spherical harmonics. *Statistica Sinica*, 18:1269–1291, 2008.
- [42] Javier Civera, Andrew Davison, and J Montiel. Inverse depth to depth conversion for monocular SLAM. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 2778–2783, Roma, Italy, April 2007.



- [43] Javier Civera, Andrew Davison, and J. Montiel. Interacting multiple model monocular SLAM. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 3704–3709, Pasadena, USA, May 2008.
- [44] T.A. Clarke and J.G. Fryer. The development of camera calibration methods and models. *Photogrammetric Record*, 16(91):51–66, April 1998.
- [45] David Claus and Andrew W. Fitzgibbon. A rational function lens distortion model for general cameras. In *Proceedings IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2005.
- [46] Peter Corke. An inertial and visual sensing system for a small autonomous helicopter. *Journal of Robotic Systems*, 21(2):43–51, 2004.
- [47] Peter Corke, Jorge Lobo, and Jorge Dias. An introduction to inertial and visual sensing. *The International Journal of Robotics Research*, 26(6):519–535, June 2007.
- [48] Peter Corke, Dennis Strelow, and Sanjiv Singh. Omnidirectional visual odometry for a planetary rover. In *Proceedings 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4007–4012, Sendai, Japan, September 2004.
- [49] Gabriella Csurka, Christopher R. Dance, Lixin Fan, Jutta Willamowski, and Cédric Bray. Visual categorization with bags of keypoints. In *In Workshop on Statistical Learning in Computer Vision, ECCV*, 2004.
- [50] Mark Cummins and Paul Newman. Probabilistic appearance based navigation and loop closing. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 2042–2048, Roma, Italy, April 2007.
- [51] Mark Cummins and Paul Newman. Accelerated appearance-only SLAM. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 1828–1833, Pasadena, USA, May 2008.
- [52] Mark Cummins and Paul Newman. FAB-MAP: Probabilistic localization and mapping in the space of appearance. *International Journal of Robotics Research*, 27:647–665, June 2008.
- [53] M. Dailey and M. Parnickun. Simultaneous localization and mapping with stereo vision. In *International Conference on Control, Automation, Robotics and Vision*, pages 1–6, Singapore, December 2006.

- [54] Konstantinos Daniilidis and Hans-Hellmut Nagel. The coupling of rotation and translation in motion estimation of planar surfaces. In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition*, pages 188–193, June 1993.
- [55] Kostas Daniilidis and Christopher Geyer. Omnidirectional vision: Theory and applications. In *Proceedings International Conference on Pattern Recognition*, 2000.
- [56] Kostas Daniilidis, Ameesh Makadia, and Thomas Bülow. Image processing in catadioptric planes: spatiotemporal derivatives and optical flow computation. In *Proceedings of the Third Workshop on Omnidirectional Vision*, 2002.
- [57] A.J. Davison, Y. González Cid, and N. Kita. Real-time 3D SLAM with wide-angle vision. In *Proceedings IFAC Symposium on Intelligent Autonomous Vehicles*, Lisbon, July 2004.
- [58] Andrew Davison, Yolanda González Cid, and Nobuyuki Kita. Real-time 3D SLAM with wide-angle vision. In *IFAC/EURON Symposium on Intelligent Autonomous Vehicles*, Lisboa, Portugal, July 2004.
- [59] Andrew Davison, Ian Reid, Nicholas Molton, and Olivier Stasse. MonoSLAM: real-time single camera SLAM. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 29(6):1–16, June 2007.
- [60] Rachid Deriche and Gerard Giraudon. A computational approach to corner and vertex detection. *International Journal of Computer Vision*, 10(2):101–124, April 1993.
- [61] Frédéric Devernay and Olivier Faugeras. Straight lines have to be straight. *Machine Vision and Applications*, 13:14–24, 2001.
- [62] L. Dreschler and H.H Nagel. Volumetric model and 3D trajectory of a moving car derived from monocular TV frame sequences of a street scene. *Computer Graphics and Image Processing*, 20:199–228, 1982.
- [63] James R. Driscoll and Dennis M. Healy. Computing fourier transforms and convolutions on the 2-sphere. *Advances in Applied Mathematics*, 15:202–250, 1994.
- [64] Yves Dufournaud, Cordelia Schmid, and Radu Horaud. Matching images with different resolutions. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 612–618, June 2000.

