UNIVERSITÀ DEGLI STUDI DI PALERMO

DIPARTIMENTO DI MATEMATICA E INFORMATICA
DOTTORATO DI RICERCA IN MATEMATICA E INFORMATICA

- XXII CICLO -

Matching Image Features

Author

Fabio Bellavia

Coordinator

Prof. Camillo Trapani

Thesis Advisor
Prof. Domenico Tegolo

Settore Scientifico Disciplinare INF/01

Reviewers

Prof. Emanuele Trucco
School of Computing, University of Dundee, Scotland, UK

Prof. Vittorio Murino
Dipartimento di Informatica, Università di Verona, Italy
Istituto Italiano di Tecnologia (IIT), Genova, Italy

Matching Image Features

Abstract: Matching points across different images is a fundamental task in most computer vision applications, since it allows in general to retrieve the position of image points. Three-dimensional object reconstruction, mosaicing, object and action detection and classification are some of the most popular computer vision applications that rely upon it. Several feature detectors and descriptors, as well as matching algorithms built upon them, have been presented in the last decades. Though a lot of progress has been done in this field, the problem of matching points across different images is far to be fully solved. The performances of the algorithms are closely related to the complexity and the type of the scenes, as well as the transformations between the images.

In this thesis contributes to the field of the image feature matching are presented. A new feature detector, named HarrisZ, has been developed. It improves the Harris corner detector by providing stable and robust features around the images in terms of the repeatability index and the matching score. The results are comparable with the state of the art affine detectors, such as the Hessian-affine detector and the MSER detector.

The sGLOH descriptor, an extension of the GLOH descriptor, has also been proposed. The new feature descriptor can check the similarity between two features not only in the gradient dominant orientation but also according to a set of discrete rotations, obtained by shifting the descriptor vector. This improves the descriptor stability to rotation for a reasonable computational cost.

A RANSAC based matching algorithm, called soft sparse matching, has been designed. As its main features, the proposed matching algorithm uses an image-guided selection of the error threshold, a soft matching strategy in contrast to the one-to-one matching required by RANSAC, and a global-to-local selection of the candidate matches inspired by the simulated annealing process. Moreover, the final matches are forced to be homogeneously distributed along the image, resulting in a more stable estimation of the correspondences.

Lastly, a validation framework to test feature detectors, descriptors and matching algorithms has also been proposed. It uses only geometric information and does not require complex methods to obtain the ground truth data, which makes it very attractive.

Keywords: feature detector, feature descriptors, image feature matching, Harris corner, SIFT, RANSAC, epipolar geometry

Acknowledgments

I would like to thank my advisor Professor Domenico Tegolo both for his insight and motivation, the opportunities which he provided to me during my PhD studies, his faith and patience.

Thanks to my fellow PhD friends, in particular Marco Cipolla, Filippo Millonzi, Luca Pinello and Filippo Utro with whom I have shared the highs and lows of the academic and of the real life. Many thanks also to Giosué Lo Bosco and Cesare Valenti for their friendship, their suggestions and for sharing their knowledge and skills. I am also indebted with Professor Vito Di Gesú who led the research group for passing his enthusiasm, his insight and knowledge about the computer science.

I would like to thank Professor Emanuele Trucco who provided me with valuable inputs and ideas, for his collaboration and the hospitality at the School of Computing at the University of Dundee in Scotland, where I spent several months during my PhD. Also thanks to the people I met in Dundee which made the experience to be away from home more pleasant due to their friendship.

Thanks to all the people I met in these three years, inside and outside the walls of the department, which made this experience more enjoyable. Lastly, and most importantly I would like to thank my family and my friends who have always been there for me providing understanding, patience, love, encouragement and support.

Thank you Giusy for the good and the bad times.

For the others who read this “800A !!!”

Originality Declaration

This work contains no material which has been accepted for the award of any other degree or diploma in any university or other tertiary institution and, to the best of my knowledge and belief, contains no material previously published or written by another person, except where due reference has been made in the test. I give consent to this copy of my thesis, when deposited in the University Library, begin available for loan and photocopying.

Signed . . . . January 2011

Contents

List of Figures

List of Tables

Introduction

The last goal of computer vision is to obtain a full understanding of the environment by the acquisition of one or more images of the surrounding area. To accomplish this task, high level data information about the objects in the scene and their interactions has to be extracted from low level data. In particular, the position of the points in the scene and their intensity similarities are required to obtain almost all the object relationships.

Matching points across different images is a fundamental task in most computer vision applications, since it allows to retrieve the position and the intensity of image points in general. All real objects must obey to geometric and radiometric constrains imposed by the physic laws, thus every time a new correspondence is found for a point, the derived constraint can be used to improve the information about its position in the real world. Three-dimensional object reconstruction, mosaicing, object and action detection and classification are some of the most popular computer vision applications based on point matching.

The disparity map between images, i.e. the function which maps points from one image to another, can be used to characterize the matching algorithms. Dense disparity maps are computed for all the image points (except for occlusions) and allow a detailed and fine representation, however they are difficult to obtain when the images are not very close (for instance when they are taken from very different points of view). Sparse disparity maps are computed only for a relatively small subset of salient image points, called image features. A rough image representation is obtained, which is more robust when images are not very close and it is less computational expensive, since not all point correspondences have to be computed. Moreover a sparse map can be used as a starting point to build a dense disparity map.

Several feature detectors and descriptors, as well as matching algorithms, have been presented in the last decades. Though a lot of progress has been done in this field, the problem of matching point around different images is far to be fully solved. The quality of the results is closely related to the complexity of the type of the scene, which implies to know some prior information about the images to be processed.

Thesis contributions and outline

The following contributions in the field of the image feature matching are mainly presented in this thesis:

- The HarrisZ feature detector, an improved affine detector based on the Harris corner detector. It provides stable and robust features in terms of the repeatability index and the matching score, without requiring a fine tuning of the algorithm parameters. According to the standard Oxford dataset for planar scenes and its extension to three-dimensional object, the results are comparable with those obtained by the state of the art affine detectors, such as the Hessian-affine detector and the MSER detector.

- The sGLOH descriptor, an extension of the GLOH descriptor. It provides stable feature descriptors by checking the similarity between two features not only in the predefined dominant orientation, but also according to a set of discrete rotations. This can be accomplished with a reasonable computational cost by shifting the descriptor vector and by using an improved feature distance. The proposed descriptor, has been compared with the SIFT and the GLOH descriptors on the Oxford image dataset and good results, which point out its robustness and stability, have been obtained.

- The sparse soft matching algorithm, based on RANSAC. Its main features are an image-guided selection of the error threshold, a soft matching strategy in contrast to the one-to-one matching required by RANSAC, which increases the number of the absolute matches. It also does a less random choice of candidate matches, guided by a global-to-local constrain generation inspired by the simulated annealing process. Final matches are forced to be homogeneously distributed on the images, thus a more stable estimation of the homography or of the fundamental matrix associated is achieved. As a weak point, it is more computationally expensive than RANSAC.

- A validation framework to test feature detectors, descriptors and matching algorithms. It uses only geometric information and does not require complex methods to obtain the ground truth data, which makes it very attractive. The soft sparse matching algorithm and RANSAC have been compared according to this new proposed framework.

According to the contributions, this thesis is divided in two main sections, where in the former section feature detectors and descriptors are discussed, while stereo geometry and matching algorithm are described in the latter.

In Chapters 1–5 an introduction to the state of the art feature detectors and descriptors developed in the last decades is given, followed by a comparison of the different methodologies and results. The section ends with Chapter 6, where the proposed HarrisZ detector is described, evaluated and compared with other detectors. Moreover, the novel sGLOH feature descriptor is also presented and a comparison with other feature descriptors is done.

In Chapter 7 the matching problem is presented, together with a short description of its application to the stereo three-dimensional reconstruction. The RANSAC paradigm as well as its extensions are also presented. In Chapter 8 the new soft sparse matching algorithm is proposed and evaluated according to the new validation framework. Conclusions are discussed in Chapter 9.

Chapter 1

Feature-based computer vision techniques

1.1 Historical background

Image features, *interest points* and *region of interests* are terms commonly used to define image regions which have some given properties. It is a very general definition, since every image region (here also a point, identified by a single pixel, is considered a region) can be a feature, depending of the task purpose.

Apart from specific image features which can have a semantic meaning, such as edges, blobs and junctions, nowadays any image region which is stable on image transformations and distinctive across other regions of the same image is considered a feature. Here, an image transformation refers to a wide range of situations, related to the task. In image reconstruction or mosaicing applications, transformations are related to the image acquisition process, in particular with the camera properties: perspective transformations, including scale and rotation, blur or illumination changes and instrumental noise are the most common ones. For object detection and classification, different instances of the same object around a class can be seen as transformations of the ideal object.

The first use of feature-based algorithm in computer vision can be traced back to the work of Marr and Poggio [103] and Harris and Stephens [69] on the stereo correspondence algorithms in the 1970s. The use of features instead of all available pixels provides a reasonable coverage of the object of interest with a reasonable computational cost. Furthermore, the distinctiveness of the points increases by improving the quality of the match. A further interest in image features raised in the 2000s when feature based techniques for object recognition [134, 46, 99] were developed, as features can be organized as primitives that compose the final object, thus providing a more simple data analysis at a less computational cost.

1.2 Feature definition

1.2.1 Feature properties

According to [177], good features should have the following properties:

- *Repeatability*: the same features should be present in the image after a transformation. It means that a feature should be *robust* to small image transformations and should have *invariant* properties for a wide degree of transformations. Of course it does not hold if the object to which the features are associated disappears from the scene after the image transformation.

- *Distinctiveness/informativeness*: the features in the same image should be different and a same feature should not varies across images of the same scene in order to be distinguished and matched.

- *Locality*: features should be local to reduce the probability of occlusions, for instance in object detection.

- *Quantity*: the number of features found could be varied according to the application purpose, to cover the best distribution. For instance in three-dimensional reconstruction, features should cover the image to allow the best reconstruction. This requirement should clearly be balanced with the distinctiveness criterion, since as the number of features increases the probability of wrong matches increases too.

- *Accuracy*: the localization of the features should be as accurate as possible. This property is relevant for example in camera calibration, while it can be almost neglected in object classification.

- *Efficiency*: the computational cost in time and space should be reasonable according to the application. It is a critical requirement for real-time applications, such as object tracking, or dealing with a large amount of data, such as high resolution three-dimensional reconstructions.

According to the feature requirements, from an operative point of view, two types of algorithms have been developed: *feature detector* algorithms and *feature descriptor* algorithms. A feature detector extracts the features from images. The extraction process provides the position of the feature, together with its *support region* and some other possible data which further characterize the feature, such as its scale and orientation. Example of feature detectors are the Harris corner detector [69] and the SIFT [100] detector.

Feature descriptors take the extracted feature and compute some meaningful vector which contains the information about it. The most simple feature descriptor is provided by the grid of the pixel intensities of the support region, while more complex descriptors are given by the gradient orientation histogram of the feature support region, as the popular SIFT descriptor [100], or by a combination of the responses of the support region to some image filters, such as the steerable filters [62].

As described above, a feature should be invariant to some transformations defined by the application task, this is achieved by the feature descriptor. Instead to be invariant, features are often *covariant* to transformations [114], i.e. the degree of transformations should vary gradually with the measure associated to the feature descriptor vector. Pre-normalizing the feature support region, also known as the *descriptor patch*, the descriptor vector becomes invariant to the given transformation. For example, steerable filters [62] are invariant to rotations, while in the case of the

SIFT descriptor and other similar descriptors [100, 164, 113, 11, 85, 86] the feature patch should be rotated in the direction of the dominant gradient orientation before computing the descriptor vector to become rotationally invariant. Affine-covariant feature detectors such as the Harris-affine and the Hessian-affine detectors [112] have gained attention since projective transformations can be approximated by piecewise local affine transformations.

1.2.2 Feature matching

The choice of the distance/similarity measure used to compare features is trictly related to the feature descriptor and to the matching process. Minkowski distances, such as the Euclidean distance, are a common choice as well as cross-correlation, but more complex distances have been developed. For example, slight feature variations can influence distant bins in histogram-based feature descriptors; the pyramid match kernel [67], the diffusion distance [97] and the SIFT-rank descriptor [163] can be used to alleviate this issue.

Lastly, the effective matching is done. The similarity threshold, the nearest neighbour or the nearest neighbour ratio approaches [100] are commonly used. The matching can be further refined when some constraining hypotheses exist. Registration tasks, mosaicing and three-dimensional reconstruction by RANSAC [51] or similar robust regression paradigms [169, 190, 170] can adjust matches while estimating the best camera parameters [70]. Homographies [70] are used to constrain matches for planar objects, while the fundamental matrix [44] and the trifocal tensor [70] can be used respectively for stereo vision and three view vision.

Another topic is the organization of the extracted features in pictorial dictionaries. This kind of approach is used together with learning algorithms for object recognition and classification, but also for data compression. The organization of the features in structures to allow a fast extraction and comparison is relevant for this kind of applications. The kd-tree [31] or the ANN tree [90] are examples of these structures. In the bag-of-words approach [87, 36] object can be extracted and classified by comparing their features according to a pictorial dictionary, created by a learning or a clustering algorithm.

1.2.3 Feature evaluation

Though it is often taken apart, the evaluation of feature detector and descriptor algorithms, as well as matching strategies, is a crucial and difficult topic. The main issue is related to the existence of a solid ground truth database, which considers the most common image transformations, not easy to obtain. In the case of planar objects, the popular Oxford dataset [114] is the first challenge a new detector or a new descriptor should deal with. It has been extended to three-dimensional objects in [59], however it contains only a few image sequences for each transformation to allow a really robust comparison. Apart from the Oxford database, which uses the *repeatability index* and the *matching score* as evaluation measures [149, 114], other approaches exist [119, 54], but their are less common. From all the cited evaluations [114] what rises up is that no detector outperforms the other ones for all transformations, though some detectors such as the MSER [107] and the Hessian-affine detector [112] usually perform better. About the feature descriptors, histogram-based descriptors are in general the best choice. However, the performances on three-dimensional data are poor and the *complementarity* between different kinds of features can improve the performances of feature based algorithms [41, 59].

Features have to be invariant to image transformations in order to be matched from one image to another one. Image transformations can be divided in two classes: *geometric transformations* and *radiometric transformations*. Geometric transformations modify the shape and the position of the feature in the space thus they can be subdivided according to a well known hierarchy of geometric transformations [161, 70], while radiometric transformations influence the feature appearance, i.e. the intensity value of the pixels.

1.3 Feature invariance

1.3.1 The hierarchy of geometric transformations

Homogeneous coordinates are first introduced [70]. Given a point $\mathbf{x} = [x, y]^T \in \mathbb{R}^2$ in the plane or $\mathbf{X} = [X, Y, Z]^T \in \mathbb{R}^3$ in the space, their respective homogeneous coordinates are $\overline{\mathbf{x}} = w[x, y, 1]^T$,$w \in \mathbb{R}$ and $\overline{\mathbf{X}} = W[X, Y, Z, 1]^T$,$W \in \mathbb{R}$, so that *inhomogeneous* points are mapped respectively to the rays starting from the coordinate centre which go through the points itself (see fig. 1.1). Moreover homogeneous points of the form $\overline{\mathbf{l}} = [x, y, 0]^T$ and $\overline{\mathbf{L}} = [X, Y, Z, 0]^T$, called *points at infinity* or *ideal points*, represent respectively the pencils of lines and of planes with normals $\mathbf{n} = [x, y]^T$, $\mathbf{N} = [X, Y, Z]^T$ [70]. To be noted that the homogeneous point $\overline{\mathbf{O}}$, a zero vector, does not represent any inhomogeneous point [70]. The relations $\simeq$ between homogeneous points implies that they represent the same inhomogeneous point, i.e. their respective vectors are equals up to a scale factor and this relation can be also extended to matrices

$$\begin{array}{lcl} \overline{\mathbf{p}} \simeq \overline{\mathbf{q}} & \Leftrightarrow & \overline{\mathbf{p}} = w\overline{\mathbf{q}} \\ \overline{\mathbf{p}} \simeq \overline{\mathbf{q}} & \Leftrightarrow & \overline{\mathbf{p}} = W\overline{\mathbf{Q}} \\ \mathrm{M} \simeq \mathrm{N} & \Leftrightarrow & \mathrm{M} = m\mathrm{N} \end{array} \tag{1.1}$$

Given a two-dimensional point $\mathbf{x} = [x, y]^T \in \mathbb{R}^2$, the *translation* is the most simple geometric transformation

$$\mathbf{x}' = \mathbf{x} + \mathbf{t} \tag{1.2}$$

where $\mathbf{t} = [t_x, t_y]^T \in \mathbb{R}^2$ is the translation vector, which gives two degrees of freedom. Next, the *rotation*

$$\mathbf{x}' = \mathrm{R}\mathbf{x} + \mathbf{t} \tag{1.3}$$

where $\mathrm{R} \in \mathbb{R}^{2\times 2}$ is an orthonormal matrix, has three degrees of freedom (two for $\mathbf{t}$ and one for R). The *similarity* adds another degree of freedom given by the scale

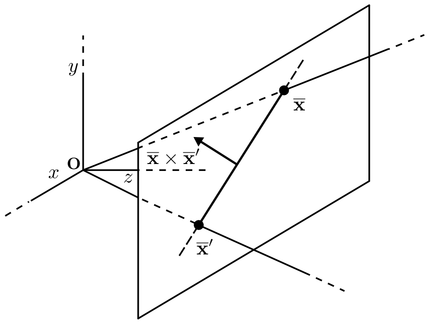


Figure 1.1: Homogeneous representation of a two-dimensional Euclidean space. The inhomogeneous points $\mathbf{x}$ and $\mathbf{x}'$ on the plane are represented respectively as the two rays from the origin $\mathbf{O}$ to their homogeneous representations $\overline{\mathbf{x}}$ and $\overline{\mathbf{x}}'$. The straight line joining $\mathbf{x}$ and $\mathbf{x}'$ is represented by the homogeneous vector $\mathbf{x} \times \mathbf{x}'$ orthogonal to the plane where both $\overline{\mathbf{x}}$ and $\overline{\mathbf{x}}'$ lie

$s \in \mathbb{R}$

$$\mathbf{x}' = s\mathrm{R}\mathbf{x} + \mathbf{t} \tag{1.4}$$

that can be put together in homogeneous coordinates

$$\overline{\mathbf{x}}' \simeq \begin{bmatrix} s\mathrm{R} & \mathbf{t} \\ \mathbf{0}^T & 1 \end{bmatrix} \overline{\mathbf{x}} \tag{1.5}$$

where $\mathbf{0}$ is a vector of zeros. A more interesting transformation is the *affine* transformation

$$\mathbf{x}' = \mathrm{A}\mathbf{x} \tag{1.6}$$

for $\mathrm{A} \in \mathbb{R}^{2\times 3}$ which has six degrees of freedom. The affine transformation maps points at infinity to points at infinity, and it can be decomposed in homogeneous coordinates as

$$\overline{\mathbf{x}}' \simeq \begin{bmatrix} \multicolumn{2}{c}{\mathrm{A}} \\ \mathbf{0}^T & 1 \end{bmatrix} \overline{\mathbf{x}} \simeq \begin{bmatrix} s\mathrm{R} & \mathbf{t} \\ \mathbf{0}^T & 1 \end{bmatrix} \begin{bmatrix} \mathrm{K} & \mathbf{0} \\ \mathbf{0}^T & 1 \end{bmatrix} \overline{\mathbf{x}} \tag{1.7}$$

where K is an upper triangular matrix which describes the shear effects [70] for which $\det(\mathrm{K}) \neq 0$. Lastly, the *perspective transformation* or *homography* has eight degrees of freedom and can be expressed in *homogeneous* coordinates as

$$\overline{\mathbf{x}}' \simeq \mathrm{H}\overline{\mathbf{x}} \simeq \begin{bmatrix} s\mathrm{R} & \mathbf{t} \\ \mathbf{0}^T & 1 \end{bmatrix} \begin{bmatrix} \mathrm{K} & \mathbf{0} \\ \mathbf{0}^T & 1 \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{v}^T & v \end{bmatrix} \overline{\mathbf{x}} \tag{1.8}$$

where $v \neq 0$ and $\mathbf{v}$ is a generic vector, so that points to infinity can be mapped to finite points and vice-versa [70].

Starting from the more complex transformations toward the simplest ones, perspective transformations preserve only straight lines, i.e. the incidence between lines, affine transformations also the parallelism between lines, while similarities, rotations and translations additively preserve also angles, distances and orientations (see fig. 1.2 for a schematic representation of the different transformations).

Though the perspective transformations are the most general geometric transformations, which cover all the other geometric transformations, it preserves only straight lines (and not their measures or orientations) due to its high degrees of freedom, so feature invariants to perspective transformations are poor in terms of feature distinctiveness and are not used.

More interesting are the affine transformations, which preserve the parallelism between lines and give a sufficient degree of distinctiveness between features. Moreover, perspective transformations can be approximated very well by piecewise local affine transformations [114, 177] (see fig. 1.3).

It is a common approach to normalize the feature patch in order to obtain invariance [113, 100, 114, 177, 112, 3, 96, 108]. Translation is trivially resolved by using the feature patch centre as the coordinate origins, while rotation, scaling and shearing require more details.

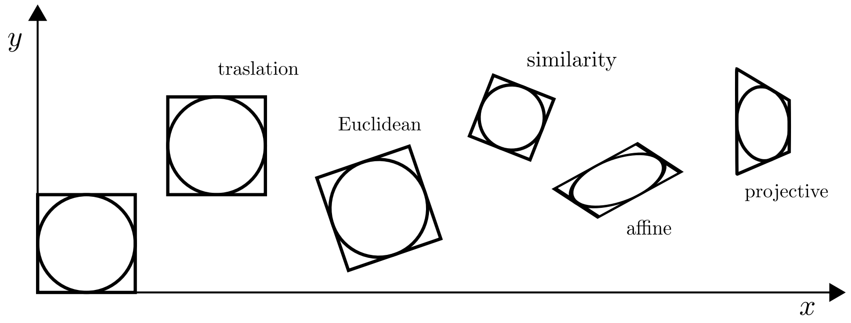


Figure 1.2: From left to right, the different geometric transformations. Every transformation includes the earlier, i.e. they are nested. Image adapted from [161]

1.3.2 Rotation invariants

Feature patches are usually normalized by a rotation according to the *gradient dominant orientation* [100]. Assuming that an image I is a continuous and differentiable two-dimensional function, the image gradient in a point $\mathbf{x}$ of the patch is

$$\nabla_{I(\mathbf{x})} = \frac{\partial}{\partial \mathbf{x}} I(\mathbf{x}) = [d_x, d_y]^T \tag{1.9}$$

An histogram of the *gradient orientation* $\phi_{I(\mathbf{x})} = \arctan(d_y/d_x)$ weighted by the squared *gradient magnitude* $\mathcal{M}_{I(\mathbf{x})} = {d_x}^2 + {d_y}^2$ is built up and the orientation of the maximal bin is selected (see fig. 1.4).