- [65] Matthew Dunbabin, Peter Corke, and Gregg Buskey. Low-cost vision-based auv guidance system for reef navigation. In *Proceedings IEEE International Conference on Robotics and Autonomous Systems*, pages 7–12, New Orleans, LA, April 2004.
- [66] Eric.L.Schwartz. Computational anatomy and functional architecture of striate cortex: a spacial mapping appraoch to perceptual coding. *Vision Research*, 20:645–669, 1980.
- [67] Ryan Eustice, Oscar Pizarro, and Hanumant Singh. Visually augmented navigation in an unstructured environment using a delayed state history. In *Proceedings of the 2004 IEEE International Conference on Robotics and Automation*, pages 25–32, New Orleans, LA, April 2004.
- [68] Cornelia Fermüller and Yiannis Aloimonos. Ambiguity in structure from motion: Sphere versus plane. *International Journal of Computer Vision*, 28(2):137–154, 1998.
- [69] David Filliat. A visual bag of words method for interactive qualitative localization and mapping. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 3921–3926, Roma, Italy, April 2007.
- [70] Martin A Fischler and Robert C Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.
- [71] Andrew Fitzgibbon. Simultaneous linear estimation of multiple view geometry and lens distortion. In *Proceedings IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 225–132, 2001.
- [72] Margaret. M. Fleck. Perspective projection: the wrong imaging model. Technical Report Technical Report TR 95-01, Computer Science Department, University of Iowa, 1995.
- [73] L.M.J. Florack, B.M. ter Haar Romeney, J.J. Koenderink, and M.A. Viergever. Linear scale-space. *Journal of Mathematical Imaging and Vision*, 4:325–351, 1994.
- [74] Luc M J. Florack, Bart M ter Haar. Romeny, and J. Koenderink, Jan. Scale and the differential structure of images. *Image and Vision Computing*, 1992.

- [75] Per-Erik Forssén. Maximally stable colour regions for recognition and matching. In *International Conference on Computer Vision and Pattern Recognition*, 2006.
- [76] W. Förstner and E. Gülch. A fast operator for detection and precise location of distinct points, corners and centres of circular features. In *ISPRS Intercommission Workshop*, Interlaken, June 1987.
- [77] Wolfgang Förstner. A framework for low level feature extraction. In *ECCV '94: Proceedings of the Third European Conference-Volume II on Computer Vision*, pages 383–394, London,UK, 1994. Springer-Verlag.
- [78] Friederich Fraundorfer and Horst Bischof. Evaluation of local detectors on non-planar scenes. In *In Proc. 28th workshop of the Austrian Association for Pattern Recognition*, pages 125–132, 2004.
- [79] Friedrich Fraundorfer, Christopher Engels, and David Nistér. Topological mapping, localization and navigation using image collections. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3872–3877, San Diego, USA, October 2007.
- [80] Willi Freeden and Michael Schreiner. Non-orthogonal expansions on the sphere. *Mathematical Methods in the Applied Sciences*, 18:83–120, 1995.
- [81] José Gaspar, Cláudia Deccó, Jun Okamoto Jr, and José Santos-Victor. Constant resolution omnidirectional cameras. In *International Conference on Computer Vision*, 2007.
- [82] Donald Gennery. Generalised camera calibration including fish-eye lenses. *International Journal of Computer Vision*, 68(3):239–266, 2006.
- [83] Christopher Geyer and Kostas Daniilidis. Catadioptric camera calibration. In *Proceedings International Conference on Computer Vision*, pages 398–404, Kerkyra, Greece, September 1999.
- [84] Christopher Geyer and Kostas Daniilidis. Equivalence of catadioptric projections and mapping of the sphere. In *Workshop on Omnidirectional Vision*, 2000.
- [85] Christopher Geyer and Kostas Daniilidis. A unifying theory for central panoramic systems and practical implications. In *Proceedings European Conference on Computer Vision*, Dublin, Ireland, June 2000.

- [86] Christopher Geyer and Kostas Daniilidis. Catadioptric projective geometry. *International Journal of Computer Vision*, 45(3):223–243, 2001.
- [87] Christopher Geyer and Kostas Daniilidis. Paracatadioptric camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(5):687–695, May 2002.
- [88] S. Gilles. *Robust Description and Matching of Images*. PhD thesis, University of Oxford, 1998.
- [89] Joshua Gluckman and Shree Nayar. Ego-motion and omnidirectional cameras. In *Proceedings International Conference on Computer Vision*, pages 999–1005, 1998.
- [90] Toon Goedeme, Tinne Tuytelaars, and Luc Van Gool. Fast wide baseline matching for visual navigation. In *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'04)*, 2004.
- [91] Stevica Graovac. Principles of fusion of inertial navigation and dynamic vision. *Journal of Robotic Systems*, 21(1):13–22, 2004.
- [92] H. Groemer. *Geometric Applications of Fourier Series and Spherical Harmonics*. Cambridge University Press, 1996.
- [93] Michael D. Grossberg and Shree K. Nayar. A general imaging model and a method for finding its parameters. In *Proceedings IEEE International Conference on Computer Vision*, pages 108–115, Vancouver, BC, 2001.
- [94] C.G. Harris and M.J. Stephens. A combined corner and edge detector. In *Proceedings Fourth Alvey Vision Conference*, pages 147–151, 1988.
- [95] Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2 edition, 2003.
- [96] Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2 edition, 2003.
- [97] Richard.I. Hartley. Self-calibration from multiple views with a rotating camera. In *Proceedings of the third European Conference on Computer Vision*, pages 471–478, Stockholm, Sweden, 1994.
- [98] Richard.I. Hartley and Sing Bing Kang. Parameter free radial distortion correction with centre of distortion estimation. In *International Conference on Computer Vision*, 2005.

- [99] Kin Leong Ho and Paul Newman. Detecting loop closure with scene sequences. *International Journal of Computer Vision*, 74(3):261–286, September 2007.
- [100] P.K. Ho and R. Chung. Stereo-motion that complements stereo and motion analyses. In *Proceedings IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 213–218, San Juan, June 1997.
- [101] Tzung-Hsien Ho, Christopher C. Davis, and Stuart D. Millner. Using geometric constraints for fisheye camera calibration. In *OMNIVIS 2005*, 2005.
- [102] Stefan Hrabar and Gaurav Sukhatme. A comparison of two camera configurations for optic flow based navigation of a uav through urban canyons. In *Proceedings of 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2673–2680, Sendai, Japan, September 2004.
- [103] Andreas Huster and Stephen Rock. Relative position estimation for intervention-capable auvs by fusing vision and inertial measurements. In *International Symposium on Unmanned Untethered Submersible Technology*, 2001.
- [104] Andreas Huster and Stephen Rock. Relative position sensing by fusing monocular vision and inertial rate sensors. In *Proceedings Interantioanl Conference on Advanced Robotics (ICAR)*, pages 1562–1567, Colmbra, Portugal, June 2003.
- [105] Seth Hutchinson, Gregory.D.Hager, and Peter.I.Corke. A tutorial on visual servo control. *IEEE Transactions on Robotics and Automation*, 12(5):651–670, October 1996.
- [106] T. Iijima. Basic theory of pattern observation. *Papers of Technical Group on Automata and Automatic Control*, 1959.
- [107] Viorela Ila, Juan Andrade-Cetto, Rafael Valencia, and Alberto Sanfeliu. Vision-based loop closing for delayed state robot mapping. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3892–3897, San Diego, USA, October 2007.
- [108] John David Jackson. *Classical Electrodynamics*. John Wiley & Sons, 2nd edition, 1975.
- [109] Andrew Johnson, Steven Goldberg, Yang Cheng, and Larry Matthies. Robust and efficient stereo feature tracking for visual odometry. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 39–46, Pasadena, USA, May 2008.



- [110] Il-Kyun Jung and Simon Lacroix. High resolution terrain mapping with an autonomous blimp. In *Proceedings of the 2002 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Lausanne, Switzerland, October 2002.
- [111] Il-Kyun Jung and Simon Lacroix. High resolution terrain mapping using low altitude aerial stereo imagery. In *Proceedings of the Ninth IEEE International Conference on Computer Vision (ICCV'03)*, 2003.
- [112] Timor Kadir. *Scale, Saliency and Scene Description*. PhD thesis, University of Oxford, Department of Engineering Science, Robotics Research Group, 2002.
- [113] Timor Kadir and Michael Brady. Scale, saliency and image description. *International Journal of Computer Vision*, 45(2):83–105, 2001.
- [114] Timor Kadir and Michael Brady. Scale saliency: A novel approach to salient feature and scale selection. In *Proceedings of International Conference on Visual Information Engineering*, pages 25–28, Guildford, UK, July 2003.
- [115] Timor Kadir, Andrew Zisserman, and Michael Brady. An affine invariant salient region detector. In *Proceedings of 8th European Conference on Computer Vision*, pages 228–241, Pague, Czech Republic, May 2004.
- [116] Takeo Kanade, Omead Amidi, and Qifa Ke. Real-time and 3d vision for autonomous small and micro air vehicles. In *Proceedings IEEE Conference on Decision and Control*, pages 1655–1662, Bahamas, December 2004.
- [117] Sing Bing Kang. Catadioptric self-calibration. In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition*, pages 201–207, 2000.
- [118] Juho Kannala and Sami Brandt. A generic camera calibration method for fish-eye lenses. In *Proceedings International Conference on Pattern Recognition*, pages 10–13, August 2004.
- [119] Juho Kannala and Sami Brandt. A generic camera model and calibration method for conventional, wide-angle and fish-eye lenses. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(8):1335–1340, August 2006.
- [120] Yan Ke and Rahul Sukthankar. PCA-SIFT: A more distinctive representation for local image descriptors. In *Proceedings IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 506–513, 2004.