Differential invariants to rotations also exist [83, 162], the most common examples are the gradient magnitude and the *Laplacian*

$$\mathcal{L}_{I(\mathbf{x})} = \nabla^2_{I(\mathbf{x})} = \frac{\partial^2}{\partial x^2} I(\mathbf{x}) + \frac{\partial^2}{\partial y^2} I(\mathbf{x}) \tag{1.10}$$

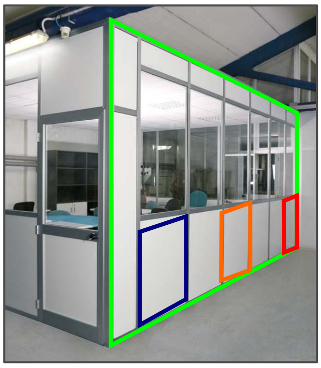


Figure 1.3: Perspective transformations can be approximated by piecewise local affine transformations. In the example a rectangular patch ongoing to a perspective transformation (the green boundary) could not be well approximated by an affine transformation, while the approximation error using local affine transformation decreases by decreasing the patch size (respectively the blue, the orange and the red boundaries)

1.3.3 The Gaussian scale-space

To handle the feature scale, the *Gaussian scale-space* theory has been introduced [94]. The main idea is to simulate the scale change factor σ by convolving the image with a Gaussian kernel with zero mean of the form

$$g_\sigma(\mathbf{x}) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \tag{1.11}$$

Different motivations have been adduced which yield to the choose of Gaussian kernel. Koendering [82] showed that the scale-space must satisfy the diffusion equation for which a Gaussian convolution is the only solution, while the different formulations proposed by Babaud [1], Lindeberg [91] and Florack [52] also conclude that the Gaussian kernel is the best choice.

The Gaussian filter benefits from the *linearity*, the *separability*, the *causality* and the *semi group* properties [94]. The separability property states that a multidimensional Gaussian kernel can be obtained as the product of one-dimensional Gaussian kernels

$$g_\sigma(\mathbf{x}) = g_\sigma(x) g_\sigma(y) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2}{2\sigma^2}} \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{y^2}{2\sigma^2}} = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \tag{1.12}$$

The causality property states that no new local maxima appear while increasing the scale, whereas the semi group property states that n successive convolutions with a scale factor σ are equal to a single convolution with a scale factor $n\ \sigma$

$$g_\sigma(\cdot) {*} g_\sigma(\cdot) = g_{2\sigma}(\cdot) \tag{1.13}$$

where $*$ means the convolution.

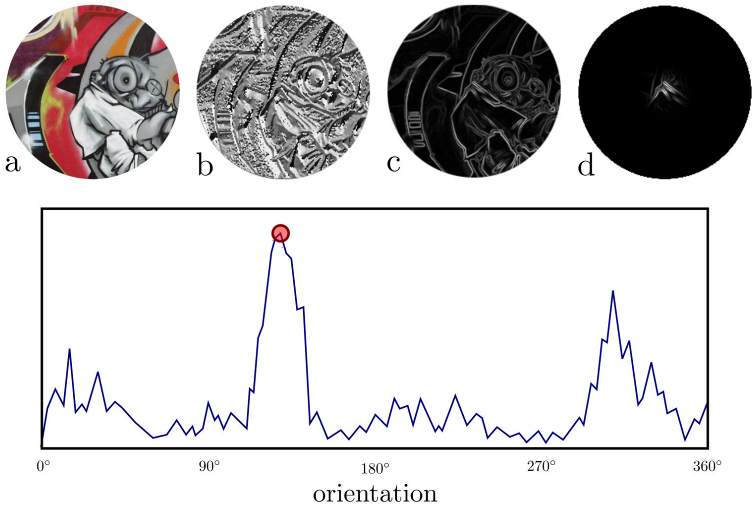


Figure 1.4: An image patch (a), its gradient orientation map (b) and the gradient magnitude map (c). The gradient orientation histogram (bottom row) is obtained using a small neighbourhood of the gradient magnitude map (d). The dominant gradient orientation is the direction for which there is the maximal peak in the gradient orientation histogram (red dot)

The scale-space derivative $L_{i_1 \dots i_m}(I(\mathbf{x}), \sigma)$ of the point $\mathbf{x}$ of image I at scale σ of order m respect to the Cartesian coordinates $i_1 \dots i_m$ can be obtained by convolving the image with the Gaussian kernel $g_\sigma(\cdot)$ and then taking the derivative, which is equivalent to convolving the image with the derivative of the Gaussian kernel. The normalization factor σ^m is introduced to take into account the decrease in the amplitude of the signal with the scales

$$L_{i_1 \dots i_m}(I(\mathbf{x}), \sigma) = \sigma^m \, g_\sigma(\cdot) * \frac{\partial^m}{\partial i_1 \dots \partial i_m} I(\mathbf{x}) = \sigma^m \, \frac{\partial^m}{\partial i_1 \dots \partial i_m} g_\sigma(\cdot) * I(\mathbf{x}) \tag{1.14}$$

In fact, if I and I' are images related to a scale change by a factor s and $\mathbf{x}' = s\mathbf{x} + \mathbf{t}$

$$I(\mathbf{x}) = I'(\mathbf{x}') \tag{1.15}$$

it follows that

$$\begin{aligned} L_{i_1 \dots i_m}(I'(\mathbf{x}'), \sigma) \quad &= \quad \sigma^m \frac{\partial^m}{\partial i_1 \dots \partial i_m} g_\sigma(\cdot) * I'(\mathbf{x}') \quad = \\ & \quad\ s^m \sigma^m \frac{\partial^m}{\partial i_1 \dots \partial i_m} g_{s\sigma}(\cdot) * I(\mathbf{x}) \quad = \quad L_{i_1 \dots i_m}(I(\mathbf{x}), s\sigma) \end{aligned} \tag{1.16}$$

that is, the scale factor is correctly taken into account as the value σ is supposed to be the scale unit [94].

In the scale-space theory the definitions of the *inner scale* and the *outer scale* are also introduced [94], the former is the minimal scale for which feature information can be detected and it is related to the image resolution, the latter is the minimal scale for which the feature is completely visible. Though the scale-space allows to simulate scale changes, the feature scale has to be chosen in order to normalize the support region of the feature.

While it possible to take a feature patch at different sampled scales [148], usually obtained by the computation of a Gaussian pyramid [20, 35] for efficiency, it is a common approach to select for each feature a *characteristic scale* for which a given function shows a particular property, for instance a local extremum over the scales [95] (see fig. 1.5). According to the results in [111], the detection of local maxima of the Laplacian over scales is a good choice.

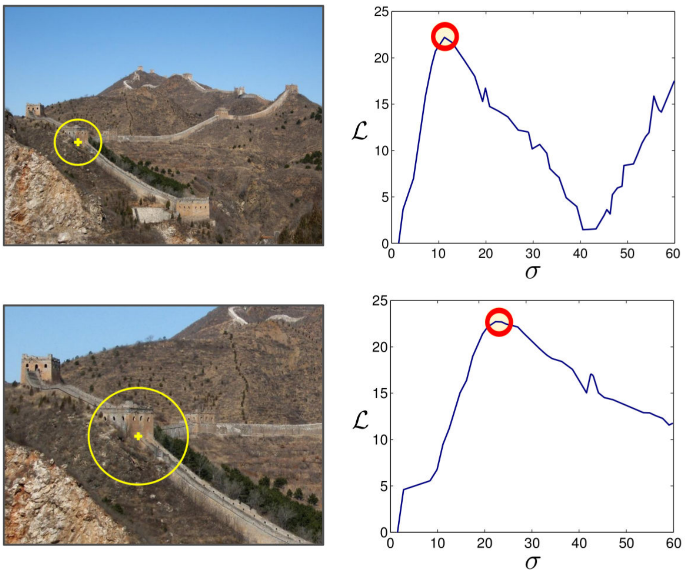


Figure 1.5: The same image at different scales (left) and the Laplacian (see eq.1.25) computed in the centre of a circular window for different scale factors σ (right). The characteristic scales detected for noise clean peaks of the Laplacian (red dots) correspond to the yellow circles in the images

1.3.4 The Affine scale-space

The *uniform scale-space* described so far, can be extended to the *affine scale-space* [94], which also considers affine transformations by introducing the *covariance matrix* Σ of the feature patch [74], also called *second moment matrix* or *autocorrelation matrix*. If $\mathrm{X} \in \mathbb{R}^{n \times m}$ is a matrix whose element X_{ij} represents the j-th feature of the i-th data (here a feature does not mean an image feature) the general expression for the covariance matrix is

$$\Sigma = (\mathrm{X} - \overline{\mathrm{X}})^T(\mathrm{X} - \overline{\mathrm{X}}) \tag{1.17}$$

where $\overline{\mathrm{X}}_{ij} = \frac{1}{n}\sum_{i=1}^{n} \mathrm{X}_{ij}$ is the mean value of the j-th feature. The Gaussian kernel becomes

$$g_\Sigma(\mathbf{x}) = \frac{1}{2\pi\mathrm{det}(\Sigma)} e^{-\frac{\mathbf{x}^T\Sigma^{-1}\mathbf{x}}{2}} \tag{1.18}$$

The covariance matrix is symmetric, thus it has positive eigenvalues and can be diagonalized. In particular in a two-dimensional space, i.e. $\mathrm{X} \in \mathbb{R}^{n\times 2}$

$$\Sigma = \mathrm{RDR}^T \tag{1.19}$$

where $\mathrm{D} = \mathrm{diag}(\lambda_1, \lambda_2)$ is a diagonal matrix, R is a orthonormal matrix and λ_1, λ_2 are the eigenvalues of Σ. As equation $\mathbf{x}^T\Sigma\mathbf{x} = 0$, an ellipse is associated to the covariance matrix, centred in the mean value of the data $\mathbf{x}_c = [\overline{\mathrm{X}}_1, \overline{\mathrm{X}}_2]^T$ with axis lengths and directions given respectively by the squared roots of the eigenvalues $\sqrt{\lambda_1}, \sqrt{\lambda_2}$ and their associated normalized eigenvectors. The affine coordinate change

$$\mathbf{x}' = \mathrm{A}\mathbf{x} \tag{1.20}$$

where $\mathrm{A} = \mathrm{D}^{-\frac{1}{2}}\mathrm{R}^T$, related to the *Mahalanobis distance*

$$\mathcal{M}(\mathbf{x}_1, \mathbf{x}_2) = \mathbf{x}_1^T\Sigma^{-1}\mathbf{x}_2 = \mathbf{x}_1^T\mathrm{A}^T\mathrm{A}\mathbf{x}_2 \tag{1.21}$$

with $\mathbf{x}_1, \mathbf{x}_2 \in \mathbb{R}^2$, allows to normalize the patch with respect to affine transformations. It geometrically corresponds to rotate the patch according to the eigenvectors of Σ and then to stretch the patch so that the eigenvalues have the same normalized length. It should be noted that the translation by a vector $\mathbf{t}$ can be neglected, assuming the feature centre as the coordinate origin. As shown in fig. 1.6 after this normalization the ellipse associated to Σ becomes a circle, so that affine covariant features become effectively affine invariant, since the stretching effects are removed.

The uniform scale-space can be seen as a particular case of the affine scale-space

$$\Sigma = \sigma^2\mathbf{I} \tag{1.22}$$

where $\mathbf{I} \in \mathbb{R}^{2\times 2}$ is the identity matrix, so that replacing the uniform Gaussian g_σ with the affine Gaussian g_Σ and using $\sqrt{\mathrm{det}(\Sigma)}$ instead of σ as scale factor in the normalized derivative $L_{i_1\ldots i_m}(I(\mathbf{x}), \sigma)$, all the formulas for the uniform scale-space still hold in the affine scale-space.

A kind of covariance matrix commonly used by feature detectors [118, 69, 56, 152, 112, 3] is the *autocorrelation matrix* of the intensity gradient

$$\mu\left(I(\mathbf{x}), \sigma_I, \sigma_D\right) = \\ = g_{\sigma_I}(\cdot) * \begin{bmatrix} L_x^2(I(\mathbf{x}), \sigma_D) & L_x(I(\mathbf{x}), \sigma_D)L_y(I(\mathbf{x}), \sigma_D) \\ L_x(I(\mathbf{x}), \sigma_D)L_y(I(\mathbf{x}), \sigma_D) & L_y^2(I(\mathbf{x}), \sigma_D) \end{bmatrix} \tag{1.23}$$

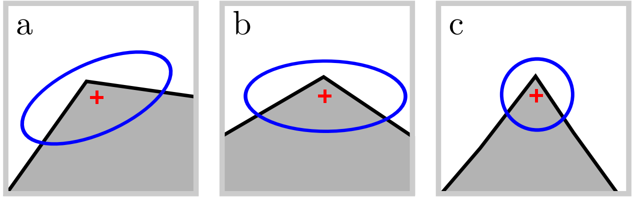


Figure 1.6: The original feature patch (a) is rotated so that the axes of the ellipse correspond to the reference coordinate axes (b) and the ellipse axes are scaled to have the same value (c)

where σ_I,σ_D are respectively the *integration scale* and the *differentiation scale*, which can be associated with the outer scale and the inner scale respectively.

The *Hessian matrix* is also employed as covariance matrix in many feature detectors [112, 5]

$$\mathcal{H}\left(I(\mathbf{x}), \sigma_D\right) = \begin{bmatrix} L_{x^2}(I(\mathbf{x}), \sigma_D) & L_{xy}(I(\mathbf{x}), \sigma_D) \\ L_{xy}(I(\mathbf{x}), \sigma_D) & L_{y^2}(I(\mathbf{x}), \sigma_D) \end{bmatrix} \tag{1.24}$$

To be noted that the Laplacian is the trace of the Hessian matrix

$$\mathcal{L}\left(I(\mathbf{x}), \sigma_D\right) = \text{trace}\left(\mathcal{H}\left(I(\mathbf{x}), \sigma_D\right)\right) \tag{1.25}$$

1.3.5 Subpixel precision

In order to better characterize the feature point, a *subpixel precision* localization can be performed. The most common approach [174] fits a parabola along both the x and y directions by using the point neighbourhood and takes the respective maxima as coordinates (see fig. 1.7(c)). If the initial point estimation is $\mathbf{x} = [x, y]^T$ then

$$\begin{aligned} \mathrm{A}\mathbf{w} &= \mathbf{b} \\ \mathrm{A}'\mathbf{w}' &= \mathbf{b}' \end{aligned} \tag{1.26}$$

represent the equations of the fitted parabolas respectively on the x and y axes, where

$$\begin{aligned} \mathrm{A} &= \begin{bmatrix} 1 & -1 & 1 \\ 0 & 0 & 1 \\ 1 & 1 & 1 \end{bmatrix}, \quad \mathbf{w} = \begin{bmatrix} a & b & c \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} I(x-1, y) \\ I(x, y) \\ I(x+1, y) \end{bmatrix} \\ \mathrm{A}' &= \begin{bmatrix} 1 & -1 & 1 \\ 0 & 0 & 1 \\ 1 & 1 & 1 \end{bmatrix}, \quad \mathbf{w}' = \begin{bmatrix} a' & b' & c' \end{bmatrix}, \quad \mathbf{b}' = \begin{bmatrix} I(x, y-1) \\ I(x, y) \\ I(x, y+1) \end{bmatrix} \end{aligned} \tag{1.27}$$

are respectively the point coordinates on the parabolas (see fig. 1.7(b)), the parabola coefficients and the parabola values for each axis. Solving by $\mathbf{w}$ and $\mathbf{w}'$

$$\begin{aligned} \mathbf{w} &= \mathrm{A}^{-1}\mathbf{b} \\ \mathbf{w}' &= \mathrm{A}'^{-1}\mathbf{b}' \end{aligned} \tag{1.28}$$

the equations of the parabolas are obtained, for which the maxima can be computed, i.e. the vertices. The final estimate of the point obtained by adding the correction factor Δ^* is

$$\mathbf{x}' = \mathbf{x} + \Delta^* = \mathbf{x} + \begin{bmatrix} -b/2a \\ -b'/2a' \end{bmatrix} \tag{1.29}$$

A more sophisticated approach fits the the *8-neighbourhood* of the point to a paraboloid [140] (see fig. 1.7(d)). A linear system can be obtained starting from the paraboloid equation as done before

$$a\,x^2 + b\,y^2 + c\,xy + d\,x + e\,y + f = I(x, y) \tag{1.30}$$

which can be solved by least-square since it is overdetermined using for instance the *pseudoinverse* matrix. After retrieving the coefficients, the maximum can be obtained by imposing the partial derivatives equal to 0, which yields to the new estimate of the feature point

$$\mathbf{x}' = \mathbf{x} + \Delta^\star = \mathbf{x} + \begin{bmatrix} \dfrac{2\,bd - ce}{c^2 - 4\,ab} \\ \dfrac{2\,ae - cd}{c^2 - 4\,ab} \end{bmatrix} \tag{1.31}$$

Proposed by Lowe and Brown [17], a last method approximates the image around the feature point $\mathbf{x}$ by the second order Taylor expansion

$$I(\mathbf{x} + \Delta) = I(\mathbf{x}) + \nabla^T_{I(\mathbf{x})}\Delta + \frac{1}{2}\Delta\mathcal{H}_{I(\mathbf{x})}\Delta \tag{1.32}$$

so that by imposing the derivatives equal to zero

$$\nabla_{I(\mathbf{x})} + \mathcal{H}_{I(\mathbf{x})}\Delta = 0 \;\Rightarrow\; \Delta = -\mathcal{H}^{-1}_{I(\mathbf{x})}\nabla_{I(\mathbf{x})} \tag{1.33}$$

the location of the new estimate maximum $\mathbf{x}'$ with respect to $\mathbf{x}$ is obtained

$$\mathbf{x}' = \mathbf{x} - \mathcal{H}^{-1}_{I(\mathbf{x})}\nabla_{I(\mathbf{x})} \tag{1.34}$$

where the derivatives in $\mathbf{x}$ can be estimated numerically.

1.3.6 Affine illumination invariance

General radiometric transformations are more difficult to handle, but in the most cases it is sufficient that features are invariant to affine illumination changes (see fig. 1.8)

$$I'(\mathbf{x}) = aI(\mathbf{x}) + b \tag{1.35}$$

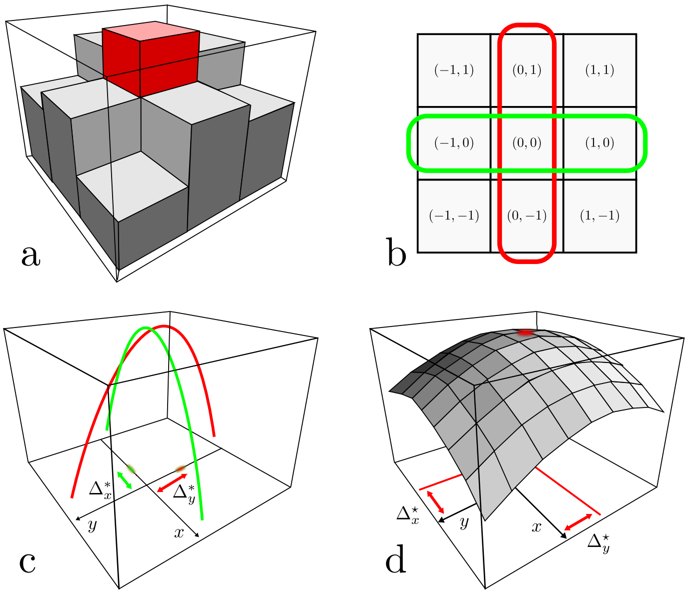


Figure 1.7: The discrete estimation of the maximum (a) can be further refined by fitting two different parabolas along each axis (c) by using the coordinate systems described in (b). A better refinement can be obtained by a paraboloid fitting (d) using a 8-neighbourhood (b). Images from [140]

where $I(\mathbf{x})$ is the pixel intensity and $a, b \in \mathbb{R}$. Invariance for image derivatives of order n to affine illumination changes can be obtained through a division by the first derivative [111]

$$\frac{\frac{\partial^n}{\partial x^n}\left(aI(\mathbf{x})+b\right)}{\frac{\partial}{\partial x}\left(aI(\mathbf{x})+b\right)} = \frac{a\frac{\partial^n}{\partial x^n}I(\mathbf{x})}{a\frac{\partial}{\partial x}I(\mathbf{x})} = \frac{\frac{\partial^n}{\partial x^n}I(\mathbf{x})}{\frac{\partial}{\partial x}I(\mathbf{x})} \tag{1.36}$$

A more general approach is to normalize the intensity value $I(\mathbf{x})$ by the mean $\overline{I}$ and standard deviation $\mathrm{std}(I)$ of the feature patch [113] because

$$\overline{I}' = a\overline{I} + b \tag{1.37}$$

$$\mathrm{std}(I') = a\,\mathrm{std}(I) \tag{1.38}$$

thus

$$\frac{I'(\mathbf{x}) - \overline{I}'}{\mathrm{std}(I')} = \frac{aI(\mathbf{x}) + b - a\overline{I} + b}{a\,\mathrm{std}(I)} = \frac{a\left(I(\mathbf{x}) - \overline{I}\right)}{a\,\mathrm{std}(I)} = \frac{I(\mathbf{x}) - \overline{I}}{\mathrm{std}(I)} \tag{1.39}$$

1.4 Invariance between object instances and classes

Though not real transformations, in this section the feature invariance to *instance transformations* and *between class transformations* will be examined. Both these

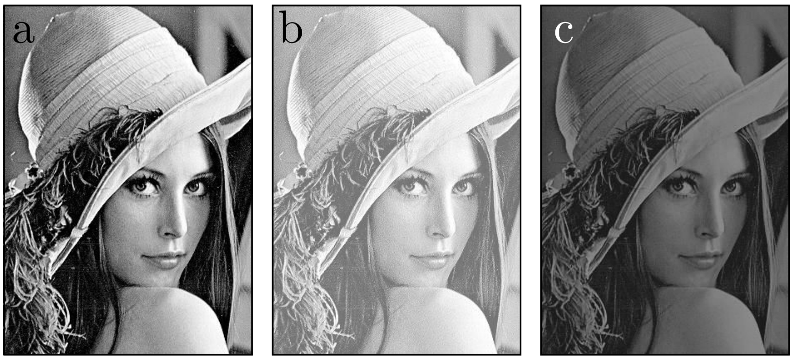


Figure 1.8: The original image (a) and some results for different affine illumination transformations (b,c)

transformations are related to detection and recognition tasks, the former is referred to a particular instance of an object, while the latter to an object class.