- [121] Jungho Kim and In-So Kweon. Robust feature matching for loop closing and localization. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3905–3910, San Diego, USA, October 2007.
- [122] Les Kitchen and Azriel Rosenfeld. Gray-level corner detection. *Pattern Recognition Letters*, 1(2):95–102, December 1982.
- [123] Jan J. Koenderink. The structure of images. *Biological Cybernetics*, 50:363–370, 1984.
- [124] Kurt Konolige, Motilal Agrawal, and Joan Solà. Large-scale visual odometry for rough terrain. In *International Symposium of Robotics Research (ISRR)*, Hiroshima, Japan, 2007.
- [125] Ben Kröse, Roland Bunschoten, Stephan Ten Hagen, Bas Terwijn, and Nikos Vlassis. Household robots look and learn. *IEEE Robotics and Automation Magazine*, pages 45–52, December 2004.
- [126] Svetlana Lazebnik, Cordelia Schmid, and Jean Ponce. A sparse texture representation using local affine regions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(8):1265–1278, 2005.
- [127] Anat Levin and Richard Szeliski. Visual odometry and map correlation. In *Proceedings IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 611–618, June 2004.
- [128] Hongdong Li and Richard Hartley. A non-iterative method for correcting lens distortion from nine point correspondences. In *Omnivision Workshop*, 2005.
- [129] Hongdong Li and Richard Hartley. Five-point motion estimation made easy. In *Proceedings of the 18th International Conference on Pattern Recognition*, pages 630–633, 2006.
- [130] Hongdong Li and Richard Hartley. Plane-based calibration and auto-calibration of a fish-eye camera. In *Computer Vision - ACCV 2006*, volume 3851/2006 of *Lecture Notes in Computer Science*, pages 21–30. Springer Berlin / Heidelberg, 2006.
- [131] Tony Lindeberg. Scale-space for discrete signals. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(3):234–254, March 1990.

- [132] Tony Lindeberg. Detecting salient blob-like image structures and their scale with a scale-space primal sketch: a method for focus-of-attention. *International Journal of Computer Vision*, 11(3):283–318, 1993.
- [133] Tony Lindeberg. Scale-space theory: A basic tool for analysing structures at different scales. *Journal of Applied Statistics*, 21(2):225–270, 1994. lindeberg3.
- [134] Tony Lindeberg. *Scale-Space Theory in Computer Vision*. Kluwer Academic Publishers, 1994.
- [135] Tony Lindeberg. *On the Axiomatic Foundations of Linear Scale-Space: Combining Semi-group Structure with Causality vs. Scale Invariance*, chapter Chapter 6. Gaussian Scale-Space Theory: Proc. PhD School on Scale-Space Theory. Kluwer Academic Publishers, Copenhagen, Denmark, May 1997.
- [136] Tony Lindeberg. Feature detection with automatic scale selection. *International Journal of Computer Vision*, 30(2):79–116, 1998.
- [137] Tony Lindeberg. Principles for automatic scale selection. In B. Jähne, editor, *Handbook on Computer Vision and Applications*, volume 2, pages 239–274. Academic Press, 1999.
- [138] Tony Lindeberg and Jonas Garding. Shape-adapted smoothing in estimation of 3-D shape cues from affine distortions of local 2-D brightness structure. *Image and Vision Computing*, 15(6):415–435, June 1997.
- [139] Haibin Ling and K. Okada. Diffusion distance for histogram comparison. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 246–253, 2006.
- [140] Jorge Lobo and Jorge Dias. Vision and inertial sensor cooperation using gravity as a vertical reference. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(12):1597–1608, December 2003.
- [141] Jorge Lobo and Jorge Dias. Inertial sensed ego-motion for 3d vision. *Journal of Robotic Systems*, 21(1):3–12, 2004.
- [142] David Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.
- [143] David G Lowe. Object recognition from local scale-invariant features. In *Seventh IEEE International Conference on Computer Vision*, pages 1150–1157, September 1999.

- [144] Bruce D Lucas and Takeo Kanade. An iterative image registration technique with an application to stereo vision. In *Proceedings of 7th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 674–679, Vancouver, August 1981.
- [145] Mark Maimone, Yang Cheng, and Larry Matthies. Two years of visual odometry on the mars exploration rovers. *Journal of Field Robotics*, 24(3):169–186, March 2007.
- [146] Ameesh Makadia, Christopher Geyer, and Kostas Daniilidis. Correspondence-free structure from motion. *International Journal of Computer Vision*, 75(3), December 2007.
- [147] Ameesh Makadia, Christopher Geyer, Shankar Sastry, and Kostas Daniilidis. Radon-based structure from motion without correspondences. In *Proceedings IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2005.
- [148] Ameesh Makadia, Lorenzo Sorigi, and Kostas Daniilidis. Rotation estimation from spherical images. In *Proceedings International Conference on Pattern Recognition*, 2004.
- [149] Anthony Mallet and Simon Lacroix. Position estimation in outdoor environments using pixel tracking and stereovision. In *Proceedings IEEE International Conference on Robotics and Automation (ICRA)*, pages 3519–3524, San Francisco, CA, April 2000.
- [150] T. Marks, Andrew Howard, Max Bajracharya, Garrison Cottrell, and Larry Matthies. Gamma-SLAM: Using stereo vision and variance grid maps for SLAM in unstructured environments. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 3717–3724, Pasadena, USA, May 2008.
- [151] J. Matas, O. Chum, M. Urban, and T. Pajdla. Robust wide-baseline stereo from maximally stable extremal regions. *Image and Vision Computing*, 22:761–767, 2004.
- [152] Jason D McEwen, Michael P Hobson, Daniel J Mortlock, and Anthony N Lasenby. Fast directional continuous spherical wavelet transform algorithms. *IEEE Transactions on Signal Processing*, 55(2):520–529, February 2007.