Objects and their classes can be considered as entities composed from some basis features, called *visual words* [153, 129, 132], which characterize their properties.

1.4.1 Object detection

First works about object detection can be traced back to Lowe [99], where after the extraction of SIFT features, the detection is performed by using a *Hough transform* approach [150], i.e. every feature increases the vote for a particular object in predefined positions, orientations and scales. In [186, 27] this approach has been extended to the *local affine frame* by the *geometric hashing* to improve the efficiency of the search. Though these approaches work well, the computation becomes prohibitive as the number of objects in the database increases. Another approach described in [49] improves the detection by increasing the number of the matched features on a candidate object while simultaneously increasing and refining the confidence of the estimation.

Using *information retrieval* techniques, Sivic and Zisserman [153], proposed the following method. They first compute the covariance matrix between normalize feature patches and then they match the features by using the Mahalanobis distance to finally cluster the features by the *k-means* [74] and obtain the final visual words. For each object in the database the *term frequency-inverse document frequency (tf-idf)* vector is computed with respect to the visual words and used whenever a query is presented. The final best candidates are verified by using geometric constrains. The computational efficiency of this method can be increased by the *hierarchical vocabulary tree* [129], where feature vectors are hierarchically clustered into a k-way tree of prototypes, or by the *randomized forest of k-d trees* [132], which allow a faster and more efficient database construction and query search.

1.4.2 Class recognition

Though object detection and classification are similar problems, the latter is more challenging, since the object ownership to a class is not only related to its appearance but also to its uses and its context. The most simple approach is the *bag-of-words* [87, 36], where a vocabulary of visual words is built up by a k-means clustering, and frequency histograms between the training images and the query images are compared. Differently form the object detection, no geometric verification is performed.

A finer approach uses the *pyramid match kernel* [67] to compare two collections of feature directly, without using visual words. At each level of the pyramid, more coarser histograms are computed (i.e. the bin size increases as the pyramid level increases). For each level the intersection between the histograms is computed as the minimum value between two corresponding bins and the final similarity measure is obtained summing up the weighted intersection between the histograms of each level, so that finer pyramid levels are more relevant (see fig. 1.9).

The *spatial pyramid matching kernel* [87] includes also geometric information. Quantized pairs of interest point location and descriptor are considered as base elements and coarser levels are obtained by merging histograms only by locations. In this way, the representation captures the distribution of both the appearance and the location of the interest points.

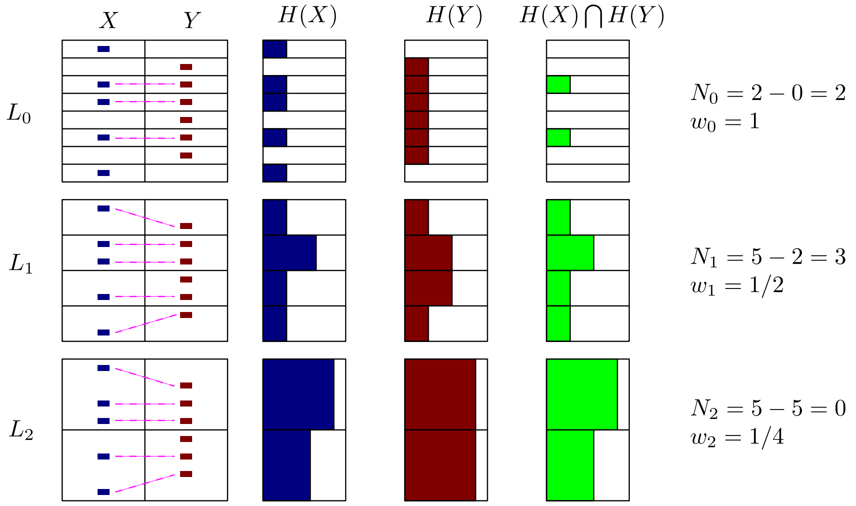


Figure 1.9: An example of the pyramid matching kernel computation for two sets X, Y. At each level L_i, $i = 0, 1, 2$ the bin size of the histograms $H(X)$ and $H(Y)$ doubles, while the weight w_i halves. The intersection between two corresponding bins is given by their minimum, and the number of new matches N_i is the difference of the matches found minus the matches of the previous levels. The final score between X and Y is $\sum_i w_i N_i$. Image from [67]

A further improvement is represented by the *proximity distribution kernel* [98], which overcomes the limit given by expressing the position by absolute coordinates. The idea is to start from triplets given by two descriptors and their relative distances to obtain coarser histograms by merging the relative distance. A generalization of these kernels is provide by the *relaxed matching kernels* introduced in [181].

Another approach to object classification is provided by the *part-based models* [19, 45, 34, 47], where object base elements are found and their geometric relationships measured. Different topologies for the geometric connections can be used, with different computational impact and performances. The most tractable are the *tree model* [45] or the *star model* [34], while the *full constellation model* [47] requires a low number of nodes, i.e. object parts, to be practical. The distribution of the relationships between visual words in an image can also be employed for classification, as done by *correlatons* [145].

To be mentioned also some models inspired by the visual cortex system, as the *HMAX* [137] and the *CNN* [88] (*Convolutional Neural Networks*), which have provided good results in classification tasks. Starting from low level layers of simple filters, such as a Gabor filter bank [66], these models combine each layer hierarchically. Small perturbations in localizations and shapes are allowed to obtain the final classification.

Chapter 2

Feature detectors

2.1 Introduction

Feature detectors are used to detect interest points. They usually not only provide information about the positions of the points but also on the shape of their support regions. Feature detectors can be classified according to the extracted regions as *corner detectors* and *blob detectors*. Obviously, edge detectors or detectors for more specific structures also exist, for instance line detectors have been applied successfully to wide baseline stereo matching [50], but they are out of the scope of this thesis. A corner detector extracts corners, defined as regions of the image with strong intensity variations along all directions. Corners usually correspond to *junctions*, even if the corner detectors extract a more general class of features, such as spots over uniform regions. A blob detector detects blob-like structures, i.e. regions with uniform intensity values. The two classes of detectors are not truly well separated: for instance the Hessian matrix was first used as a corner detector since it finds corner points, but these are usually localized at the boundaries of uniform regions, thus it should be considered as a blob detector [112]. Some authors [177] use to introduce the additional class of the *region detectors*, which are concerned with extraction of image regions in general, however such detectors can be usually classified as blob detectors.

2.2 Corner detectors

2.2.1 The autocorrelation matrix properties

The first corner detector can be attributed to Moravec [118]. As shown in fig. 2.1, checking the intensity variation along all the possible directions by a sliding window centered on the interest point, the following cases can be distinguished:

- no relevant intensity variation along all directions, i.e. a flat region;
- a strong intensity variation along one direction, i.e. an edge in the orthogonal direction of the intensity variation;
- a strong intensity variation along all directions, i.e. a corner.

The above considerations have been formalized by the mean of the autocorrelation matrix [69, 101]. If

$$\mathcal{C}(\mathbf{x}) = \sum_{\mathbf{x}_i \in \mathcal{R}} [I(\mathbf{x}_i) - I(\mathbf{x}_i + \Delta)]^2 \tag{2.1}$$

where $\mathcal{R}$ is the window centered in $\mathbf{x}$, $\Delta = \mathbf{t}$ is the shift vector and $\mathrm{x}_i \in \mathbb{R}^2$ is a pixel inside the window, the first order Taylor expansion can be used to approximate the translation

$$I(\mathbf{x}_i + \Delta) \approx I(\mathbf{x}_i) + \nabla^T_{I(\mathrm{x}_i)} \Delta \tag{2.2}$$

where

$$\nabla_{I(\mathbf{x}_i)} = \frac{\partial}{\partial \mathbf{x}_i} I(\mathbf{x}_i) = [d_{x_i}, d_{y_i}]^T \tag{2.3}$$

Thus, by substituting equation 2.2 in equation 2.1

$$\begin{aligned} \mathcal{C}(\mathbf{x}) &= \sum_{\mathbf{x}_i \in \mathcal{R}} [I(\mathbf{x}_i) - I(\mathbf{x}_i + \Delta)]^2 = \sum_{\mathbf{x}_i \in \mathcal{R}} \left[I(\mathbf{x}_i) - I(\mathbf{x}_i) - (\nabla^T_{I(\mathrm{x}_i)} \Delta) \right]^2 = \\ &= \sum_{\mathbf{x}_i \in \mathcal{R}} \Delta^T \nabla_{I(\mathrm{x}_i)} \nabla^T_{I(\mathrm{x}_i)} \Delta = \Delta^T \mu \left(I(\mathbf{x}) \right) \Delta \end{aligned} \tag{2.4}$$

where $\mu \left(I(\mathbf{x}) \right)$ is the autocorrelation matrix

$$\mu \left(I(\mathbf{x}) \right) = \begin{bmatrix} \sum\limits_{\mathbf{x}_i \in \mathcal{R}} d^2_{x_i} & \sum\limits_{\mathbf{x}_i \in \mathcal{R}} d_{x_i} d_{y_i} \\ \sum\limits_{\mathbf{x}_i \in \mathcal{R}} d_{x_i} d_{y_i} & \sum\limits_{\mathbf{x}_i \in \mathcal{R}} d^2_{y_i} \end{bmatrix} \tag{2.5}$$

The autocorrelation matrix μ is symmetric thus it has positive eigenvalues λ_1, λ_2 where $\lambda_1 \geq \lambda_2$ and the following relations can be derived:

- $\lambda_1 \approx 0$ and $\lambda_2 \approx 0$, the region is flat since there are no relevant intensity variations;
- $\lambda_1 \gg \lambda_2$ and $\lambda_2 \approx 0$, there is an edge since there is a strong intensity variation along the direction orthogonal to the eigenvector corresponding to λ_1;
- $\lambda_1 \approx \lambda_2$ and $\lambda_1, \lambda_2 \gg 0$, there is a corner where the directions of maximum intensity variation are given by the eigenvectors of μ.

Moreover, since eigenvalues are invariant to rotation, the extracted feature is also rotational invariant.

2.2.2 The Harris corner detector

Different function have been proposed to take into account the cornerness relation given by the eigenvalues of the autocorrelation matrix. Considering that for a generic matrix the product of its eigenvalues is its determinant while their sum corresponds to its trace, Harris e Stephens proposed the function [69]

$$H = \det(\mu) - \kappa \, \mathrm{trace}^2(\mu) \tag{2.6}$$

since the determinant is mostly sensible to corners (see fig. 2.2(d)), while the trace is sensible to both corners and edges (see fig. 2.2(e)). The linear coefficient κ is chosen empirically and feasible values usually range in $[0.04, 0.06]$ [112]. A *cornerness map* for all points in the image is computed and local maxima greater than a threshold value $H > th_H$ are selected, as described in fig. 2.2(f).

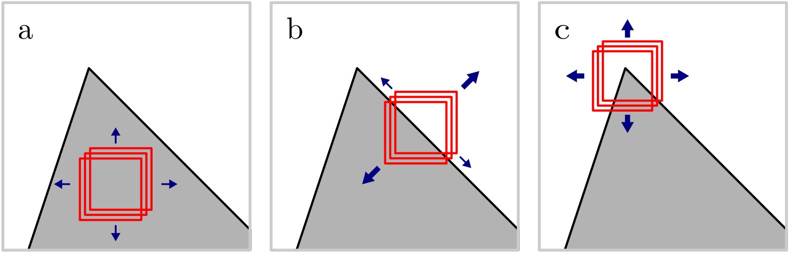


Figure 2.1: A flat region is present when the intensity variation of a sliding window along all directions is negligible (a). For edges the intensity variation is relevant only along the direction orthogonal to the edge (b). When the intensity variation is strong along all direction there is a corner (c)

2.2.3 The Förstner detector

The measure F similar to H was proposed by Förstner [56] (see fig. 2.2(g))

$$F = \frac{\det(\mu)}{\operatorname{trace}^2(\mu)} \tag{2.7}$$

$$C = 1 - \left(\frac{\lambda_1 - \lambda_2}{\lambda_1 + \lambda_2}\right)^2 = \frac{4\,\det(\mu)}{\operatorname{trace}^2(\mu)} \tag{2.8}$$

where C measures the *eccentricity* of the ellipse associated with the autocorrelation matrix μ, with axes given by its eigenvectors (see fig. 2.2(h)). A point is selected as interest point if both $F > th_F$ and $C > th_C$ where th_F is usually chosen in $[0.5, 1.5]\,\overline{F}$, with $\overline{F}$ the mean value of F over the image, and C ranges in $[0.5, 0.75]$. The SFOP (Scale-invariant Feature OPerator) detector [55] is an extensions in the scale-space of the Förstner detector which unifies different types of features within the same framework by using the general spiral feature model by Bigün [15].

2.2.4 The Shi and Tomasi detector

Shi and Tomasi [152, 165] proposed to use the minimum eigenvalue (see fig. 2.2(i))

$$S = \lambda_2 \tag{2.9}$$

and to take the local maxima greater than a threshold $S > th_S$.

2.2.5 The adaptive non-maximal suppression

Most feature detectors look for local maxima in the 8-neighbourhood to extract the feature. This can lead to an irregular distribution of the features, for instance there can be more features in regions of higher contrast. To alleviate this problem the *adaptive non-maximal suppression* introduced in [18] can be used. A point is selected as feature if it attains to a local maximum which is significantly greater

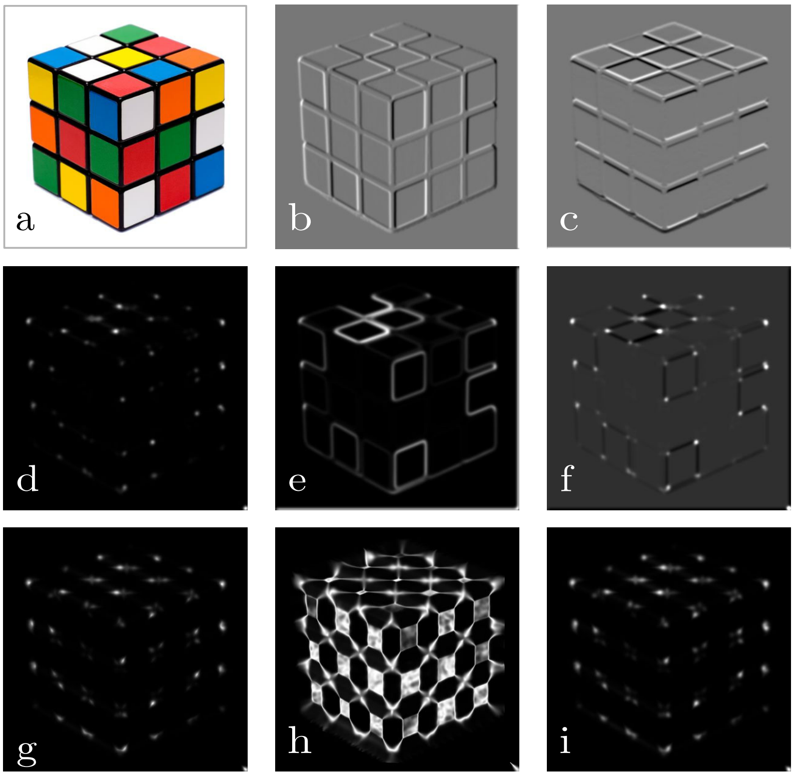


Figure 2.2: The original image (a) and the derivatives of the luminance map I_x, I_y (b,c). The determinant $\det(\mu)$ (d) and the squared trace $\text{trace}^2(\mu)$ (e) of the autocorrelation matrix used to compute the cornerness function H (f). The maps F, C used by the Förstner detector (g,h). The map of minimum eigenvalue S used by Shi and Tomasi (i). Brighter points of H, F and S indicate higher cornerness values

than all its neighbours within a radius r (see fig. 2.3). The adaptive non-maximal suppression can be done efficiently using a sorted list [161].

In a similar way in [140] instead of using a global threshold, the image is divided into different equal sized subregions and an adaptive threshold is applied for each subregion in order to optimize the point distribution, avoiding an unfavourable accumulations of features.

2.2.6 Scale and affine extensions to corner detectors

The autocorrelation matrix can be easily extended to the scale-space by using the equation 1.23, where the window size is given in terms of the integration scale σ_I, while the image resolution is computed according to the differentiation scale σ_D. Features at different scales can be extracted by varying σ_I, σ_D, where the ratio between the two standard deviations is kept fixed to reduce the computational complexity, i.e. $\sigma_I = s\sigma_D$ with $s \in \mathbb{R}$. A finer approach is done by the Harris-

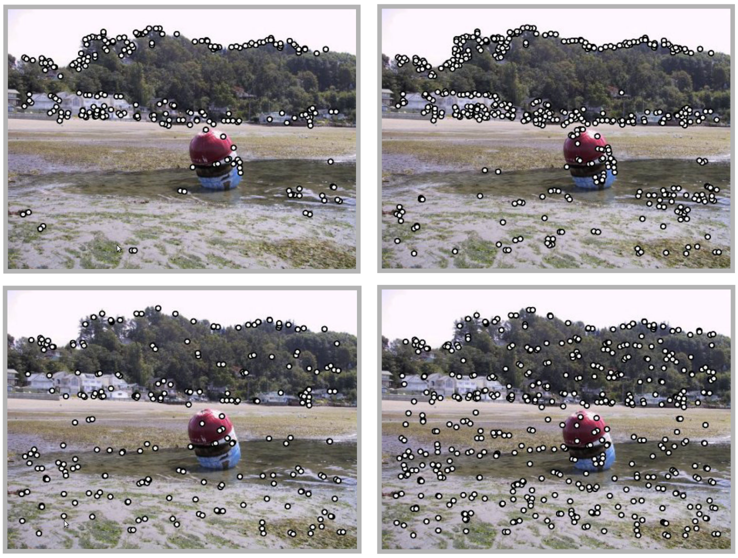


Figure 2.3: The strongest 250 corners (left) and 500 cornes (right), as best local maxima (top) and using the adaptive non-maximal suppression (bottom) by decreasing the radius r. It can be seen that in the latter case features are better distributed along the image. Images from [18]

Laplace detector [112] by computing the characteristic scale given by the Laplacian of the image (1.25). For each Harris corner $(\mathbf{x}, \sigma)$ extracted at scale σ, the local maximum over a set of scales near the current scale, i.e. $\overline{\sigma} = [0.7, \ldots, 1.4]\,\sigma$, is searched and the local neighbourhood of $\mathbf{x}$ is inspected to maximize the function H at the new scale. These steps are repeated until no change in the position or in the scale occurs (see fig. 2.4).

Corner detection can be extended in similar way in the case of the affine scale-space [95, 3, 112], where the shape of the corner patch is obtained from the ellipse associated to the autocorrelation matrix μ. The Harris-affine detector [112] is the most popular affine covariant Harris detector. For each extracted Harris corner $(\mathbf{x}, \sigma_I, \sigma_D)$, starting in the uniform coordinate space (see eq. 1.22), the the following steps are repeated until convergence (see fig. 2.5):

- normalize the coordinate space as described by equation 1.20 where the covariance matrix Σ is given by the autocorrelation matrix μ;
- update σ_I to the the best integration scale as done for the Harris-Laplace method;
- select the differentiation scale from $[0.5, \ldots, 0.75]\,\sigma_D$ which maximizes the ratio λ_2/λ_1 for μ;
- update the corner localization, by choosing from the point neighbours which maximized the H (2.6) as in the Harris-Laplace detector.

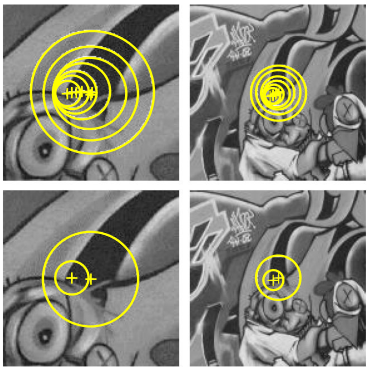


Figure 2.4: Initial points selected at different scales (top) and the final points selected by the Harris-Laplace detector by using the characteristic scale given by the image Laplacian (bottom) for different image scale factors (left, right). Images from [111]

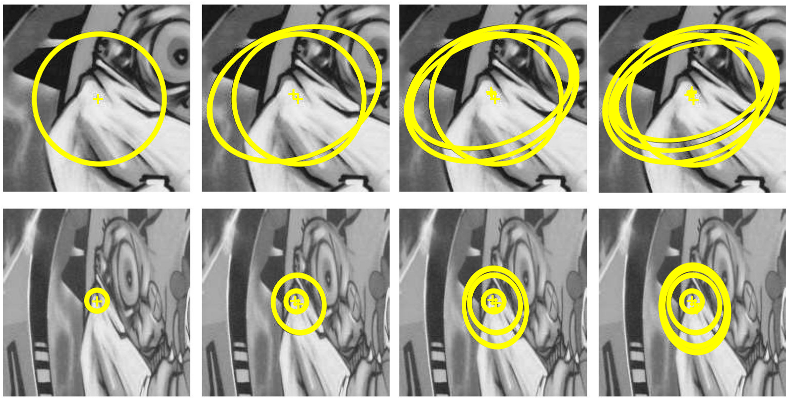


Figure 2.5: The affine refinement steps performed by the Harris-affine detector (left to right) for two corresponding regions (top, bottom). Images from [111]

Another strategy which considers clusters of Harris corners is followed by the MSCC (Maximal Stable Corner Cluster) detector [60]. These clusters are obtained by building the MST (minimum spanning tree) [31] of the extracted features and then by cutting the edges of the MST for different increasing thresholds. The connected components obtained for each threshold represent the final features. The feature patch shapes are obtained by using the covariance matrix given by the coordinates of the corners which form a cluster (see fig. 2.7).