- [153] Christopher Mei and Patrick Rives. Single view point omnidirectional camera calibration from planar grids. In *IEEE International Conference on Robotics and Automation*, pages 3945–3950, Roma, Apr 2007.
- [154] C. Meyer and M. Deans. Content based retrieval of images for planetary exploration. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1377–1382, San Diego, USA, October 2007.
- [155] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. In *Proceedings of the 2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 257–263, June 2003.
- [156] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schafalitzky, T. Kadir, and L. Van Gool. A comparison of affine region detectors. *International Journal of Computer Vision*, 2006.
- [157] Krystian Mikolajczyk. *Detection of local features invariant to affines transformations*. PhD thesis, INPG, Grenoble, 2002.
- [158] Krystian Mikolajczyk and Cordelia Schmid. Indexing based on scale invariant interest points. In *Proceedings of 8th International Conference on Computer Vision*, pages 525–531, Vancouver, July 2001.
- [159] Krystian Mikolajczyk and Cordelia Schmid. An affine invariant interest point detector. In *Proceedings of the 7th European Conference on Computer Vision*, pages 128–142, Copenhagen, Denmark 2002.
- [160] Krystian Mikolajczyk and Cordelia Schmid. Scale and affine invariant interest point detectors. *International Journal of Computer Vision*, 60(1):63–86, 2004.
- [161] Krystian Mikolajczyk and Cordelia Schmid. A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(10):1615–1630, October 2005.
- [162] Branislav Mičušík. *Two-View Geometry of Omnidirectional Cameras*. PhD thesis, Czech Technical University, Centre for Machine Perception, Prague, Czech Republic, June 2004.
- [163] Branislav Mičušík and Thomas Pajdla. Non-central para-catadioptric camera model. Technical Report CTU-CMP-2003-19, CMP K13133 FEE, Czech Technical University, Prague, 2003.

- [164] Branislav Mičušík and Tomáš Pajdla. Estimation of omnidirectional camera model from epipolar geometry. In *Proceedings IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2003.
- [165] Farzin Mokhtarian and Riku Suomela. Robust image corner detection through curvature scale space. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(12):1376–1380, December 1998.
- [166] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit. FastSLAM: A factored solution to the simultaneous localization and mapping problem. In *Proceedings of the AAAI National Conference on Artificial Intelligence*, 2002.
- [167] P. Montesinos, V. Gouet, D. Deriche, and D. Pelé. Matching color uncalibrated images using differential invariants. *Image and Vision Computing*, 18:659–671, 2000.
- [168] Hans Moravec. Towards automatic visual obstacle avoidance. In *Proceedings of the 5th International Joint Conference on Artificial Intelligence*, page 584, August 1977.
- [169] Pierre Moreels and Pietro Perona. Evaluation of features detectors and descriptors based on 3D objects. In *International Conference on Computer Vision*, pages 800–807, October 2005.
- [170] E. Mouragnon, M. Lhuillier, M. Dhome, F. Dekeyser, and P. Sayd. Real time localization and 3D reconstruction. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2006.
- [171] Stephen M. Smith and J. Michael Brady. SUSAN—a new approach to low level image processing. *International Journal of Computer Vision*, 23(1):45–78, 1997.
- [172] C. Murillo, A. J. Guerrero, J., and C. Sagüés. SURF features for efficient robot localization with omnidirectional images. In *IEEE International Conference on Robotics and Automation*, pages 3901–3907, Roma, Italy, April 2007.
- [173] Hajime Nagahara, Koji Yoshida, and Masahiko Yachida. An omnidirectional vision sensor with single view and constant resolution. In *International Conference on Computer Vision*, 2007.
- [174] Shree Nayar. Catadioptric omnidirectional camera. In *Proceedings IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 482–488, San Juan, Puerto Rico, June 1997.

- [175] Shree Nayar and Simon Baker. Catadioptric image formation. In *Proceedings DARPA Image Understanding Workshop*, New Orleans, 1997.
- [176] Shree Nayar and Venkata Peri. Folded catadioptric cameras. In *Proceedings IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 217–225, 1999.
- [177] C. Nelson, R and J. Aloimonos. Finding motion parameters from spherical motion fields (or the advantages of having eyes in the back of your head). *Biological Cybernetics*, 58:261–273, 1988.
- [178] Jan Neumann, Cornelia Fermüller, and Yiannis Aloimonos. Eyes form eyes: New cameras for structure from motion. In *Proceedings Workshop on Omnidirectional Vision (OMNIVIS)*, 2002.
- [179] David Nistér. Reconstruction from uncalibrated sequences with a hierarchy of trifocal tensors. In *Proceedings European Conference on Computer Vision (ECCV 2000)*, pages 649–663, 2000.
- [180] David Nistér. An efficient solution to the five-point relative pose problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 26(6):756–770, June 2004.
- [181] David Nistér, Oleg Naroditsky, and James Bergen. Visual odometry. In *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2004.
- [182] David Nistér, Oleg Naroditsky, and James Bergen. Visual odometry for ground vehicle applications. *Journal of Field Robotics*, 23(1):3–20, January 2006.
- [183] David Nistér and Henrik Stewénus. Scalable recognition with a vocabulary tree. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2161–2168, 2006.
- [184] Navid Nourani-Vatani, Jonathan Roberts, and Mandyam Srinivasan. IMU aided 3d visual odometry for car-like vehicles. In *Australasian Conference on Robotics and Automation*, 2008.
- [185] Navid Nourani-Vatani, Jonathan Roberts, and Mandyam Srinivasan. Practical visual odometry for car-like vehicles. In *IEEE International Conference on Robotics and Automation*, 2009. To Appear.

- [186] Stephen Nuske, Jonathan Roberts, and Gordon Wyeth. Visual localisation in outdoor industrial building environments. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 544–550, Pasadena, USA, May 2008.
- [187] Kohtaro Ohba, Yoichi Sato, and Katsushi Ikeuchi. Appearance-based visual learning and object recognition with illumination invariance. *Machine Vision and Applications*, 12(4):189–196, 2000.
- [188] Clark Olson, Larry Matthies, Marcel Schoppers, and Mark Maimone. Robust stereo ego-motion for long distance navigation. In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition*, pages 453–458, June 2000.
- [189] Clark Olson, Larry Matthies, Marcel Schoppers, and Mark Maimone. Stereo ego-motion improvements for robust rover navigation. In *Proceedings of the 2001 IEEE International Conference on Robotics and Automation*, pages 1099–1104, Seoul, Korea, May 2001.
- [190] Clark Olson, Larry Matthies, Marcel Schoppers, and Mark Maimone. Rover navigation using stereo ego-motion. *Robotics and Autonomous Systems*, 43:215–229, 2003.
- [191] P.I.Corke. Machine vision toolbox. *IEEE Robotics and Automation Magazine*, 12(4):16–25, nov 2005.
- [192] Pedro Piniés, Todd Lupton, Salah Sukkarieh, and Juan Tardós. Inertial aiding of inverse depth SLAM using a monocular camera. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 2797–2802, Roma, Italy, April 2007.
- [193] Pedro Piniés and Juan Tardós. Scalable SLAM building conditionally independent local maps. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3466–3471, San Diego, USA, October 2007.
- [194] David Prasser, Michael Milford, and Gordon Wyeth. Outdoor simultaneous localisation and mapping using RatSLAM. In *Field and Service Robotics*, pages 143–154, 2006.
- [195] P.R.Beaudet. Rotationally invariant image operators. In *Proceedings of the 4th International Joint Conference on Pattern Recognition*, pages 579–583, Tokyo, 1978.



- [196] A. Pronobis and B. Caputo. Confidence-based cue integration for visual place recognition. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2394–2401, San Diego, USA, October 2007.
- [197] Yossi Rubner, Carlo Tomasi, and Leonidas Guibas. The earth mover’s distance as a metric for image retrieval. *International Journal of Computer Vision*, 40(2):99–121, 2000.
- [198] Davide Scaramuzza. *Omnidirectional Vision: From Calibration to Robot Motion Estimation*. PhD thesis, ETH Zurich, 2008.
- [199] Davide Scaramuzza, Agostino Martinelli, and Roland Siegwart. A flexible technique for accurate omnidirectional camera calibration and structure from motion. In *IEEE International Conference on Computer Vision Systems*, 2006.
- [200] Davide Scaramuzza and Roland Siegwart. Appearance-guided monocular omnidirectional visual odometry for outdoor ground vehicles. *IEEE Transactions on Robotics*, 24(5), October 2008.
- [201] Schaffalitzky and Zisserman. Viewpoint invariant texture matching and wide-baseline stereo. In *International Conference on Computer Vision*, 2001.
- [202] David Schleicher, Luis Bergasa, Rafael Barea, Elena López, Manuel Oca na, and Jesús Nuevo. Real-time wide-angle stereo visual SLAM on large environments using SIFT features correction. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3878–3883, San Diego, USA, October 2007.
- [203] Cordelia Schmid, Roger Mohr, and Christian Bauckage. Evaluation of interest point detectors. *International Journal of Computer Vision*, 37(2):151–172, 2000.
- [204] Cordelia Schmid, Roger Mohr, and Christial Bauckhage. Comparing and evaluating interest points. In *International Conference on Computer Vision*, pages 230–235, 1998.
- [205] Stephen Se, David Lowe, and Jim Little. Vision-based mobile robot localization and mapping using scale-invariant features. In *Proceedings IEEE International Conference on Robotics and Automation*, Seoul, Korea, May 2001.
- [206] Shishir Shah and J.K. Aggarwal. A simple calibration procedure for fish-eye (high distortion) lens camera. In *Proceedings IEEE International Conference on Robotics and Automation*, pages 3422–3427, San Diego, CA, May 1994.