2.2.7 Detectors based on the Hessian matrix

The Hessian matrix (eq. 1.24) has also been used as a corner detector [5]. When a surface is expressed as a *Monge patch* [30], i.e. by a triple $(x, y, I(x, y))$, as an image

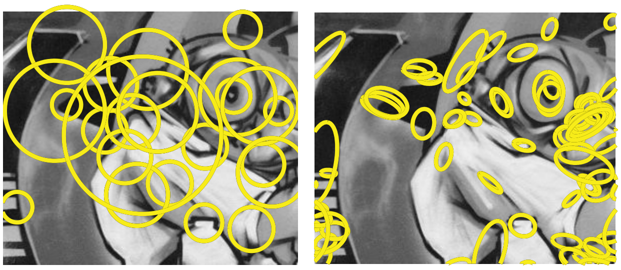


Figure 2.6: Features detected by the Harris-Laplace detector (left) and by the Harris affine detector (right). Images from [177]

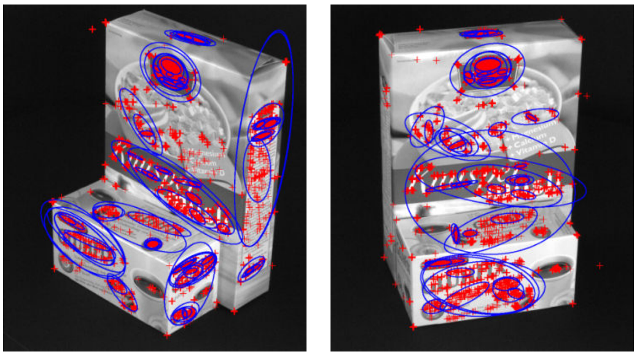


Figure 2.7: Features detected by the MSCC detector (blue ellipses) by clustering the Harris corners (red crosses) on two different views of the same object. Images from [60]

commonly is, the *Gaussian curvature* [30] of a point is

$$\mathcal{K} = \kappa_1 \kappa_2 = \frac{\det(\mathcal{H})}{(1+\mathcal{M})^2} \tag{2.10}$$

where κ_1, κ_2 are the *principal curvatures*, $\mathcal{H}$ is the Hessian and $\mathcal{M}$ is the squared gradient magnitude. It is known from the differential geometry that

- $\kappa_1, \kappa_2 > 0$ for an *elliptic point*;
- $\kappa_1, \kappa_2 = 0$ for a *parabolic point*;
- $\kappa_1, \kappa_2 < 0$ for a *hyperbolic point*;

moreover the denominator of eq. 2.10 is always positive and can be neglected in this classification. It was shown [40] that on both the edge sides of a corner (here a corner means a junction) there is an elliptic and an hyperbolic part and only an elliptic maximum (positive maximum along all directions) with no hyperbolic maxima (negative minima along all directions). Moreover the use of the Hessian matrix is robust in detecting corners but not in their localization, as the point location is pushed away from the corner. This make the Hessian based detectors more appealing to detect blob structure near corners.

2.2.8 Junction detectors

Another junction detector was proposed by Kitchen and Rosenfeld [81]. They first detect the edges on the image by a non-maxima suppression [174] on the gradient magnitude and next the cornerness function is computed as the the product of the curvature for a plane curve times the gradient magnitude

$$K = \frac{L_{xx}L_y^2 - 2L_{xy}L_xL_y + L_{yy}L_x^2}{L_x^2 + L_y^2} \tag{2.11}$$

This corner detector is not robust to noise as it relies on second-order derivatives and has a poor localization rate [40], however it was successively extended in the first example of automatic scale selection by Lindeberg [93]. In the first phase the scale-space formulation of K is used to locate the possible corner candidates and in the next step the localization of the point $\mathbf{x}$ is improved iteratively by solving

$$\min_{\mathbf{x}\in\mathbb{R}^2} \int_{\mathbf{x}'\in\mathbb{R}^2} D(\mathbf{x},\mathbf{x}')^2\, w(\mathbf{x}'-\mathbf{x})\, d\mathbf{x}' \tag{2.12}$$

where w is a weighting function, for instance a Gaussian, and D is the distance function (see fig. 2.8). The minimization is performed by considering that the direction, given by a point on the the edge of a corner and the centre of the corner itself, should be perpendicular to the edge gradient in that point, i.e. $D = 0$ (see fig. 2.8) for

$$D(\mathbf{x},\mathbf{x}') = \nabla^T_{I(\mathbf{x}')}\,(\mathbf{x}-\mathbf{x}') \tag{2.13}$$

as proposed by Förstner and Gülch [56].

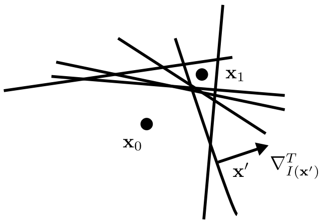


Figure 2.8: The initial localization $\mathbf{x}_0$ of the corner is refined using eq. 2.12 obtaining the final estimate $\mathbf{x}_1$. Image from [93]

A more sophisticated approach was the Kona (corner in Hindi) detector [130], which fits a feature patch to the best piecewise constant junction model (see fig. 2.9) by using the dynamic programming paradigm.

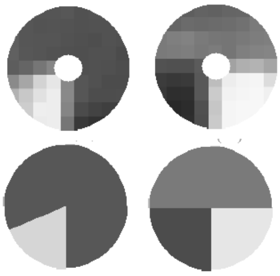


Figure 2.9: Initial image patches detected by Kona (top) and the final junction estimations (bottom). Image from [130]

2.2.9 Corner detector based on kernel masks

Another class of corner detectors is based on the difference between the intensity values of the points within a kernel mask and its centre [154, 172, 141]. These detectors can detect other features more than corners and do not rely on derivatives. Moreover, they are very fast, with good results on synthetic test images but they are less performing on real data and they are not affine invariant. The SUSAN (Smallest Univalue Segment Assimilating Nucleus) detector [154], places a circular mask $\mathcal{R}$ centred on the point $\mathbf{x}$ with radius t (see fig. 2.10) and computes the following function

$$N(\mathbf{x}) = \sum_{\mathbf{x}_i \in \mathcal{R}} e^{\frac{(I(\mathbf{x}_i)-I(\mathbf{x}))^6}{t}} \tag{2.14}$$

The final cornerness function S is

$$S(\mathbf{x}) = \begin{cases} th_g - N(\mathbf{x}) & \text{if } N(\mathbf{x}) - th_g < 0 \\ 0 & \text{otherwise} \end{cases} \tag{2.15}$$

where th_g is the threshold value used to select the corners.

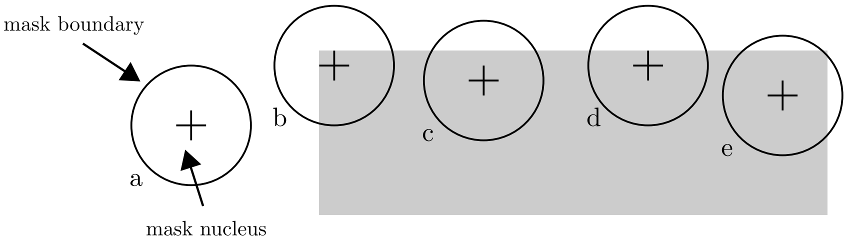


Figure 2.10: According to the percentage of the points inside the kernel mask $\mathcal{R}$ with intensity values close to that of the central point $\mathbf{x}$ inside the mask, the SUSAN detector can classify a region as flat zone (a,c,e), as a corner (b) or as an edge (d). Image from [154]

In a similar way the FAST (Features from Accelerated Segment Test) detector [141], considers a point as a corner if there is a predefined number of continuous pixels with intensity value less than that of point in the centre of the kernel mask. The detector proposed by Trajkovic and Hedley [172] uses opposing pixels on the kernel plus the central kernel pixel. If $\mathbf{x}_i, \mathbf{x}'_i$ are the opposing pixels on the diameter of the kernel mask $\mathbb{R}$ centered in $\mathbf{x}$ (see fig. 2.11), the cornerness function

$$T(\mathbf{x}) = \min_{\mathbf{x}_i \in \mathbb{R}} \left(I(\mathbf{x}_i) - I(\mathbf{x})\right)^2 + \left(I(\mathbf{x}'_i) - I(\mathbf{x})\right)^2 \tag{2.16}$$

is low for plane regions and edges, as shown in fig. 2.11, however this detector is not robust to noise, though it is computational efficient.

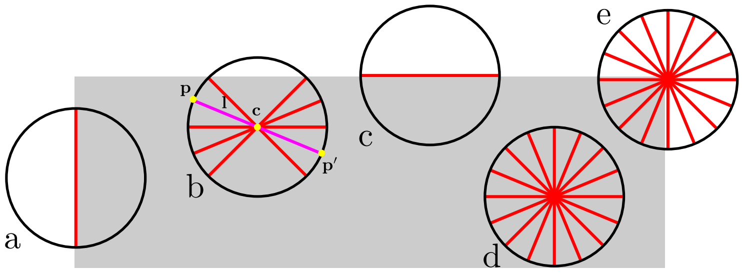


Figure 2.11: The Trajkovic detector uses the information provided by opposing points $\mathbf{p}$, $\mathbf{p}'$ in the mask and its central point $\mathbf{c}$ on a line segment $\mathbf{l}$. If only one segment exists for which the intensity values of $\mathbf{p}$, $\mathbf{p}'$ and $\mathbf{c}$ are similar, the region is on an edge (a,c), while in the case of a flat region this holds for almost all the segments (b,d). Lastly, if the value of the central point is similar only to one of its extrema point values for the majority of segments then a corner is detected (e)

2.3 Blob detectors

2.3.1 The MSER detector

The MSER (Maximaly Stable Extremal Regions) detector [107] is one of the most popular affine covariant blob detector (even if in some new classification it is reported as a region detector, since it does not only detect blobs but more general uniform shapes [177]). It is a watershed based method [150] which sequentially thresholds the image. The most stable connected components with respect to different thresholds are taken as feature regions (see fig. 2.12). In particular a region $\mathcal{R}_i$ is considered as a feature if for all its n nested connected component $\mathcal{R}_1, \ldots, \mathcal{R}_i, \ldots, \mathcal{R}_n$, obtained for different threshold values, it attains to a local minimum for the function $q_i = |\mathcal{R}_{i+\Delta} - \mathcal{R}_{i-\Delta}|/|\mathcal{R}_i|$, where $|\cdot|$ is the cardinality of the connected component in pixels and $\Delta \in \mathbb{N}$ is a user defined parameter. The final feature patch shape is obtained by using the covariance matrix given by the coordinates of the connected component (see fig. 2.13). The MSER detector is very robust to affine transformation [114], especially for non textured images, but it suffers from blurred images. An extension to resolve this issue was proposed in [54], where the MSER algorithm is executed at different levels of a scale pyramid, while another extension was proposed in [53] to improve the detection for colour images.

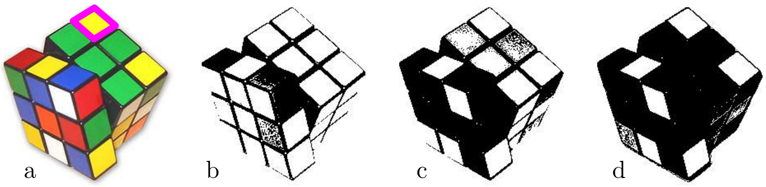


Figure 2.12: Original image (a) and sequential thresholds (b-d). It can be noted that the image patch highlighted in the original image is very stable across the different thresholds

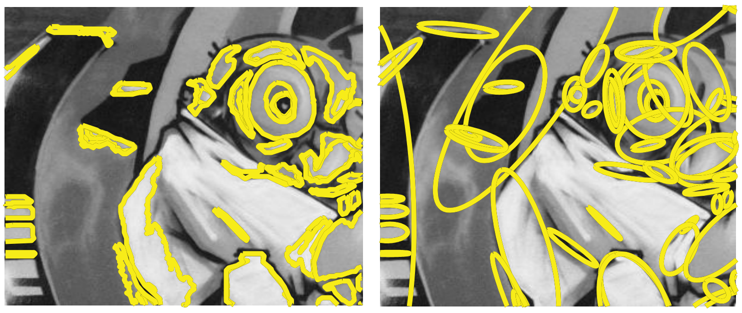


Figure 2.13: Original features detected by the MSER (left) and the associated elliptical patches (right). Image from [177]

The MSER detector has also been employed in the local affine frame [108]. Given three affine covariant points for each MSER feature, for instance the associated ellipse centre and the two axes scaled to the unit according to their ratio, the local affine frame is obtained by an affine normalization which maps the point to a canonical frame (see fig. 2.14). For each local affine frame L a triplet of points associated with another MSER feature which is close to L in the affine transformation inducted by L itself is considered using polar coordinates. The descriptor so obtained has been used in object recognition with geometric hashing [27] as an extension of the Hough transform to vote for the model supported by L.

2.3.2 Hessian based blob detectors

One of the first blob detector was proposed by Lindeberg [92]. The points $(\mathbf{x}, \sigma)$ in the scale-space which are local maxima or minima for the Laplacian of a Gaussian $\mathcal{L}$ (see eq. 1.25) are selected as blobs. As it can be seen from fig. 2.15 the Laplacian

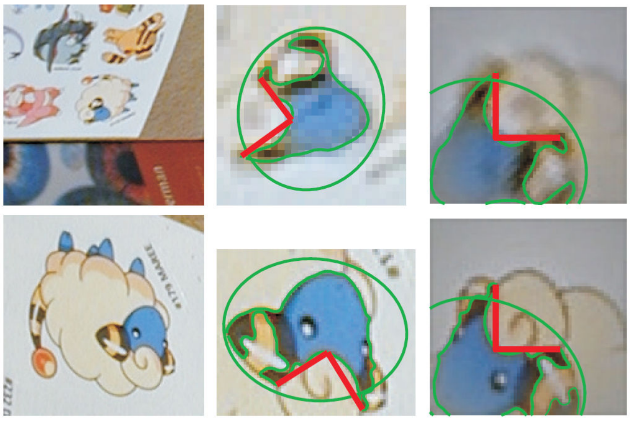


Figure 2.14: Corresponding features extracted by the MSER detector (left column), detected local affine frames using bi-tangants, i.e. the line segments of convex hull bridging concavities (central column), and the final normalized frames (right column). Images from [108]

kernel gives a positive, negative strong response respectively to a light, dark circular blob. Another similar function proposed in [95] is the scale-space determinant of the Hessian det($\mathcal{H}$), which was first introduced as a corner detector by Beaudet [5]. As it can be seen from fig. 2.15, the Laplacian, i.e. the trace of the Hessian matrix, and the determinant of the Hessian, correspond to similar filter, where the latter is more peaked.

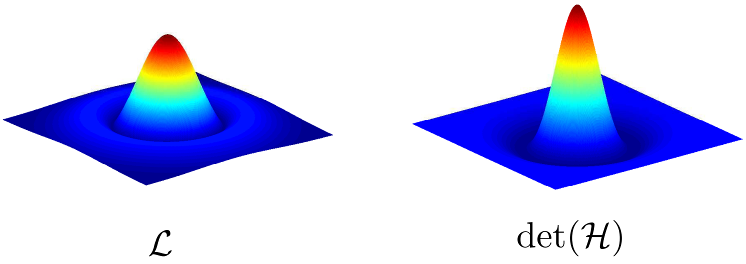


Figure 2.15: The Laplacian (left) and the determinant of the Hessian (right) of a Gaussian kernel

Several affine scale-space detectors have been proposed [95, 3, 112, 107, 176, 76] that try to iteratively refine the scale by using the Laplacian $\mathcal{L}$ as the characteristic scale and the Hessian determinant det($\mathcal{H}$) as the covariance matrix Σ. For instance the Hessian-Laplace [112] and the Hessian-affine [112] detectors (see fig. 2.16) are the equivalent blob detectors of the Harris-Laplace and the Harris-affine corner detectors respectively, where the cornerness function H (see eq.2.6) is replaced by det($\mathcal{H}$). Both the detectors have been proved to give robust and stable features with respect to the scale and to affine transformations respectively.

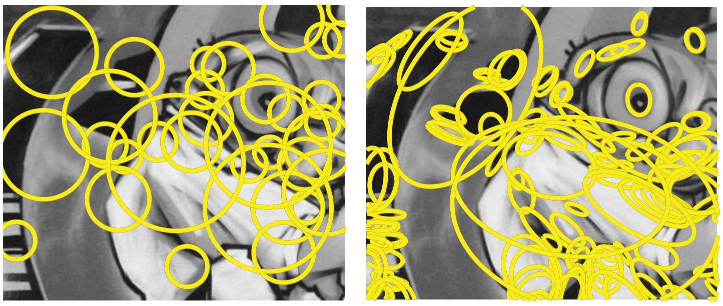


Figure 2.16: Features detected by the Hessian-Laplace detector (left) and by the Hessian affine detector (right). Images from [177]

2.3.3 The SURF detector

The SURF (Speed Up Robust Feature) detector [4] searches for local maxima of the Hessian determinant in the scale-space (see fig. 2.17). For each scale the Hessian determinant is computed efficiently by using a discrete approximation of the Gaussian second order partial derivatives $\widehat{L}_{xx}, \widehat{L}_{yy}, \widehat{L}_{xy}$ (see fig. 2.18), which can be done very fast by *integral images* [182]. A proper weight $w \approx 0.9$ is introduced to correct the Gaussian partial derivatives in order to compute a better approximation $\det(\mathcal{H}_{approx})$ of the Hessian determinant

$$\det(\mathcal{H}_{approx}) = \widehat{L}_{xx}\widehat{L}_{yy} - (w\widehat{L}_{xy})^2 \tag{2.17}$$

Differently from other approaches, the scale is not obtained by decreasing the image size after a proper smooth [20, 35], but by increasing the discrete kernel dimension. The final localization of the maxima is improved by using eq. 1.34 (see fig. 2.19).

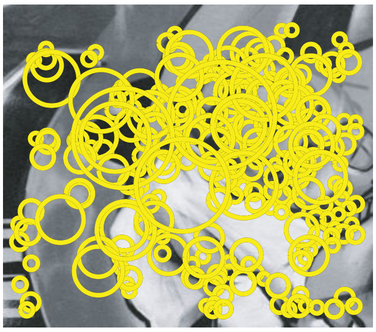


Figure 2.17: Features detected by the SURF detector. Image from [4]

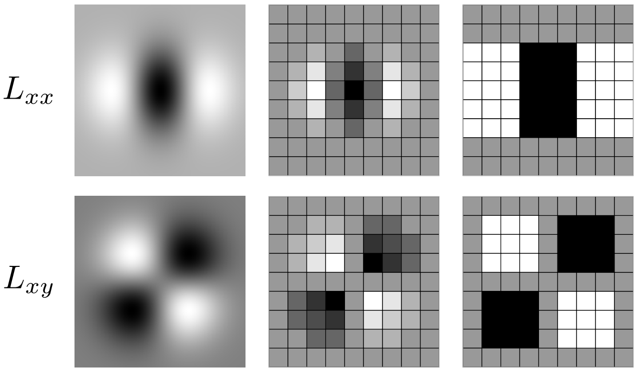


Figure 2.18: Gaussian second order partial derivatives (left), their quantized versions (center) and the approximations used by the SURF detector (right). The partial derivatives L_{yy} and L_{yx} are obtained by rotating the corresponding kernels by 90° degrees. Image adapted from [4]

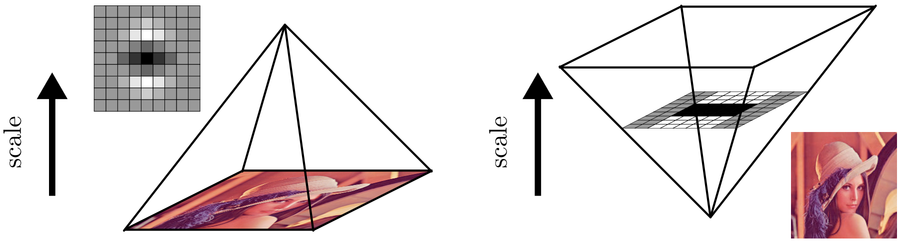


Figure 2.19: While it is the common approach to downsample the image, holding the kernel dimension fixed to be fast (left), the SURF detector increases the kernel dimension while the image size remain fixed by integral images (right). Image adapted from [4]

2.3.4 The SIFT detector

The most popular blob detector is definitively the SIFT (Scale Invariant Feature Transform) detector [100]. It is based on the DoG (Difference of Gaussians) operator $\mathcal{D}$ computed on an scale pyramid image representation

$$\mathcal{D} = (g_\sigma - g_{\sigma'})* \tag{2.18}$$

where $*$ mean the convolution and $\sigma, \sigma' \in \mathbb{R}$. The difference of Gaussians can be seen as an approximation of the Laplacian of Gaussian $\mathcal{L}$ because from the diffusion equation [100]

$$\frac{\partial}{\partial\sigma} g_\sigma = \sigma\,\nabla^2 g_\sigma = \frac{\mathcal{L}}{\sigma} \tag{2.19}$$

by finite difference approximation it follows that

$$\frac{\mathcal{L}}{\sigma} = \frac{\partial}{\partial\sigma} g_\sigma \approx \frac{g_{k\sigma} - g_\sigma}{k\sigma - \sigma} \;\Rightarrow\; g_{k\sigma} - g_\sigma \approx (k-1)\mathcal{L} \tag{2.20}$$

where $\mathcal{L}$ is intended to be scale-space normalized. The difference of Gaussians can be computed efficiently by smoothing each image octave with different Gaussian kernels and then by subtracting them (see fig. 2.20). Local scale-space maxima are detected and their localization improved by using eq. 1.34.

Moreover in order to drop false features due to edges, since the eigenvalues $\lambda_1, \lambda_2 = \kappa\lambda_1$ of $\mathcal{H}$, with $\lambda_1 > \lambda_2$, are proportional to the principal curvatures of the image, the candidate features can be discarded as done for the cornerness response H (see eq. 2.6), by a threshold th_s on the eigenvalue ratio κ

$$\frac{\text{trace}(\mathcal{H})^2}{\det(\mathcal{H})} = \frac{(\lambda_1+\lambda_2)^2}{\lambda_1\lambda_2} = \frac{(\lambda_1+\kappa\lambda_1)^2}{\kappa\lambda_1^2} = \frac{(1+\kappa)^2}{\kappa} > th_s \tag{2.21}$$

Though it is only rotation and scale invariant, the SIFT detector is widely used because it is computational efficient and it provides good results, even for relatively high perspective transformations (see fig. 2.21). An affine invariant extension of the SIFT detector has been proposed as for other detectors. The ASIFT (Affine SIFT) [120] detector simulates the distortions caused by a variation of the direction of the camera optical axis by transforming artificially the image and then executes the SIFT detector for each warped image.

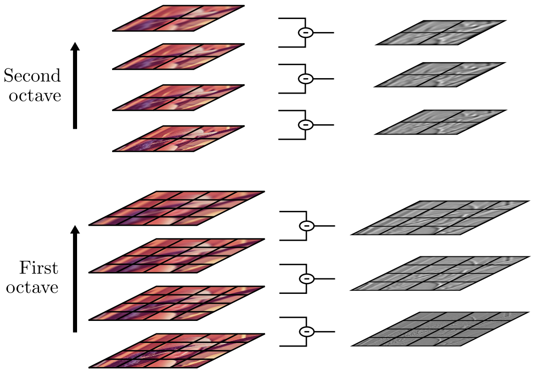


Figure 2.20: For each octave, obtained by smoothing and downsampling the image, the DoG is computed by the difference between successive smoothed versions of the octave. Image adapted from [100]

To be mentioned also the work described in [175], where two feature operators G_1, G_2 have been discovered by using genetic programming [2]

$$\begin{aligned} G_1(\mathbf{x}) &= (g_3 - g_2)*I(\mathbf{x}) \\ G_2(\mathbf{x}) &= g_1 * \det(\mathcal{H}_{I(\mathbf{x})}) \end{aligned} \tag{2.22}$$

As it can be seen, the former is a difference of Gaussians, while the latter is the Hessian determinant convolved with a Gaussian kernel. They both provide another clue on the goodness of the DoG and the det($\mathcal{H}$) operators as detectors.

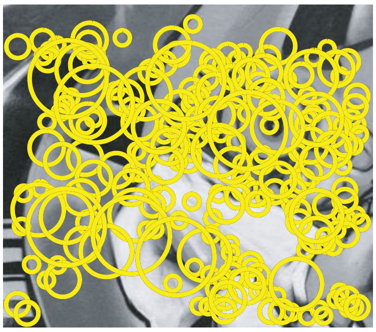


Figure 2.21: Features detected by the SIFT detector. Image from [177]

2.3.5 Other blob detectors

Another blob detector similar to the MSER was proposed in [176]. The IBR (Intensity Based Region) detector [176] (see fig. 2.22) starts from an image extremum $\mathbf{x}_0$ and for each ray $\mathbf{x}_{r,\theta}$, with radius r along the direction θ exiting from $\mathbf{x}_0$, evaluates the function

$$T_{r,\theta}(\mathbf{x}_0) = \frac{\operatorname{abs}\left(I(\mathbf{x}_{r,\theta}) - I(\mathbf{x}_0)\right)}{\max\left(\frac{1}{r}\int_{r'=0}^{r} \operatorname{abs}\left(I(\mathbf{x}_{r',\theta}) - I(\mathbf{x}_0)\right) dr', \varepsilon\right)} \tag{2.23}$$

where ε is a small value to avoid an accidental division by zero. The radii for the possible directions θ for which $T_{r,\theta}$ gives a maximum are connected to form the boundaries of the feature patch and the final elliptic region is obtained by the covariance matrix of the edge coordinates (see fig.2.23).

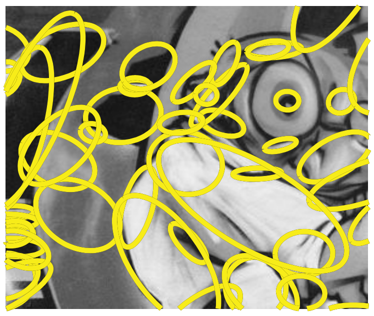


Figure 2.22: Features detected by the IBR detector. Image from [177]

Though they are not pure blob detectors, the EBR (Edge Based Region) detector [176] and the salient region detector [76], should also be mentioned. The EBR (see fig. 2.24) starts by extracting Harris corners, to be used as anchor points, near edges obtained by the Canny edge detector [21]. For each anchor point $\mathbf{p}$ two points $\mathbf{p}_1, \mathbf{p}_2$, move at the same speed on the edge in opposite directions, drawing a family

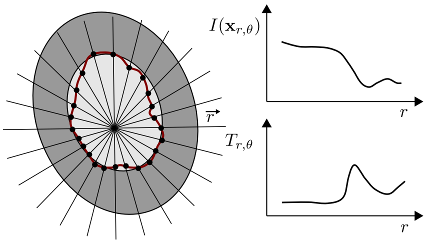


Figure 2.23: Starting from an intensity extremum, the function T is computed along each ray r. Points along the rays for which maxima of T are connected (red boundary) and the final corresponding feature ellipse is extracted. Image adapted from [176]

of parallelograms. A set of functions related on the centre of mass $\mathbf{q}$ of the parallelogram which measure how much it is pushed away from the diagonals of the parallelogram has been designed by the authors so that the extrema of the functions are affine invariant. When the points $\mathbf{p}_1, \mathbf{p}_2$ move away and extrema for any of these functions are met, the corresponding parallelogram is taken as a feature (see fig. 2.25).

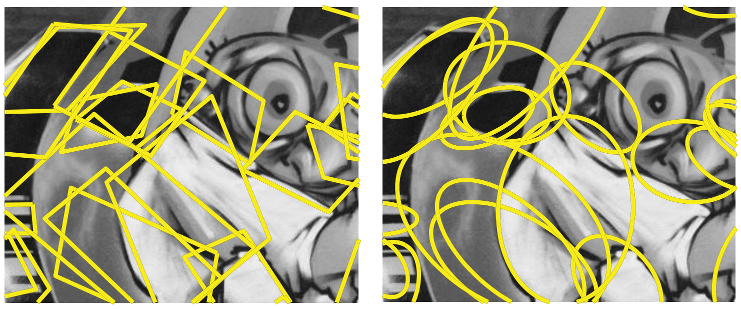


Figure 2.24: Original features detected by the EBR detector (left) and the associated elliptic patches (right). Images from [177]

The salient region detector characterizes features by the entropy (see fig. 2.26). Given the probability $p(v, \mathbf{x}, \sigma)$ of the intensity value $v \in \mathbb{R}^+$ in the region centered in $\mathbf{x}$ at scale σ, the entropy of the region is

$$H_\sigma(\mathbf{x}) = -\int_{v \in \mathbb{R}^+} p(v, \mathbf{x}, \sigma) \log_2 p(v, \mathbf{x}, \sigma) dv \tag{2.24}$$

A further weight is required to discriminate between unstructured random region and meaningful region

$$W_\sigma(\mathbf{x}) = \sigma \int_{v \in \mathbb{R}^+} \text{abs}\left(\frac{\partial}{\partial \sigma} p(v, \mathbf{x}, \sigma)\right) dv \tag{2.25}$$

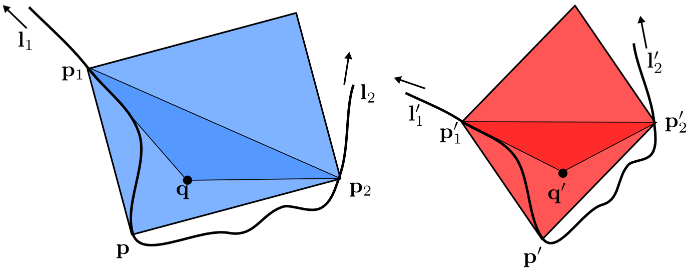


Figure 2.25: Given a patch (left) and its affine transformation (right), for each anchor point, respectively $\mathbf{p}$ and $\mathbf{p}'$, two points $\mathbf{p}_1, \mathbf{p}_2$ and $\mathbf{p}'_1, \mathbf{p}'_2$ move at the same speed on the edge in opposite directions. A set of functions related on the centres of mass, respectively $\mathbf{q}$, $\mathbf{q}'$ of the corresponding parallelograms are computed. When the points move away and an extremum of any of these functions is met a feature is detected. Image from [176]

since when the neighbourhood of a region with random values increases, there should not be variations in the probability of the intensity values (see fig. 2.27). The saliency function is

$$S = H_{\sigma'} W_{\sigma'} > th_s \tag{2.26}$$

where σ' is the scale value for which the function H attains to a maximum. After applying the global threshold th_s, points close in the scale-space are clustered together. As proposed in its first implementation, the salient detector is not affine invariant, but it has been extended by iteratively refining the scale and the shape of the region until no variation in both scale and position is present. While the best scale search is achieved by searching for the maxima of H, the (elliptic) shape is searched by applying some small deformation and by retrieving the one which maximizes W. The salient detector performances are lower than those of other detectors [114] but it has successfully been used in recognition tasks [77].

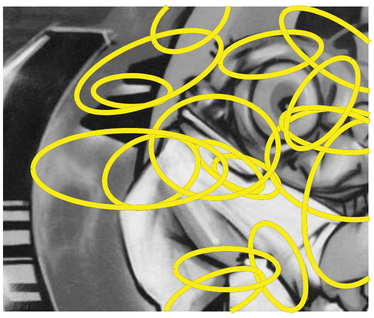


Figure 2.26: Features detected by the Salient region detector. Image from [177]

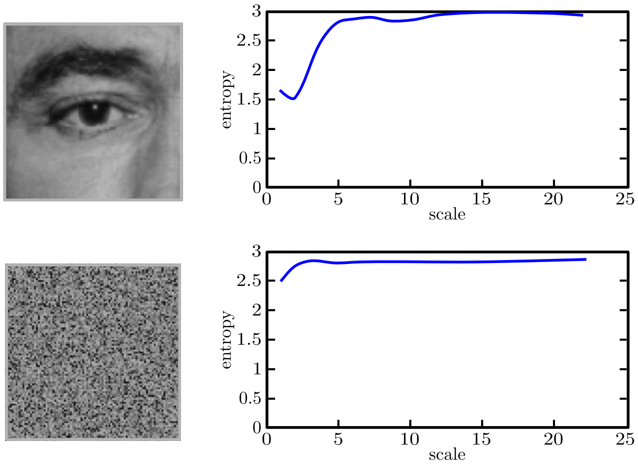


Figure 2.27: Original image (top) and a random permutation of its pixels (bottom). For a meaningful region there is a fast variation of the entropy as the scale increases, which is not present for a patch of random points. Image from [76]

Chapter 3

Feature descriptors

3.1 Introduction

A feature descriptor for the k-th feature $\mathcal{F}_k$ is a numeric vector which embodies feature data information $\mathcal{F}_k = [f_{k1}, \ldots, f_{kn}]^T \in \mathbb{R}^n$. The descriptor vectors are used to compare features by a similarity/dissimilarity function $d(\mathcal{F}_1, \mathcal{F}_2)$. A good feature vector should be compact, discriminant and robust to noise. The simplest feature detector is the feature patch itself, where patches can be compared by the simple *SSD* (*Sums of Squared Distance*). However, the use the patch as descriptor is not a good choice because it has an high dimension, it is very sensible to small variations and it is very redundant.

The state of the art descriptors can be divided into the following main classes: *distribution based descriptors*, *differential descriptors* and *spacial-frequency based descriptors*.

Spacial-frequency descriptors include texture analysis techniques such as Gabor filters [66], DCT (Discrete Cosine Transform) [66] or wavelets [66], however their performances are relatively poor in comparison to other descriptors.

3.2 Distribution based descriptors

3.2.1 Patch normalization

Distribution based descriptors are mainly based on the histograms which represent the distribution of some particular data relationship between features. In order to compute the distribution, the feature patch has to be normalized. The normalization of the intensity values by mean and standard deviation makes the patch invariant to affine illumination changes (see Sec. 1.3.6). The translation factor is removed by fixing the coordinate origin into the feature central point, while for affine invariant detector the relative covariance matrix Σ is used to remove the shear (see Sec. 1.3.4). The affine coordinate change described in eq. 1.20 is applied to the covariance matrix, so that the elliptic neighbourhood is normalized to a circular one and the scale information is used to normalize to a unit scale. More in detail, the circular patch radius of the normalized feature, usually chosen to be 3σ where σ is the feature scale, is mapped to the radius of the normalized patch r, usually 20 pixels, that is

$$\mathbf{x}' = \frac{r}{3\sigma}\mathbf{x} \tag{3.1}$$

3.2.2 Orientation normalization

The orientation invariance is obtained by a rotation of the patch toward the direction of the dominant gradient orientation (see Sec. 1.3.2). The most common used approach was proposed by Lowe [100]. The gradient orientation histogram of the patch $\mathcal{R}$, weighted by the gradient magnitude and by a Gaussian window centered in the patch center $\mathbf{x}_c$, is computed

$$h_\theta = \sum_{\mathbf{x}\in\mathcal{R}_\theta} g_\sigma(\mathbf{x}-\mathbf{x}_c)\sqrt{L_x^2\left(I(\mathbf{x})\right)+L_y^2\left(I(\mathbf{x})\right)} \tag{3.2}$$

where $\mathcal{R}_\theta$ is the set of points in the normalized patch with gradient orientation $\theta = \arctan(L_y/L_x)$. The dominant gradient orientation is given by

$$\Theta = \arg\max_\theta h_\theta \tag{3.3}$$

In order to increase the robustness of the estimation, Lowe proposes in its original paper to also take the orientations for which the histogram bins are within the 80% of the maximal bin and to use a parabolic fitting (see Sec. 1.3.5) to increase the accuracy of the estimation, however these last steps are usually neglected [113].

Another approach was proposed by Mikolajczyk [111], which uses the gradient orientation in the feature centre corrected by the average gradient orientation of the local neighbourhood

$$\Theta' = \theta_{\mathbf{x}_c} - \frac{\sum_{\mathbf{x}\in\mathcal{R}} g_{\sigma/3}(\mathbf{x}-\mathbf{x}_c)\left(\theta_{\mathbf{x}_c}-\theta_{\mathbf{x}}\right)}{\sum_{\mathbf{x}\in\mathcal{R}} g_{\sigma/3}(\mathbf{x}-\mathbf{x}_c)} \tag{3.4}$$

After the patch is fully normalized the descriptor is computed.

3.2.3 The rank and the census transforms

The first distribution based detectors can be found in [188] where the rank and the census transforms were introduced. The rank transform is the number of pixels $\mathbf{q}$ in the feature patch $\mathcal{R}$ which have an intensity value greater than the value of the central pixel $\mathbf{c}$

$$R(\mathcal{F}) = |\{\mathbf{q} : I(\mathbf{q}) > I(\mathbf{c}) \wedge \mathbf{q} \in \mathcal{R}\}| \tag{3.5}$$

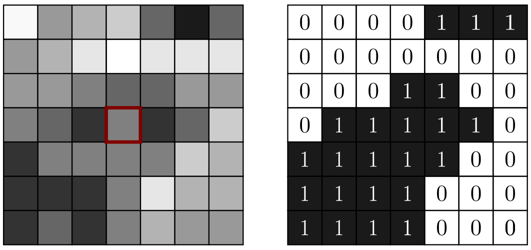
where $|\cdot|$ is the cardinality of the set (see fig. 3.1). The concatenation $\bigoplus$ of the boolean values given by the evaluation of the inequality used for the rank transform

$$B(\mathbf{p},\mathbf{q}) = \begin{cases} 1 & if\ I(\mathbf{q}) > I(\mathbf{c}) \\ 0 & \text{otherwise} \end{cases} \tag{3.6}$$

can be used instead to obtain the census transform (see fig. 3.1)

$$C(\mathcal{F}) = \bigoplus_{\mathbf{q}\in\mathcal{R}} B(\mathbf{q},\mathbf{c}) \tag{3.7}$$

which can be evaluated by using the *Hamming distance* [74].



0	0	0	0	1	1	1
0	0	0	0	0	0	0
0	0	0	1	1	0	0
0	1	1	1	1	1	0
1	1	1	1	1	0	0
1	1	1	1	0	0	0
1	1	1	1	0	0	0

Figure 3.1: Given a pixel, highlighted in red, and a window centered in it (left), each pixel can be labelled according to the sign of the difference between its value and that of the central pixel (right). The rank transform counts the number of pixel with the same label of the central pixel, while the census transform is the ordered concatenation of the labels of each pixel

3.2.4 Spin images

Another approach is the spin image [85], originally developed for three-dimensional range images [75]. A spin image is a two-dimensional soft histogram where one dimension is given by the distance from the centre of the feature patch and the other one by the range of the intensity values (see fig. 3.2)

$$H_P(d, v) = \sum_{\mathbf{q} \in \mathcal{R}} e^{-\left(\frac{\|\mathbf{q}-\mathbf{c}\|-d}{2\sigma^2} + \frac{I(\mathbf{q})-v}{2\sigma'^2}\right)} \tag{3.8}$$

where $\|\cdot\|$ is the Euclidean distance and σ, σ' are the smoothing factors. The final descriptor is obtained by concatenating the histogram bins for the radius set $\mathcal{D}$ and the orientation set $\mathcal{V}$

$$P(\mathcal{F}) = \bigoplus_{i \in \mathcal{D}, j \in \mathcal{V}} H_P(d, v) \tag{3.9}$$

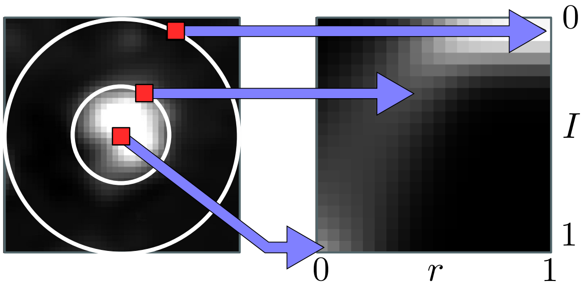


Figure 3.2: A feature patch (left) and the corresponding spin image (right). The bin locations of some pixels, according to their intensity value I and the distance r from the centre, are highlighted. Image adapted from [85]

3.2.5 Shape context

The shape context was proposed in [11]. It is a three-dimensional histogram of locations and orientations of edge points. The bins are arranged into a log-polar grid, which simulates the human eye behaviour [184]. Each point in the bins is weighted by its gradient magnitude and the gradient orientation of the central point in the grid is used as the reference orientation (see fig. 3.3(a-f)). To be mentioned that the shape context descriptor has been used in a more sophisticated approach to recognize deformation in objects [11], where correspondences are found by solving the bipartite graph matching problem [31] and by obtaining the best shape transformation on the thin plate model [43] (see fig. 3.3(g)).

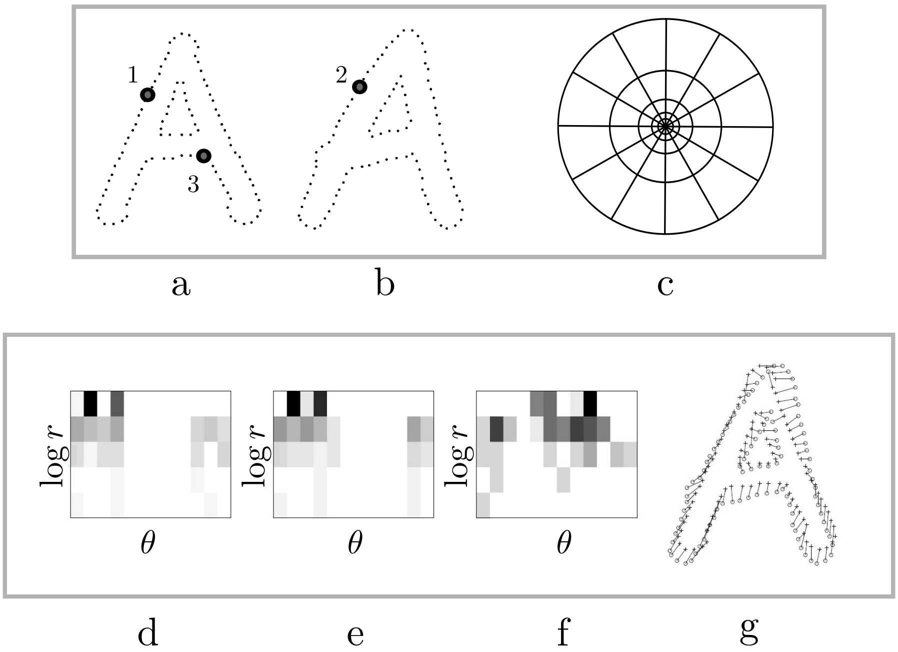


Figure 3.3: Two similar shapes (a-b) and the log-polar grid used by the shape context descriptor (c). The two-dimensional histograms (d-f) are respectively the shape context representations of the points 1-3. The thin plate model is used to compute the best transformation between the two shapes (g). Image from [11]

3.2.6 The SIFT descriptor

The SIFT descriptor [100] introduced by Lowe is nowadays the most popular feature descriptor. The feature patch, weighted by a Gaussian window centered in the patch $\mathbf{x}_c$, is subdivided by a Cartesian grid and for each cell $\mathcal{R}_k$ of the grid an histogram of gradient orientations, weighted by the gradient magnitudes, is computed (see fig. 3.4)

$$H_S(k,\theta) = \sum_{\mathbf{x}_i \in \mathcal{R}_k,\theta} g_\sigma(\mathbf{x}_i - \mathbf{x}_c)\sqrt{L_x^2\left(I(\mathbf{x}_i) + L_y^2\left(I(\mathbf{x}_i)\right)\right.} \tag{3.10}$$

where $\mathcal{R}_{k,\theta}$ is the set of the points in the k-th cell with gradient orientation θ. In order to obtain a smoothed histogram Lowe proposes to use a trilinear interpolation on the bin dimensions. Each bin entry is multiplied by a weight of $1 - d$ for each dimension, where d is the distance of the point sample from the bin centre. The distance d is measured in units of the histogram bin spacing [100]. In most other implementations the patch is instead convolved with a small Gaussian kernel g_1 [113]. The typical size of the grid is 4×4 for 8 directions so that the resulting histogram size is $4 \times 4 \times 8 = 128$. The histogram is then normalized to the unit to remove the effect of affine illumination transformations. Moreover, to further remove non linear illumination changes due to camera saturation or surface properties, a threshold on each bin is applied so that each normalized bin cannot exceed 0.2 and the histogram is normalized to the unit again.

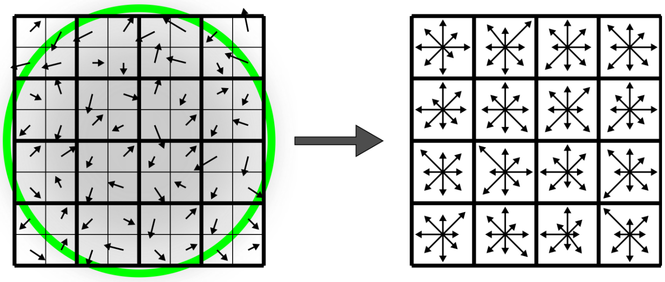


Figure 3.4: The feature patch is divided by a 4×4 grid and the gradient magnitude is weighted by a Gaussian window (left). For each cell grid the gradient orientation histogram is computed (right). Image adapted from [100]

Many extensions of the SIFT descriptor has been proposed in the last decade. The PCA-SIFT descriptor [78] uses the *PCA (Principal Component Analysis)* [74] to reduce the descriptor dimension and remove useless data; other data reduction methodologies have also been applied to the SIFT [24, 72] (see Sec. 4.2.2). The GLOH (Gradient Local Orientation Histogram) detector [113] uses a log-polar grid to compute a histogram of size 272 which is successively reduced by the PCA to 128. Also overlapping grid cells have been employed [37], which seem to improve the descriptor robustness to scale (see fig. 3.5). The search of the best standard deviation for the gradient computation has also been performed in [121] by using the Gabor filters. The RIFT (Rotational Invariant Feature Transform) detector [85] uses concentric rings as bins and obtains the rotation invariance by computing the orientation at each point relative to the direction pointing outward from the center, avoiding the dominant gradient orientation estimation (see fig. 3.6).