- [207] Shishir Shah and J.K. Aggarwal. Intrinsic parameter calibration procedure for a (high-distortion) fish-eye lens camera with distortion model and accuracy estimation. *Pattern Recognition*, 29(11):1775–1788, 1996.
- [208] Shishir Shah and J.K. Aggarwal. Mobile robot navigation and scene modelling using stereo fish-eye lens. *Machine Vision and Applications*, 1997.
- [209] Jianbo Shi and Carlo Tomasi. Good features to track. In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 593–600, Seattle, June 1994.
- [210] J. Sivic and A. Zisserman. Video Google: A text retrieval approach to object matching in videos. In *International Conference on Computer Vision (ICCV)*, pages 1470–1477, October 2003.
- [211] Iryna Skrypnik and David Lowe. Scene modelling, recognition and tracking with invariant image features. In *3rd IEEE and ACM International Symposium on Mixed and Augmented Reality*, pages 110–119, Arlington, VA, November 2004.
- [212] H. Stewénius, C. Engels, and D. Nistér. Recent developments on direct relative orientation. *ISPRS Journal of Photogrammetry and Remote Sensing*, 60(4):284–294, 2006.
- [213] Birger Streckel and Reinhard Koch. Lens model selection for visual tracking. DAGM Symposium Vienna Austria, 2005.
- [214] Dennis Strelow and Sanjiv Singh. Motion estimation from image and inertial measurements. *The International Journal of Robotics Research*, 23(12):1157–1195, December 2004.
- [215] Rahul Swaminatham, Michael Grossberg, and Shree Nayar. A perspective on distortions. In *Proceedings IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 594–601, June 2003.
- [216] Rahul Swaminathan, Michael D. Grossberg, and Shree K. Nayar. Caustics of catadioptric cameras. In *International Conference on Computer Vision*, pages 2–9, 2001.
- [217] Rahul Swaminathan and Shree K. Nayar. Non-metric calibration of wide-angle lenses and polycameras. *Transactions on Pattern Analysis and Machine Intelligence*, 22(10):1172–1178, 2000.

- [218] Jean-Philippe Tardif, Yanis Pavlidis, and Kostas Daniilidis. Monocular visual odometry in urban environments using an omnidirectional camera. In *IEEE/RSJ International Conference in Intelligent Robots and Systems*, pages 2531–2538, 2008.
- [219] Ash Tews, Cédric Pradalier, and Jonathan Roberts. Autonomous hot metal carrier. In *IEEE International Conference on Robotics and Automation*, pages 1176–1182, Roma, Italy, April 2007.
- [220] Trung Ngo Thanh, Hajime Nagahara, Ryusuke Sagawa, Yasuhiro Mukaigawa, Masahiko Yachida, and Yasushi Yagi. Robust and real-time egomotion estimation using a compound omnidirectional sensor. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 492–497, Pasadena, USA, May 2008.
- [221] SriRam Thirthala and Marc Pollefeys. The radial trifocal tensor: a tool for calibrating the radial distortion of wide-angle cameras. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 321–328, 2005.
- [222] P. Tissainayagam and D. Suter. Assessing the performance of corner detectors for point feature tracking applications. *Image and Vision Computing*, 22:663–679, 2004.
- [223] Carlo Tomasi and Takeo Kanade. Detection and tracking of point features. Technical Report CMU-CS-91-132, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, April 1991.
- [224] Mashiro Tomono. Monocular SLAM using a rao-blackwellised particle filter with exhaustive pose space search. In *IEEE International Conference on Robotics and Automation (ICRA)*, Roma, Italy, April 2007.
- [225] Antonio Torralba, Kevin Murphy, William Freeman, and Mark Rubin. Context-based vision system for place and object recognition. In *International Conference on Computer Vision (ICCV)*, pages 273–280, October 2003.
- [226] Bill Triggs. Autocalibration from planar scenes. In *European Conference on Computer Vision*, pages 89–105, 1998.
- [227] Roger.J. Tsai. A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses. *IEEE Journal of Robotics and Automation*, RA-3(4):323–344, August 1987.