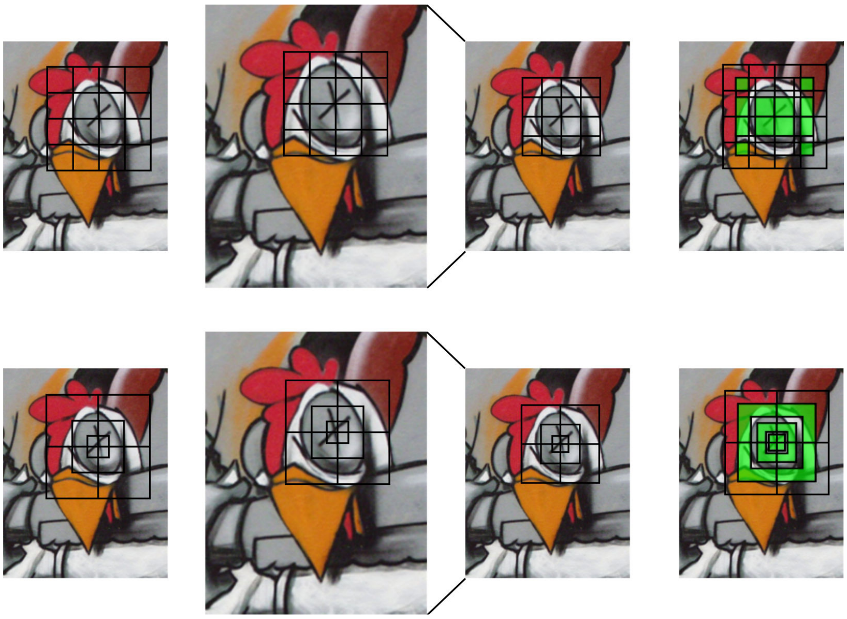


Figure 3.5: A comparison between the SIFT descriptor (top) and the irregular orientation histogram binning (bottom). The descriptors are computed on the original feature patches (first, second columns). By rescaling the second patch (third column) it can be noted a better overlap between cells for the irregular binning method (fourth column). Images from [37]

3.2.7 The DAISY descriptor

Another fast descriptor which has been employed in dense map estimation with good result is the DAISY descriptor [164]. It uses a circular grid with small overlap between cells and, to improve the descriptor robustness, circular cells of increasing radius, weighted by a Gaussian window. The name DAISY is due to its shape (see fig. 3.7). For each cell the gradient orientations weighted by the gradient magnitudes are computed and each cell histogram is normalized to the unit. The DAISY descriptor can be densely computed on the image very efficiently. More in detail, the gradient along the axis directions are first computed using the kernels $[-1\ 1]$, $[-1\ 1]^T$ to obtain the gradient along the direction θ (usually there are 8 directions) as a linear combination

$$\frac{\partial}{\partial\theta}I = \cos\theta\frac{\partial}{\partial x}I + \sin\theta\frac{\partial}{\partial y}I \tag{3.11}$$

The value of a bin in the direction θ for a particular grid cell is just the value of the convolution in the cell centre of the gradient map θ with a Gaussian kernel where the standard deviation is given by the cell radius. A dense computation can then be achieved efficiently by using successive convolutions on the gradient orientation maps.

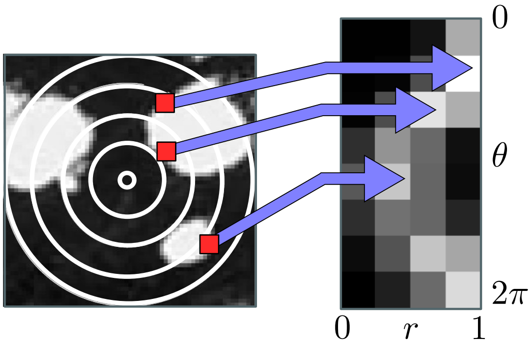


Figure 3.6: A feature patch (left) and the corresponding RIFT descriptor (right). For each pixel the corresponding bin is given by its distance from the centre r and its gradient orientation θ. The reference orientation is given by the direction outward from the center. Bin locations of some pixels are highlighted. Image from [85]

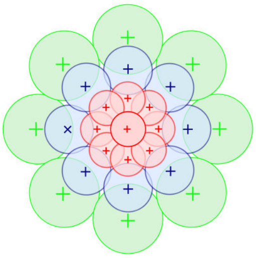


Figure 3.7: The grid cells used by the DAISY descriptor. It can be noted that there is a small amount of overlap between adjacent cells. Image from [164]

3.3 Other descriptors

3.3.1 Generalized color moment

Other descriptors to be mentioned are the invariants up to second order based on the *generalized color moments* with order $p+q$ and degree $a+b+c$ introduced in [116]

$$M_{xy}^{abc} = \iint_{(x,y)\in\mathcal{R}} x^p y^q R^a(x,y) G^b(x,y) B^c(x,y) dxdy \tag{3.12}$$

where $R(x,y), G(x,y), B(x,y)$ are the three color values for the pixel (x,y). The generalized moments characterize the shape, the intensity and the color distribution in the feature neighbourhood.

3.3.2 The geometric blur

The geometric blur [12, 13] is a smoothed version of the signal around a feature point, blurred by a spatially varying kernel. The geometric blur version of a patch centred in $\mathbf{x}_c$ is defined as

$$B_{\mathbf{x}_c}(\mathbf{x}) = I * g_{\alpha\|\mathbf{x}\|+\beta}(\mathbf{x}_c - \mathbf{x}) \tag{3.13}$$

The final descriptor is made up of sampled points of the geometric blur of the patch on a log-polar grid (see fig. 3.8).

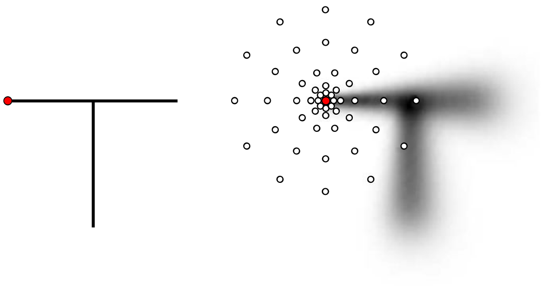


Figure 3.8: A feature patch (left) is extracted and its geometric blur is computed (right), where the patch centre is highlighted by a red dot. The descriptor vector is obtained by sampling only the geometric blur in the location highlighted by the dots. Image from [12]

3.3.3 Differential descriptors

Differential descriptors arise from the Taylor series approximation of a function

$$\begin{aligned} I(x_0 + x, y_0 + y) = I(x_0, y_0) + x\frac{\partial}{\partial x} I(x_0, y_0) + y\frac{\partial}{\partial y} I(x_0, y_0) + \\ + \ldots + \sum_{i=1}^{N} x^p y^{N-p} \frac{\partial^N}{\partial x^p \partial y^{N-p}} I(x_0, y_0) + \mathcal{O}(x^N, y^N) \end{aligned} \tag{3.14}$$

so that the derivatives can be seen as the fingerprints of the function in a local neighbourhood. The *local jet* $\mathcal{J}^N$ for a scale factor σ is defined as a set of local derivatives up to order N [83]

$$\mathcal{J}^N(I(\mathbf{x}), \sigma) = \{L_{i_1,\ldots,i_n}(I(\mathbf{x}), \sigma) : n = 0, \ldots, N; i_k \in \{x, y\}; k = 1, \ldots, n\} \tag{3.15}$$

The derivatives can "be steered" along any direction θ as described in [62] using the components of the local jet

$$L_\theta = L_x \cos\theta + L_y \sin\theta \tag{3.16}$$

and by iterating

$$\begin{aligned} L_{\theta^2} &= L_{xx}\cos^2\theta + L_{xy}\cos\theta\sin\theta + L_{yx}\cos\theta\sin\theta + L_{yy}\sin^2\theta = \\ &= L_{xx}\cos^2\theta + 2L_{yx}\cos\theta\sin\theta + L_{yy}\sin^2\theta \end{aligned} \tag{3.17}$$

that is

$$L_{\theta^n} = \sum_{k=0}^{n} \binom{n}{k} L_{x^{n-k}y^k} cos^{n-k}\theta \sin^k \theta \tag{3.18}$$

where $\binom{n}{k}$ is the binomial coefficient. Illumination invariance can be easily achieved by using eq. 1.36. Moreover, the directional derivative of n-th order can be represented by a combination of the $n+1$ basis directional derivative $L_{\theta^n_{i,n}}$ [62], with $\theta_{i,n} = i\pi/(n+1) + \theta_g$ where $i = 0, \ldots, n$ and θ_g is an orientation related to the image structure. The following feature vector of length 12 can be used [62, 111]

$$\left[\frac{L_{\theta^2_{0,2}}}{L_{\theta_{0,2}}}, \ldots, \frac{L_{\theta^2_{2,2}}}{L_{\theta_{2,2}}}, \frac{L_{\theta^3_{0,3}}}{L_{\theta_{0,3}}}, \ldots, \frac{L_{\theta^3_{3,3}}}{L_{\theta_{3,3}}}, \frac{L_{\theta^4_{0,4}}}{L_{\theta_{0,4}}}, \ldots, \frac{L_{\theta^4_{4,4}}}{L_{\theta_{4,4}}} \right] \tag{3.19}$$

Differential invariants to rotation combining the local jet have been introduced in [83, 162]

$$\begin{bmatrix} L \\ L_xL_x + L_yL_y \\ L_{xx} + Lyy \\ L_{xx}L_xL_x + 2L_{xy}L_xL_y + L_{yy}L_yL_y \\ L_{xx}L_{xx} + 2L_{xy}L_{xy} + L_{yy}L_{yy} \\ L_{xxx}L_yL_yL_y + 3L_{xyy}L_xLxLy - 3L_{xxy}L_xL_yL_y - LyyyL_xL_xL_x \\ L_{xxx}L_xL_yL_y + L_{xxy}(-2L_xL_xL_y + L_yL_yL_y) + L_{xyy}(-2L_xL_yL_y + L_xL_xL_x) + L_{yyy}L_xL_xL_y \\ L_{xxy}(-L_xL_xL_x + 2L_xL_yL_y) + L_{xyy}(-2L_xL_xL_y + L_yL_yL_y) - L_{yyy}L_xL_yL_y + L_{xxx}L_xL_xL_y \\ L_{xxx}L_xL_xL_x + 3L_{xxy}L_xL_xL_y + 3L_{xyy}L_xL_yL_y + L_{yyy}L_yL_yL_y \end{bmatrix} \tag{3.20}$$

they can also be made invariant to affine illumination changes by eliminating the first two components and by dividing the other invariants by a proper power of the second component, i.e. the squared gradient magnitude.

A complex bank of filters, similar to the Gaussian derivatives is applied in [3, 146] instead

$$K(m,n) = (x+iy)^m(x-iy)^n g_\sigma(x,y) \tag{3.21}$$

the effect of a rotation by θ on the filter is a multiplication by $e^{i(m-n)\theta}$. For all the value $c = m - n$ the filters give the same response and for different values of c they are orthogonal, so an orthonormal filter bank is obtained. The filter bank differs from the Gaussian derivatives by a linear change of coordinates in the filter response space. The magnitude of response is not affected by the transformations, but only the phase. The authors obtain 16 filters by combining m and n which avoid the problems related to the estimation of a dominant orientation of a local feature.

3.3.4 The self-similarity descriptor

The self-similarity descriptor [151] has been used for template matching. The SSD between a small neighbourhood of a point $\mathbf{x}_c$ (tipically 5×5 pixels) and the surrounding region $\mathcal{R}$ is computed by obtaining the correlation surface

$$S_{\mathbf{x}_c}(\mathbf{x}) = e^{\frac{SSD(\mathbf{x},\mathbf{x}_c)}{\max(\sigma_{noise},\sigma_{auto}(\mathbf{x}))}} \tag{3.22}$$

where $\mathbf{x} \in \mathcal{R}$ and σ_{noise} is a constant that corresponds to acceptable photometric variations, while $\sigma_{auto}(\mathbf{x})$ takes into account the patch contrast and its pattern structure. The correlation surface is then mapped into a log-polar grid centered in $\mathbf{x}_c$ and the maximum value in each grid cell is taken to form the final descriptor vector (see fig. 3.9). Self-similarities are treated as local image properties and are accordingly measured locally. Moreover, the log-polar representation accounts for local affine deformations in the self-similarity descriptors while by choosing the maximal correlation value in each bin, the descriptor becomes insensitive to the exact position of the best matching patch within that bin [151]. The descriptor is computed densely on the whole image and through a modified version of the ensemble matching algorithm [151], which employs a probabilistic star graph model. This descriptor has been reported to provide good results in the template matching tasks.

3.3.5 The SURF descriptor

Similar to the SIFT, the SURF descriptor [4] uses a Cartesian grid but instead of computing a gradient orientation for each grid cell, the *Haar wavelet* [150] along the directions of the axes are computed. For each direction the sums of their values and of their absolute values are retained to obtain the descriptor.

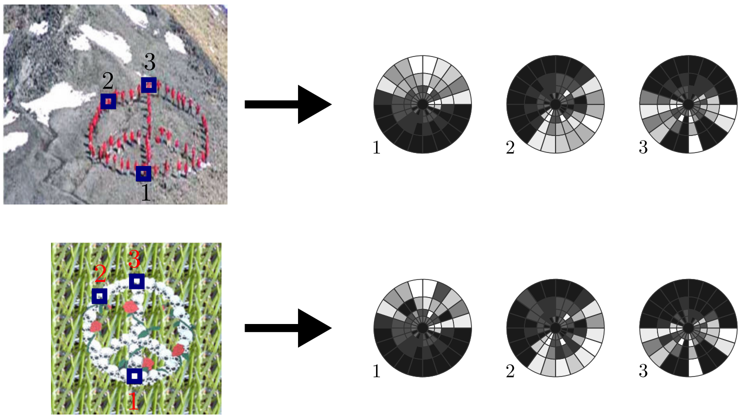


Figure 3.9: Two similar images (left) and the self similarity descriptors for the point highlighted by the blue squares (right). Image from [151]

Chapter 4

Feature similarity distances

4.1 The standard approaches

The choice of the right similarity/dissimilarity measure between two feature descriptor vectors is a well discussed topic. The *Minkowski distance* of order m

$$L_m(\mathcal{F}_1, \mathcal{F}_2) = \left(\sum_{i=1}^{n} |f_{1i} - f_{2i}|^m \right)^{1/m} \tag{4.1}$$

is the most common choice. In particular, the *Manhattan distance*, the *Euclidean distance* and the *Chessboard distance* respectively for $m = 1, 2, \infty$ are commonly used. Another common similarity measure derived by statistics is the *Chi squared distance*

$$\chi^2(\mathcal{F}_1, \mathcal{F}_2) = \sum_{i=1}^{n} \frac{2(f_{1i} - f_{2i})^2}{f_{1i} + f_{2i}} \tag{4.2}$$

which follows a chi squared distribution with $n - 1$ degrees of freedom under the assumption that the n vector elements are sampled from independent Gaussian variables [143, 111], i.e. they are not correlated.

The *cross correlation* and the *normalized cross correlation* are related to the independence of the two vector distributions [143, 33]

$$\begin{aligned} \mathcal{C}(\mathcal{F}_1, \mathcal{F}_2) &= \frac{1}{n} \sum_{i=1}^{n} \left((f_{1i} - \overline{\mathcal{F}}_1)(f_{2i} - \overline{\mathcal{F}}_2) \right) \\ \mathcal{C}^{\star}(\mathcal{F}_1, \mathcal{F}_1) &= \frac{\mathcal{C}(\mathcal{F}_1, \mathcal{F}_2)}{\sqrt{\mathcal{C}^2(\mathcal{F}_1, \mathcal{F}_1)\mathcal{C}^2(\mathcal{F}_2, \mathcal{F}_2)}} \end{aligned} \tag{4.3}$$

where $\overline{\mathcal{F}}_w = \frac{1}{k} \sum_{i=1}^{k} f_{wi}$. Further distances are the *symmetric Kullback-Leibler divergence* (also known as *Jeffrey divergence*)

$$\mathcal{J}(\mathcal{F}_1, \mathcal{F}_2) = \sum_{i=1}^{n} \left(f_{1i} \log \frac{f_{1i}}{f_{2i}} + f_{2i} \log \frac{f_{2i}}{f_{1i}} \right) \tag{4.4}$$

which measures how inefficient on average it would be to code one histogram by using the other one as the code-book and vice-versa [143, 33], and the *Bhattacharyya distance*

$$\mathcal{B}(\mathcal{F}_1, \mathcal{F}_2) = \sqrt{\sum_{i=1}^{n} f_{1i} f_{2i}} \tag{4.5}$$

that is an approximate measurement of the amount of overlap between two statistical samples [42] which is commonly used to compare histograms [73]. A last distance is the *intersection distance* between two histograms [143, 160]

$$\bigcap(\mathcal{F}_1, \mathcal{F}_2) = \sum_{i=1}^{n} \min(f_{1i}, f_{2i}) \tag{4.6}$$

The distance measures described above do not take into account the correlation between the different vector elements. In particular, the data provided by the descriptor are usually redundant, noisy and some descriptor elements are more discriminative than others. Moreover, dealing with histogram based descriptors, spatial relation between bins is not considered, especially for three-dimensional histograms as in the SIFT because the histogram is linearized into the descriptor vector. To be also noted the crucial role played by the bin size, since a coarse binning has no sufficient discriminative power, while a fine binning could be too discriminative [143].

4.2 Feature space dimension reduction

4.2.1 Mahalanobis distance and PCA reduction

Data analysis techniques are employed to reduce the correlation between the descriptor elements and to remove useless data, decreasing the descriptor dimension. The most common tools are the Mahalanobis distance and the PCA, which are very similar. In particular they both remove the correlation between descriptor elements while the latter also decreases the descriptor dimension, removing useless data. The covariance matrix Σ (see eq. 1.17) of the descriptor vector is learnt on a large database of feature patches. As seen in Sec. 1.3.4 the covariance matrix for the $m \times n$ descriptor matrix $\mathrm{X} = [\mathcal{F}_1 \ldots \mathcal{F}_m]^T$ is

$$\Sigma = (\mathrm{X} - \overline{\mathrm{X}})(\mathrm{X} - \overline{\mathrm{X}})^T \tag{4.7}$$

where $\overline{\mathrm{X}}_{ij} = \frac{1}{n}\sum_{i=1}^{n} \mathrm{X}_{ij}$ is the mean value of the j-th element. The covariance matrix is symmetric, thus it has positive eigenvalues and can be diagonalized

$$\Sigma = \mathrm{RDR}^T \tag{4.8}$$

where $\mathrm{D} = \mathrm{diag}(\lambda_1, \ldots, \lambda_n)$ is a diagonal matrix, R is a orthonormal matrix and $\lambda_1, \ldots, \lambda_n$ are the eigenvalues of Σ. If the features are normalized by the mean of each element in the covariance matrix, i.e. $\overline{\mathrm{X}} = \mathbf{0}$, the Mahalanobis distance

$$\mathcal{M}(\mathcal{F}_1, \mathcal{F}_2) = \mathcal{F}_2 \Sigma^{-1} \mathcal{F}_1 = \mathcal{F}_2 \mathrm{A}^T \mathrm{A} \mathcal{F}_1 = \widehat{\mathcal{F}}_2^T \widehat{\mathcal{F}}_1 \tag{4.9}$$

where $\mathrm{A} = \mathrm{D}^{-\frac{1}{2}} \mathrm{R}^T$ allows to use the Euclidean distance in an uncorrelated feature space inducted by the transformation $\widehat{\mathcal{F}} = \mathrm{A}\mathcal{F}$. Moreover, in the new space the data are uncorrelated since the correlation matrix becomes

$$\begin{aligned}\widehat{\Sigma} = \mathrm{A}\Sigma\mathrm{A}^T &= \mathrm{D}^{-\frac{1}{2}}\mathrm{R}^T\Sigma\mathrm{R}\mathrm{D}^{-\frac{1}{2}} = \\ &= \mathrm{D}^{-\frac{1}{2}}\mathrm{R}^T\mathrm{RDR}^T\mathrm{R}\mathrm{D}^{-\frac{1}{2}} = \mathbf{I}\end{aligned} \tag{4.10}$$

For the sake of efficiency, in practice the feature vectors are pre-multiplied by A. The PCA is an extension to the Mahalanobis distance which removes the elements with a low variance in a new feature space inducted by $\widehat{\mathcal{F}} = \mathrm{A}\mathcal{F}$, i.e. the less meaningful elements. The eigenvalues of D are ordered so that $\lambda_1 \geq \ldots, \geq \lambda_n$ and the dimensions $i, \ldots, n$ for which $\sum_{k=1}^{i} \lambda_i > th_\lambda$ are removed, replacing D by $\widehat{\mathrm{D}} = \mathrm{diag}(\lambda_1, \ldots, \lambda_i)$ and R by $\overline{\mathrm{R}} = [\mathbf{R}_1 \ldots \mathbf{R}_i]$, where $\mathbf{R}_k$ is the k-th column vector of R, i.e. the eigenvector corresponding to λ_k.