- [228] Tinne Tuytelaars and Luc Van Gool. Content-based image retrieval based on local affinity invariant regions. In *In Int. Conf. on Visual Information Systems*, pages 493–500, 1999.
- [229] Tinne Tuytelaars and Luc Van Gool. Wide baseline stereo matching based on local, affinity invariant regions. In *Proceedings British Machine Vision Conference (BMVC 2000)*, 2000.
- [230] Tinne Tuytelaars and Luc Van Gool. Matching widely separated views based on affine invariant regions. *International Journal of Computer Vision*, 59(1):61–85, 2004.
- [231] M. Ullah, M. A. Pronobis, B. Caputo, J. Luo, P. Jensfelt, and I. Christensen, H. Towards robust place recognition for robot localization. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 530–537, Pasadena, USA, May 2008.
- [232] Kane Usher. *Visual homing for a car-like vehicle*. PhD thesis, Queensland University of Technology, Brisbane, Australia, June 2005.
- [233] Christoffer Valgren and Achim Lilienthal. SIFT,SURF and seasons: long-term outdoor localization using local features. In *European Conference on Mobile Robots*, 2007.
- [234] Raquel Frizzera Vassallo, José Santos-Victor, and Hans Jorg Schneebeli. A general approach for egomotion estimation with omnidirectional images. In *Workshop on Omni-directional Vision OMNIVIS*, Copenhagen, Denmark, June 2002.
- [235] Paul Viola and Michael Jones. Rapid object detection using a boosted cascade of simple features. In *IEEE International Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1–511–1–518, 2001.
- [236] Juyang Wang, Paul Cohen, and Marc Herniou. Camera calibration with distortion models and accuracy evaluation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(10):965–980, Oct 1992.
- [237] Joachim Weickert, Seiji Ishikawa, and Atsushi Imiya. Linear scale-space has first been proposed in japan. *Journal of Mathematical Imaging and Vision*, 10:237–252, 1999.
- [238] Brian Williams, Mark Cummins, José Neira, Paul Newman, Ian Reid, and Juan Tardós. An image-to-map loop closing method for monocular SLAM. In *International Conference on Intelligent Robots and Systems*, 2008.

- [239] Stefan Williams and Ian Mahon. Simultaneous localisation and mapping on the great barrier reef. In *IEEE International Conference on Robotics and Automation*, pages 1771–1776, New Orleans, USA, April 2004.
- [240] G. Willson, Reg. *Modeling and Calibration of Automated Zoom Lenses*. PhD thesis, The Robotics Institute, Carnegie Mellon University, Pittsburgh, Pennsylvania USA, 1994.
- [241] G. Willson, Reg and A. Ahafer, Steven. What is the center of the image? *Journal of the Optical Society of America*, 11(11):2946–2955, November 1994.
- [242] Andrew.P. Witkin. Scale-space filtering. In *Proceedings Eight International Joint Conference on Artificial Intelligence*, pages 1019–1022, Karlsruhe, August 1983.
- [243] Yalin Xiong and Ken Turkowski. Creating image-based VR using a self-calibrating fisheye lens. In *Proceedings International Conference on Computer Vision and Pattern Recognition*, pages 237–243, 1997.
- [244] Xianghua Ying and Zhanyi Hu. Can we consider central catadioptric cameras and fisheye cameras within a unified imaging model. In *8th European Conference on Computer Vision ECCV*, pages 442–455, Prague, Czech Republic, May 2004.
- [245] Xianghua Ying and Zhanyi Hu. Catadioptric camera calibration using geometric invariants. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(10):1260–1271, October 2004.
- [246] Alan L. Yuille and Tomaso A. Poggio. Scaling theorems for zero crossings. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8(1):15–25, January 1986.
- [247] Y.Wiaux, L.Jacques, P.Vielva, and P.Vanderghelynst. Fast directional correlation on the sphere with steerable filters. *The Astrophysical Journal*, 652:820–832, November 2006.
- [248] Munir Zaman. High precision relative localization using a single camera. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 3908–3914, Roma, Italy, April 2007.
- [249] Zhengyou Zhang. On the epipolar geometry between two images with lens distortion. In *Proceedings International Conference on Pattern Recognition*, pages 407–411, Vienna, Aug 1996.

- [250] Zhengyou Zhang. Flexible camera calibration by viewing a plane from unknown orientations. In *International Conference on Computer Vision*, pages 666–673, 1999.
- [251] Huaibin Zhao and J.K. Aggarwal. 3D reconstruction of an urban scene from synthetic fish-eye images. In *Proceedings 4th IEEE Southwest Symposium on Image Analysis and Interpretation*, 2000.