4.2.2 Linear embeddings

More complex data manipulations have been also proposed in recent years. The *linear discriminative embedding* [72] learns the best projection $\mathbf{w} \in \mathbb{R}^n$ which maximizes the ratio of the variance between the non-match and match differences along the direction $\mathbf{w}$.

$$\mathcal{Q}(\mathbf{w}) = \frac{\sum_{l_{ij}=1} \left(\mathbf{w}^T(\mathcal{F}_i - \mathcal{F}_j)\right)^2}{\sum_{l_{ij}=0} \left(\mathbf{w}^T(\mathcal{F}_i - \mathcal{F}_j)\right)^2} \tag{4.11}$$

where $l_{ij} = 1$ if the i-th and the j-th descriptor vectors belong to the same feature, i.e. the patches represent the same feature after a transformation, and $l_{ij} = 0$ otherwise. The expression 4.11 can be rewritten in terms of the covariance matrix

$$\mathcal{Q}^\star(\mathbf{w}) = \frac{\mathbf{w}^T \left(\sum_{l_{ij}=1} (\mathcal{F}_i - \mathcal{F}_j)(\mathcal{F}_i - \mathcal{F}_j)^T\right) \mathbf{w}}{\mathbf{w}^T \left(\sum_{l_{ij}=0} (\mathcal{F}_i - \mathcal{F}_j)(\mathcal{F}_i - \mathcal{F}_j)^T\right) \mathbf{w}} = \frac{\mathbf{w}^T \mathrm{A}\mathbf{w}}{\mathbf{w}^T \mathrm{B}\mathbf{w}} \tag{4.12}$$

The solution is provided by the eigenvector associated to the largest eigenvalue of the generalized eigensystem

$$\mathrm{A}\mathbf{w} = \lambda \mathrm{B}\mathbf{w} \tag{4.13}$$

To form a linear embedding, the first n eigenvectors associated with the largest n generalized eigenvalues are chosen. In order to provide a better projection space, some regularizations on the eigenvalues of the covariance matrices and the orthogonality constrain can be imposed [72]. Similar solution have been obtained starting from a different problem formulation in the LDE (Local Discriminant Embedding) [24]. Moreover, extensions to non linear transformations by using kernels have also been proposed [23], as well as the combinations of several transformations by averaging different models in the UFT (Universal Feature Transform) [23].

4.2.3 Learning methods

It has been shown in [179] that also the SVM (*support vector machine*) [32] can improve the distance. In particular a polynomial SVM kernel $\mathcal{D}$ of the form $\mathcal{D}(\mathcal{F}_1, \mathcal{F}_2) < th$, is learnt in the L_2 norm space by using correct feature correspondences as positive examples and wrong correspondences very close in the descriptor space as negative samples. The learned distance $\mathcal{D}$ shows a better a discriminative power.

Another approach by [185] learns on a large training dataset the best parameters which maximize the relative ROC (*receive operator characteristic*) curve [74] for different descriptor configurations.

Also the entropy maximization has been used to find the best patch normalization before computing the descriptor [180].

4.3 Cross bin distances

4.3.1 Quadratic form distance

In order to improve the histogram based descriptor similarity, several *cross-bin distances* has been proposed, which try to take into account the relative spatial position between bins. The *quadratic form distance* [125]

$$\mathcal{Q}(\mathcal{F}_1, \mathcal{F}_2) = \sqrt{(\mathcal{F}_1 - \mathcal{F}_2)^T \mathrm{A} (\mathcal{F}_1 - \mathcal{F}_2)} \tag{4.14}$$

improves the cross-bin information by using the matrix A where $\mathrm{A}_{ij} = 1 - d_{ij}/d_{max}$, d_{ij} is the distance between the bin centres and d_{max} the maximum distance, however its performances are not so good as expected [143].

4.3.2 The earth moving distance

A better choice is provided by the *Earth Moving Distance* (*EMD*) [143]. Intuitively, given two distributions, one seen as a mass of earth distributed in the space and the other as holes in the same space, the EMD is the minimum amount of work to fill the holes with earth.

The EMD solution is based on the well-known *transportation problem* [122]. Given a number of suppliers of a limited capacity, a number of consumers should be satisfied by giving a defined amount of goods stored in the suppliers. For each customer-supplier pair, a cost for transporting a single unit of goods is given. The transportation problem is solved by finding the least-expensive flow of goods from the suppliers to the consumers. The problem can be formalized as the following linear problem where the quantity

$$\sum_{i=1}^{m} \sum_{j=1}^{n} d_{ij} s_{ij} \tag{4.15}$$

must be minimized, subject to

$$\begin{array}{c} s_{ij} > 0 \qquad 1 \leq i \leq m, 1 \leq j \leq n \\ \sum_{j=1}^{n} s_{ij} \leq f_{1i} \qquad 1 \leq j \leq n \\ \sum_{i=1}^{m} s_{ij} \leq f_{2j} \qquad 1 \leq i \leq m \\ \sum_{i=1}^{m} \sum_{j=1}^{n} s_{ij} = \min \left(\sum_{i=1}^{m} f_{1i}, \sum_{j=1}^{n} f_{2j} \right) \end{array} \tag{4.16}$$

The value d_{ij} is the distance between the i-th and j-th bin centres and s_{ij} is the flux from bin i to bin j. The first constrain says that the flux cannot be negative, while the others impose a limit on the total flux from each bin, i.e. the goods cannot be more than those provided by the corresponding supplier or no more than those the customer can use. The final EMD distance is

$$\mathcal{E}(\mathcal{F}_1,\mathcal{F}_2)=\frac{\sum_{i=1}^{m}\sum_{j=1}^{n}d_{ij}s_{ij}}{\sum_{i=1}^{m}\sum_{j=1}^{n}s_{ij}} \tag{4.17}$$

Though the EMD distance provides good results, it is really slow to compute because it requires the solution of a minimization problem which can be resolved by the *simplex method* [122] or as a network flow minimization [31] in $\mathcal{O}(n^3\log n)$. Moreover it is a metric only if the histograms are normalized to the unit.

4.3.3 The earth moving distance approximations

In [131] an EMD based distance, named SIFT distance, is derived. A circular cross bin distance is used with a transportation cost equal to 1 for two nearby bins, while distant bins have a transportation cost equal to 2. The circularity of the cross bin takes into account the cyclic nature of the gradient orientations. The final SIFT distance is obtained by summing up the EMD based distances for each SIFT cell. The SIFT distance can be computed efficiently in linear time.

Faster approximations of the EMD have been developed [97, 187], some of them model the histogram difference by the diffusion process and are similar to the pyramidal matching kernel (see Sec. 1.4.2).

The heat diffusion equation for an isolated temperature field $T(x,t)$ with initial condition $T(x,0)$

$$\frac{\partial}{\partial t}T=\frac{\partial}{\partial x^2}T \tag{4.18}$$

has an unique solution

$$T(x,t)=g_t{*}T(x,0) \tag{4.19}$$

Since it is a conservative field, the mean distance is zero and as t increases $T(x,t)$ goes to zero. In this sense, $T(x,t)$ can be viewed as a process of exchanges between histogram bins until they become equal. A dissimilarity can be extracted by measuring the process diffusion. The diffusion process of the difference between the histograms through the time can be seen as successive small steps in the transportation problem to balance the same difference, where the system conservation equals the EMD constrains (see fig. 4.1).

The *diffusion distance* [97] is then given as the sum of different layers which mimics the histogram difference diffusion for increasing discrete time intervals

$$\mathcal{G}(\mathcal{F}_1,\mathcal{F}_2)=\sum_{i=0}^{m}g_i*\mathcal{D}(\mathcal{F}_1,\mathcal{F}_2) \tag{4.20}$$

where $\mathcal{D}$ is a distance, for instance the Manhattan $\mathcal{L}_1$, and m is the final state for which the difference is zero. In practice each layer of the diffusion distance can be obtained from the precedent by smoothing with a Gaussian g_σ with a constant standard deviation, and then subsampling, so that $m=\log n$.

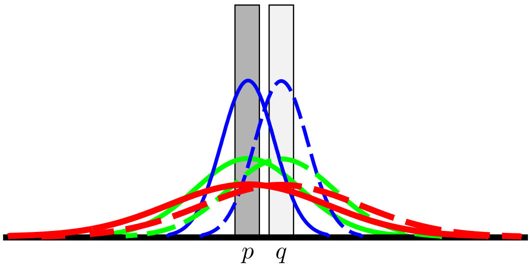


Figure 4.1: Given two bins p and q for which any standard approach would give a maximum distance though the bins are very close, successive diffusion steps (blue, green and red lines), makes the bin difference close to zero

In a similar way the topology preserved diffusion distance is derived [187]. In comparison with the diffusion distance, the topology of the bin distribution can be taken into account by using numerical methods to solve the diffusion equation.

4.4 Rank based distances

In the SIFT rank descriptor [163] the rank order of the descriptor vector is used, i.e. the feature vector $\mathcal{F}_k = [f_{k1}, \ldots, f_{kn}]^T$ is replaced by $\mathcal{F}_k^* = [f_{k1}^*, \ldots, f_{kn}^*]^T$ where f_{ki}^* is the rank of f_{ki}, i.e.

$$f_{ki}^* = |\{f_{kw} : f_{kw} \leq f_{ki}\}| \tag{4.21}$$

the *Spearman correlation coefficient* [157]

$$\rho(\mathcal{F}_1^*, \mathcal{F}_2^*) = 1 - \frac{6 \sum_{i=1}^n (f_{1i}^* - f_{2i}^*)^2}{n(n^2 - 1)} \tag{4.22}$$

or the *Kendall coefficient* [79]

$$\tau(\mathcal{F}_1^*, \mathcal{F}_2^*) = 1 - \frac{2 \sum_{i=1}^n \sum_{j=i+1}^n \operatorname{sign}(f_{1i}^* - f_{1j}^*) \operatorname{sign}(f_{2i}^* - f_{2j}^*)}{n(n-1)} \tag{4.23}$$

can be used to measure the similarity between the two rank order descriptors.

Chapter 5

Feature detector and descriptor evaluation

5.1 Evaluation methodologies

The comparison between feature descriptors and detectors is a difficult task, because it not easy to define a reliable quantitative measure to compare detectors or descriptors for all possible situations. Some detectors as well as some descriptors are more sensible to a class of images than others, or to a particular kind of transformations, thus the choice of the image dataset is critical. Moreover implementation details can also cause different results. Another issue is represented by the availability of a ground truth data, especially for three-dimensional scenes, which is more difficult to obtain. To overcome this issue synthetic generated images have been used sometimes [179].

5.1.1 The repeatability index

The *repeatability index* is commonly adopted the test the detectors [149]. Given a reference image, the repeatability index measures how well the features are repeated for some transformation of the image (see fig. 5.1). In order to define the repeatability index, the *overlap error* [149] between two feature patches should be introduced. Here the i-th feature is considered as the pair $p = (\mathbf{x}, \mathcal{R})$ where $\mathbf{x}$ is the feature centre and $\mathcal{R}$ is the shape of the patch, not normalized, i.e the associated ellipse. The overlap error between two features is

$$\mathcal{E}_o(p_1, p_2) = 1 - \frac{\mathcal{R}_1 \bigcap \mathcal{R}_2}{\mathcal{R}_1 \bigcup \mathcal{R}_2} \tag{5.1}$$

and it is $\mathcal{E}_o(p_1, p_2) = 0$ if the feature patches are the same, $\mathcal{E}_o(p_1, p_2) = 1$ if they do not overlap (see fig. 5.2). Let the reference image I_a contain the set of features $A = \{a_i\}$ and let the test image I_b contain the feature set $B = \{b_j\}$, with $i = 1, \ldots, n$ and $j = 1, \ldots, m$. The transformation $\mathcal{T}$ which maps points from I_a to I_b, where $\mathcal{T}(p)$ is the reprojection of the feature ellipse from I_a to I_b should also be known.

Given a dissimilarity score $\mathcal{D}(p, q)$ with $p \in A$ and $q \in B$, in order to obtain a hard match an ordered set $Q_{\mathcal{D}}(A, B) = \{q_1, \ldots, q_k\}$ of pairs (p, q), $p \in A$, $q \in B$, is built as follows. All possible pairs according to the dissimilarity value $\mathcal{D}(p, q)$ are ordered by their increasing values and starting by an empty set $Q_{\mathcal{D}}$, the first pair is inserted in the set $Q_{\mathcal{D}}$. All other pairs that share the same p or q are removed and

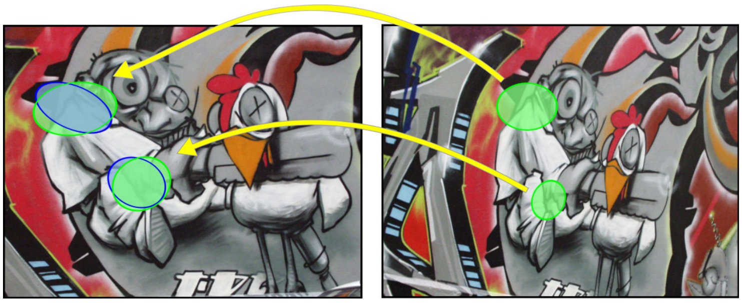


Figure 5.1: The feature patch is transferred from the right image to left and some error measure between corresponding patches, such as the repeatability index or the matching score, is computed

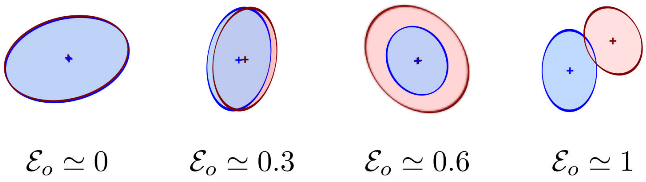


Figure 5.2: The overlap errors $\mathcal{E}_o$ for different patch superimposition. Image adapted from [114]

the process is repeated until there are no more pairs left. Clearly the cardinality of the set is $|Q_{\mathcal{D}}(A,B)| = \min(m,n)$

An hard match set $Q_{\mathcal{E}_o}$ is built for a given overlap error threshold th_o as well as the set $Q_{\mathcal{E}_o^{th_o}}$ which contains all the pair elements $(p,q) \in Q_{\mathcal{E}_o}$ for which $\mathcal{E}_o\left(p, \mathcal{T}^{-1}(q)\right) < th_o$. The repeatability index is the ratio between the cardinality of the two sets

$$rpt_{th_o}(A,B) = \frac{\left|Q_{\mathcal{E}_o^{th_o}}(A,B)\right|}{|Q_{\mathcal{E}_o}(A,B)|} \tag{5.2}$$

The repeatability index assesses the goodness of a detector in terms of stability and robustness of the extracted features.

However some considerations should be done. A detector which extracts a high number of features has a high probability to increase its repeatability score but it decreases the chances of a correct match due to more possible matches, so detectors should output the same reasonable number of features on average in the test. The location accuracy of the feature, measured from the patch centre, and of the overlap error, depend on the scale, i.e. on the ellipse size (see fig. 5.3), so it could be a good choice to normalize the patches to the same scale before computing the overlap error. It should be noted that while for some applications, for instance object detection, the localization error is less relevant, it is critical for others, such as three-dimensional reconstruction or registration.

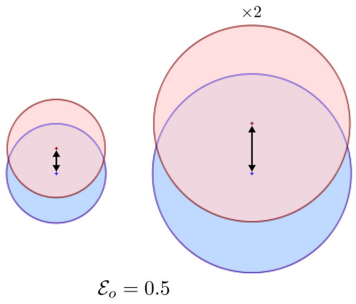


Figure 5.3: Overlapping patches (left) and the same patches with a double scale factor (right). Though both patches have the same overlap error $\mathcal{R}_o = 0.5$, the localization error of the left patch is doubled. Image from [68]

5.1.2 The matching score

Another measure is the *matching score* [114], where the set $Q_{\mathcal{E}_m}$ is built as done for the repeatability index, by using the dissimilarity measure obtained by applying the L_2 distance on pairs of SIFT descriptors. Given also the set $Q_{\mathcal{E}_m^{th_o}}$ which contains the pairs of $Q_{\mathcal{E}_m}$ with an overlap error less than th_o, the matching score is then defined by

$$match_{th_o}(A,B) = \frac{\left|Q_{\mathcal{E}_m^{th_o}}(A,B)\right|}{|Q_{\mathcal{E}_m}(A,B)|} \tag{5.3}$$

The use of the SIFT descriptor is nowadays standard due to its robustness and popularity.

5.1.3 Datasets

The most popular database which has become standard for detectors comparison is the Oxford dataset, available at [114, 115]. It contains six different image sequences, subject to different transformations. The images can be divided in *textured images*, i.e. with a large number of textures and repeated patterns, and *structured images*, i.e. with homogeneous and well defined regions. In particular the “graffiti” and “wall” scenes represent respectively structured and textured images under different perspective transformations, the “boat” and the “bark” sequences under scale and rotation, the “bikes” and the “trees” under blur, while the “Leuven” sequence contains a balance between structured and textured regions for luminance changes, as well as the “UBC” sequence for JPEG compression (see figs. 5.4–5.7). This database contains only planar scenes, so the ground truth is easy to estimate. The database lacks of three-dimensional sequences, which are crucial to compare detectors and descriptors for real applications.

The Oxford dataset has been extended to three-dimensional images in [59, 58]. Two sequences have been added, the “group” sequence contains a full three-dimensional scene with different planar surfaces, while the “room” sequence a more

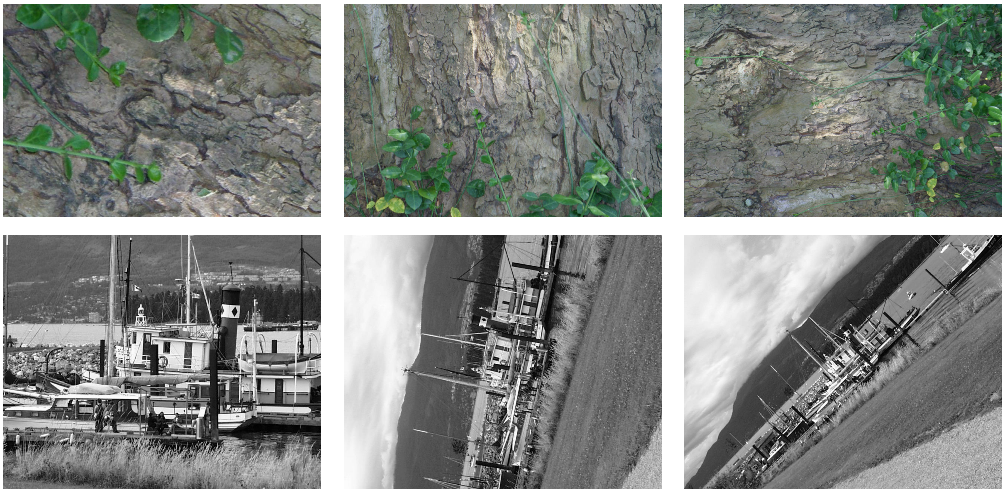


Figure 5.4: The bark sequence (top) and the boat sequence (bottom) for different scale and rotation degrees of transformation (left to right). Images from [115]

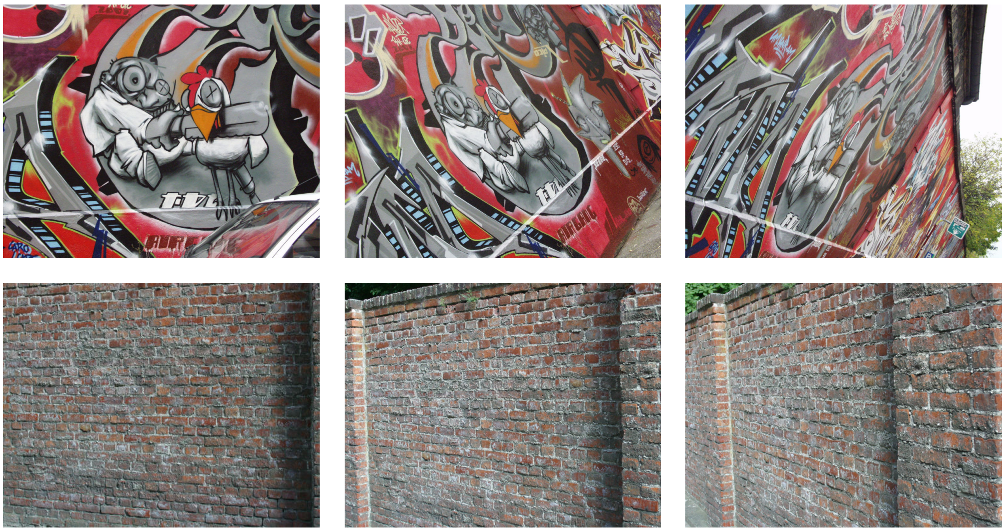


Figure 5.5: The graffiti sequence (top) and the wall sequence (bottom) for different viewpoints (left to right). Images from [115]

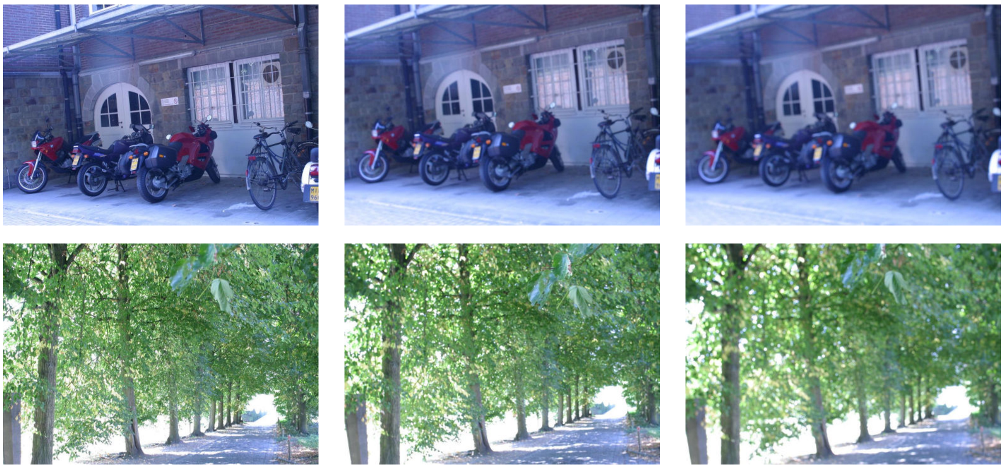


Figure 5.6: The bikes sequence (top) and the trees sequence (bottom) for different degrees of blur (left to right). Images from [115]

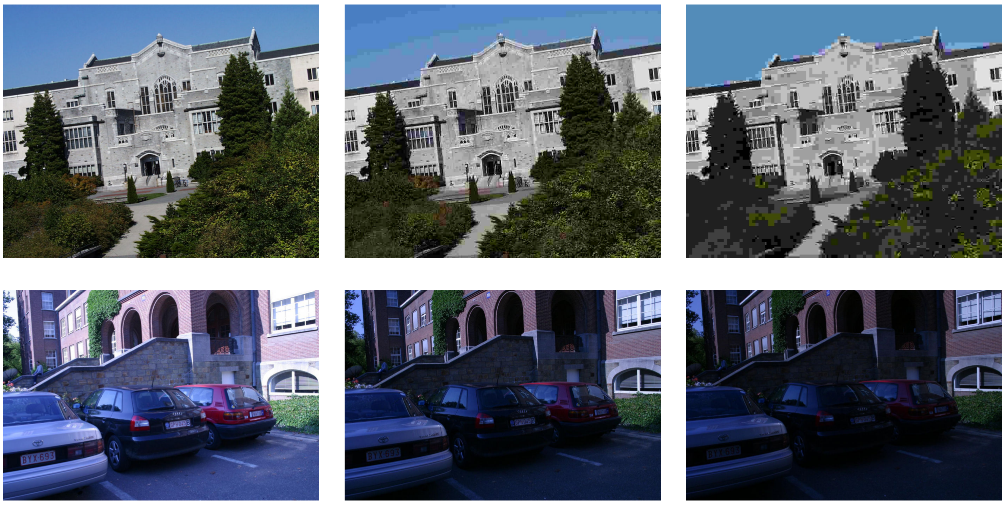


Figure 5.7: The Leuven sequence (top) and the UBC sequence (bottom), respectively for different JPEG compression factors and illuminations (left to right). Images from [115]

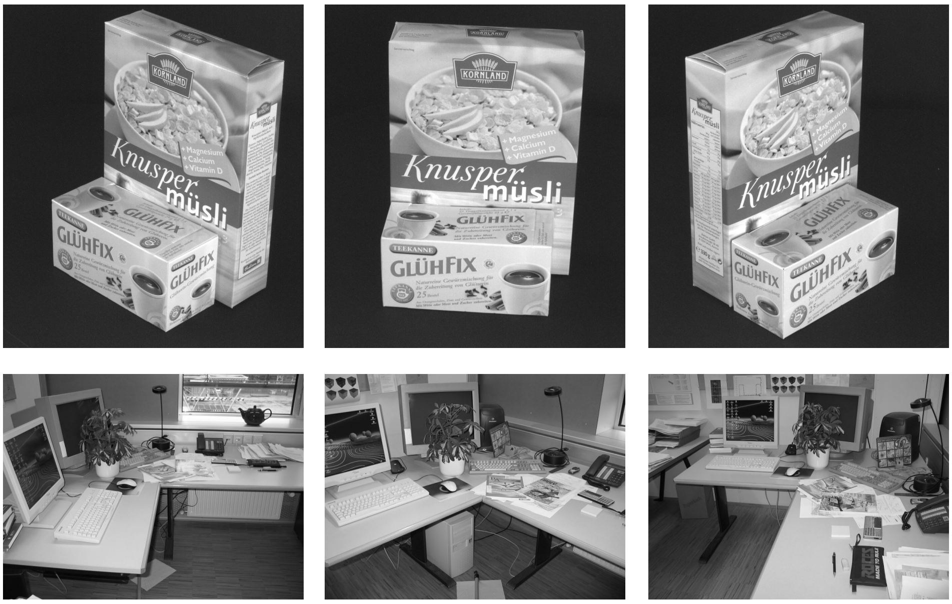


Figure 5.8: The group sequence (top) and the room sequence (bottom), for various viewpoint changes (left to right). Images from [58]

complex three-dimensional scene (see fig. 5.8). In order to generate the ground truth an intermediate image between two transformations is used. A dense disparity map between the reference image and the intermediate image is computed, which allows to find the location of the points. Then, by using the *trifocal tensor* [70], the location of a point can be finally found on the test image. Points on uniform regions for which the dense estimation could not be achieved are considered as a non-intersection for the computation of the overlap error. However detectors usually do not detect features in these regions, so this issue can be neglected.

5.1.4 Polar relationship

Another approach to compare detectors in three-dimensional scenes is provided in [54]. The *fundamental matrix* [70] (see Sec. 7.2.4 for more details) for the epipolar stereo geometry is first computed by taking correspondences by hand. Polar relationships are invariant to perspective transformations (the polar of a point to a curve is the straight line incident to the tangent point on the curve of the considered point, and perspective transformations preserve line incidence), which allow to use the polar relation of the epipole and the feature ellipse for the evaluation. In particular the tangent points $\mathbf{t}_1$, $\mathbf{t}_2$ of the ellipse p_1 through the epipole $\mathbf{e}$ in the reference image are projected in the test image to epipolar lines $\mathbf{l}_1$, $\mathbf{l}_2$. The epipolar lines intersect in $\mathbf{r}_1$ and $\mathbf{r}_2$ the line through the tangents $\mathbf{t}'_1$, $\mathbf{t}'_2$ of the epipole $\mathbf{e}'$ in the corresponding ellipse p_2 of the test image (see fig. 5.9). An overlapping error

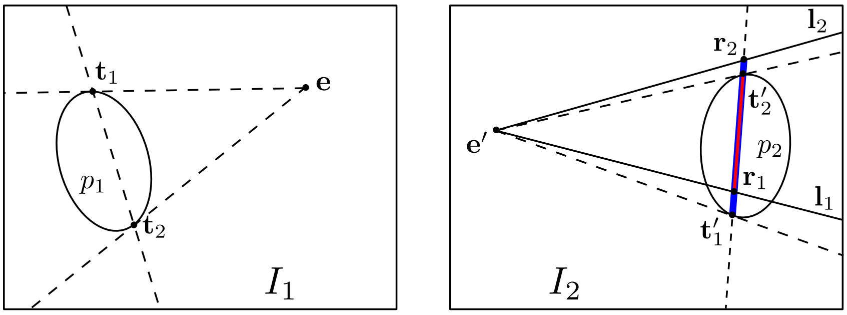


Figure 5.9: An ellipse patch p_1 on the first image I_1 (left) and the candidate match p_2 on the second image I_2 (right). In I_1 the polar line to the epipole $\mathbf{e}$ intersects the ellipse p_1 into the points t_1 and t_2 and, in similar way, in I_2 the polar line to the epipole $\mathbf{e}'$ intersects the ellipse p_2 into the points t'_1 and t'_2. The image of these point in I_2 through the fundamental matrix are the epipolar lines $\mathbf{l}_1$ and $\mathbf{l}_2$, which intersect the polar line into $\mathbf{r}_1$ and $\mathbf{r}_2$. The overlap error $\mathcal{E}'_o$ is computed as the ratio between the inner "smaller" red segment and the outer "wider" blue segment. Image adapted from [54]

between segments is computed as the ratio

$$\mathcal{E}'_o(p_1, p_2) = 1 - \frac{\max\left(0, \min(\mathbf{r}_h, \mathbf{t'}_h) - \max(\mathbf{r}_l, \mathbf{t'}_l)\right)}{\max(\mathbf{r}_h, \mathbf{t'}_h) - \min(\mathbf{r}_l, \mathbf{t'}_l)} \tag{5.4}$$

where $\mathbf{r}_l, \mathbf{r}_h$ are the lower and the higher point of $\{\mathbf{r}_1, \mathbf{r}_2\}$ as $\mathbf{t'}_l, \mathbf{t'}_h$ are for $\{\mathbf{t}'_1, \mathbf{t}'_2\}$, for a fixed direction on the incident line. This method does not need an intermediate image, so missed matches due to the presence of the feature point only in the reference image or only in the intermediate image are avoided, however it can generate some false matches, but according to the authors they are quite rare.

5.1.5 Epipolar relation using intermediate images

A last method which also makes use of an intermediate image was proposed [119] for the purpose of testing detectors and descriptors for object recognition. The authors build a database of images obtained by rotating a turntable and proceed as follows. A feature patch p is extracted from a reference image and paired with its nearest neighbour q according to the *nearest neighbour ratio* measure proposed by Lowe [100]

$$\frac{\mathcal{D}(p, q)}{\mathcal{D}(p, q')} \tag{5.5}$$

where $\mathcal{D}$ is a distance and q' is the second nearest neighbour of p. The pair is considered if the similarity between p and q is below a threshold, otherwise it is discarded and counted as "no match". If both the features are not related to the same object the pair counts as "false alarm", otherwise a further check is performed. The epipolar line corresponding to the point p is computed on the intermediate image and the features $t_1, \ldots, t_m$ in the intermediate image close to the epipolar line are scanned. If the intersection of the epipolar lines of p and one of $t_1, \ldots, t_m$ into the test image where there is q are close to q there is a "correct match" otherwise another "false alarm" is generated (see fig. 5.10). If no point from $t_1, \ldots, t_m$ was present, the pair does not contribute to any statistics, because the inability to establish a triple match is not caused by a poor performance of the detector on the target image. ROC curves [74] of the "false alarms" with respect to the "correct matches" are obtained by varying the similarity threshold. This methods is fully geometrical and does not require any further information as the feature shape. Some matching errors can be present due to the estimation of the fundamental matrix and when incident epipolar line are almost parallel, but the author verified statistically that the error is below 0.05%.

Aside from the tests depicted above, other comparisons to asses the complementarity between different detectors have been done [59, 41] as well as some tests to measure the informative content extracted by the detectors.

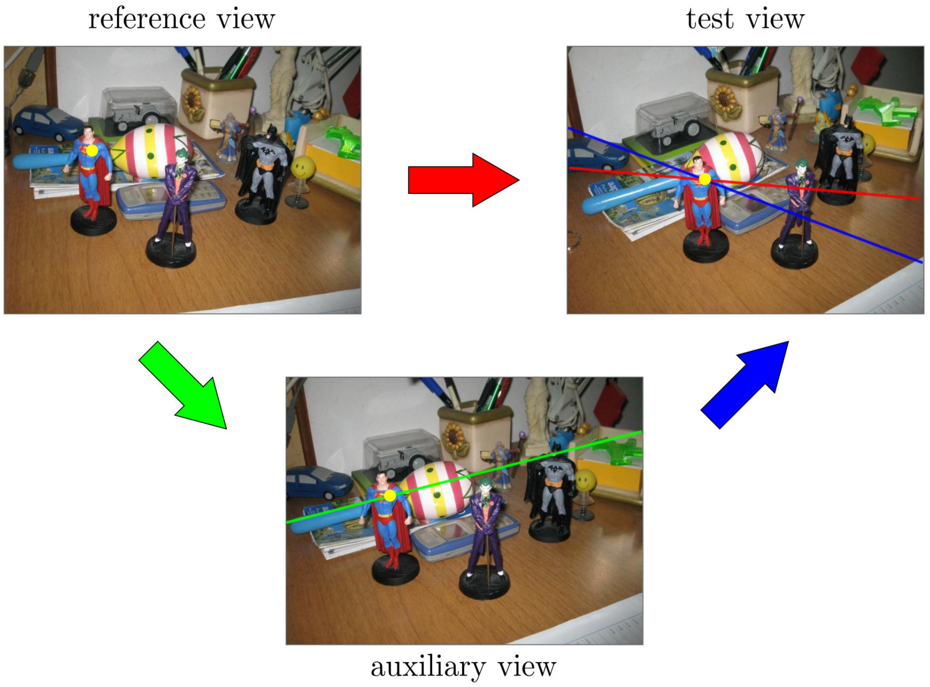


Figure 5.10: A feature point (yellow dot) in the reference view is mapped into the epipolar lines in the auxiliary and the test image using the respective fundamental matrices (green and red line). If in the auxiliary image a feature point close to the epipolar line exists (yellow dot) it is also mapped into an epipolar line in the test image (blue line). If the candidate match (yellow dot) in the test image is close to the intersection of the two epipolar line then the match is accepted

5.2 Detector comparison

To conclude the feature detector evaluation, a resume of the results from the cited comparisons is given. Non affine-covariant detectors perform well for scene were there is not a high degree of perspective distortion, especially the SIFT detector, and sometimes can outperform their affine covariant counterparts [59]. The MSER detector has good performances in particular on planar and structured scene, usually followed by the Hessian-affine and Harris-affine detectors [114]. However for fully three-dimensional scenes the performances on any detector degrade noticeably [119]. Corner based detectors seem to be more stable to illumination changes and in junction localization, instead of blob detectors which are however more stable to strong perspective distortion, because uniform regions are less like to be strongly modified by a geometric distortion [177]. The localization error depends linearly on the region size, with the exception of the MSER, so a filtering on the feature size can improve the matching process [68]. Moreover a careful implementation, as well as the choice of detector parameters, is also relevant for the detector performances [140]. The choice of the best detector is related to the input images and to the task. A combination of detectors based on different approaches can increase the overall performances [41].

Also to be mentioned, a theoretical comparison between autocorrelation based cornerness functions [80]. According to their assumption the authors show that the Shi and Tomasi function (see eq. 2.9) has the best theoretical properties followed by the Förstner function (see eq. 2.7), while the Harris and Stephens function (see eq. 2.6) seems to be the worst. However, their assumption are a bit constrained, for instance they assume that the pixel intensity variation between transformations is constant and not at least linear, and they restrict the comparison to affine transformations only. Moreover, some properties presented, valid for multidimensional or multispectral images, are unneeded in the usual image definition and practical applications have shown the validity of all the cornerness function described.

5.3 Descriptor evaluation framework

The most used framework to evaluate feature descriptors [113] is also based on the Oxford dataset. The features are extracted by a detectors (the Hessian-affine in the original paper) and matched (the L_2 metric is commonly used as distance $\mathcal{D}$). Three different approaches are presented to match features. The threshold matching associates features if

$$\mathcal{D}(\mathcal{F}_1, \mathcal{F}_2) < th \tag{5.6}$$

The other two criteria are the nearest neighbour according to the rank, i.e. the first $th_{nn} \in \mathbb{N}$ elements of the ordered set $Q_{\mathcal{D}}$ (see eq. 5.2) are considered, and the nearest neighbour ratio, i.e. the elements of $Q_{\mathcal{D}}$ are re-scored according to the ratio in eq. 5.5 and a threshold $th_{nnr} \in [0, 1]$ is applied.

The descriptors are evaluated according to the *1-precision/recall* curve obtained by increasing the corresponding threshold value for a given overlap error. The recall is defined as the number of correct matches over all possible correspondences for the given overlap error, while the "1-precision" is the number of the correct matches over all matched pairs. A perfect descriptor would give a recall equal to 1 for any precision, but in practice, the recall increases by increasing the threshold as the noise introduced by the image transformations increases the distance between similar descriptors. Horizontal curves indicate that the recall is attained with a high precision while a slowly increasing curve shows that the descriptor is affected by image degradation [113].

5.4 Descriptor comparison

Different comparisons have been done [113, 37, 163, 185, 121]. However, though the evaluation frameworks adopted are similar, there are some differences in the choice of some parameters. The histograms-based descriptors usually give the best results [113, 185], in particular the SIFT-based descriptors. Steerable filters are however a good choice when a low dimensionality is needed [113, 185]. The EMD distance usually gives the best results, followed by its approximations [97, 131, 187], however the performance gain with respect to the use of the Minkowsky norms is not so high. Dimensionality reduction methods such as the LDE are more effective than the PCA to drastically reduce the dimension of the SIFT descriptor while maintaining a high discriminative power [72].

Chapter 6

Improvements to feature detection and description

6.1 Overview

In the following a new feature detector based on the Harris corner (see Sec. 2.2.2) and an improvement to the SIFT descriptor (see Sec. 3.2.6) are presented.

The new detector, called HarrisZ [10, 9], allows a fine and stable corner selection without tuning the method. It achieves good results comparable with other state of the art detectors on the Oxford dataset and its three dimensional extension (see Sec. 5.1.3).

An extension of the GLOH descriptor (see Sec. 3.2.6), which improves the robustness to rotations is also presented. The proposed descriptor, called sGLOH [8], has also been compared on the Oxford image dataset (see Sec. 5.3), with good results which point out its stability.

6.2 The HarrisZ detector

6.2.1 Motivations and basic assumptions

As described in Sec. 2.2.2, the coefficient κ controls the corner response function H (see eq. 2.6). In particular, the sensitivity of H is reduced when κ is increased. Moreover, a point is selected as a corner if its response to H is greater than th_H. It should also be noted that both th_H and κ rely upon local properties of the input image, such as luminosity, noise or the intrinsic structure of the image (e.g. textured and non-textured image regions).

In data analysis, the mean can be adopted as a reference point to compare inhomogeneous data, while the standard deviation can be used as measure unit. Therefore, a good normalization choice is given by the z-score [74] function

$$Z(x) = \frac{x - \overline{x}}{\sigma} \tag{6.1}$$

where $\overline{x}$ and σ are respectively the mean value and the standard deviation. If two different quantities x_1 and x_2, with their respective mean values $\overline{x}_1$, $\overline{x}_2$ and standard deviations σ_1, σ_2, are associated to the same conditions, then x_1 and x_2 can be compared after a z-score normalization.

These argumentations provide the basic assumptions for the new detector. Indeed, both the average values of the determinant and of the trace of the autocorrelation matrix of the gradient magnitude map can be associated to flat regions, usually considered as background. Moreover, corners are usually near edges, or where a strong intensity variation is noticeable. The mean gradient magnitude itself provides a rough separation between flat regions and edges.

The proposed algorithm starts by computing an edge mask, used to enhance the derivatives according to the edges. Next the corner strength is measured by a function H_z, based on H, using the z-score normalization. According to H_z, corners are points $\mathbf{x}$ for which $H_z(\mathbf{x})>0$. Selected points which attain to a local maximum over H_z and are close to edges, according to the edge mask, are finally selected as corners. All these steps are repeated for different scales.

6.2.2 Algorithm description

Let $L_x(\mathbf{x},\sigma)$, $L_y(\mathbf{x},\sigma)$ be the scale-normalized derivatives (see eq. 1.14) and let $G(\mathbf{x},\sigma) = \sqrt{L_x^2(\mathbf{x},\sigma)+L_y^2(\mathbf{x},\sigma)}$ be the gradient magnitude. A simple threshold mask computed on the whole image through the mean value of the gradient $\overline{G}$, is used to separate flat regions and edges

$$K_{\sigma_D}(\mathbf{x}) = \begin{cases} 0 & \text{if} \quad G(\mathbf{x},\sigma_D)\leq\overline{G} \\ 1 & \text{otherwise} \end{cases} \tag{6.2}$$

where σ_D is the differentiation scale (see Sec. 1.3.3). By assuming that discontinuities have the same order of the current image resolution given by σ_D, the mask K is convolved with the Gaussian kernel g_{σ_D} which smooths strong discontinuities (see fig. 6.1(a-d))

$$M(\mathbf{x},\sigma_D) = g_{\sigma_D}*K(\mathbf{x},\sigma_D) \tag{6.3}$$

The initial derivatives are enhanced by a pixel-wise multiplication with the edge mask M (see fig. 6.1(e-h))

$$\begin{aligned} \overline{L}_x(\mathbf{p},\sigma_D) &= M(\mathbf{p},\sigma_D)\, L_x(\mathbf{p},\sigma_D) \\ \overline{L}_y(\mathbf{p},\sigma_D) &= M(\mathbf{p},\sigma_D)\, L_y(\mathbf{p},\sigma_D) \end{aligned} \tag{6.4}$$

The resulting derivatives $\overline{L}_x(\mathbf{p},\sigma_D)$ and $\overline{L}_y(\mathbf{p},\sigma_D)$ are used in place of $L_x(\mathbf{p},\sigma_D)$ and $L_y(\mathbf{p},\sigma_D)$ in the autocorrelation matrix $\mu(\mathbf{p},\sigma_I,\sigma_D)$ (eq. 1.23), to suppress the current scale noise.

Similar to the function H (see eq. 2.6), the cornerness function H_z is defined in a in a local neighbourhood determined by the integration scale σ_I as

$$H_z(\mathbf{x},\sigma_I,\sigma_D) = Z\left(\det\left(\mu(\mathbf{x},\sigma_I,\sigma_D)\right)\right) - Z\left(\operatorname{trace}^2\left(\mu(\mathbf{x},\sigma_I,\sigma_D)\right)\right) \tag{6.5}$$

where the autocorrelation matrix is computed according to eq. 1.23 using the improved derivatives $\overline{L}_x$ and $\overline{L}_y$ while the mean and the standard deviation in the z-score (see eq. 6.1) are computed on the whole image. The function H_z allows the comparison of the determinant and the squared trace of the autocorrelation matrix without the usual coefficient κ. Moreover, the mean values of both the determinant and the trace, which are equal to zero after the z-score normalization, can be associated to flat regions. Negative and positive values of H_z are shown in fig. 6.2(a,b) as dark and bright regions respectively.

Recalling that the determinant of the autocorrelation matrix is sensitive to corners, while the trace is sensitive to both edges and corners, it follows that

- $H_z \gg 0$, when the corner response is greater than the edge response;
- $H_z \ll 0$, when the edge response is greater than the corner response.

A candidate corner $\mathbf{x}$ is selected if it attains to a local maximum for H_z greater than zero, within a circular window with the same radius of the Gaussian kernel g_{σ_D}, and if $\mathbf{x}$ is close to an edge, that is when $M(\mathbf{x}, \sigma_D) > th_m$, for a proper threshold value th_m (see fig. 6.2(c,d)). A general lower bound for the threshold $th_m = 0.31$ which does not depend on the edge position neither on the scale is derived in the next section.

The whole method is repeated for different scales σ_I and the affine invariance is obtained by considering the ellipse associated to the autocorrelation matrix μ as described in Sec. 1.3.4 (see fig. 6.2(e,f). It should be noted that no image pyramid approach [20, 35] has been used, but increasing kernels are instead employed as done by the SURF detector (see Sec. 2.3.3), though the former approach speeds-up the process, because it also decreases the corner localization accuracy. Moreover the automatic scale selection (see Sec. 1.3.3) is not performed and, to better distribute features along the image, an approach similar to the adaptive non-maximal suppression (see Sec. 2.2.5) is used, which models the local maxima window according to the differentiation scale σ_D.

6.2.3 Threshold derivation

The following general notes must be considered for the choice of the threshold value th_m. Firstly, it must be noted that the values of M range in $[0, 1]$ (see eq. 6.3), since they have been obtained by the Gaussian convolution of the binary mask K (see eq. 6.2). The differentiation scale σ_D can be assumed to be the unit length of a pixel at the current observation scale, thus its size is σ_D.

As in fig. 6.3, let consider a long enough ideal step edge $f(x)$ in $x = 0$, that is the pixel on the ramp lies in $[-0.5\sigma_D, 0.5\sigma_D]$. After the convolution of f with g_{σ_D}, the initial value of the pixel can be restored with a cut-off at $x = -0.5\sigma_D$, that corresponds to $th_m \approx 0.31$

$$\begin{aligned} th_m &= g_{\sigma_D} * f(-0.5\sigma_D) = \int_{-\infty}^{\infty} f(x)\, g_{\sigma_D}(x+0.5\sigma_D)dx = \\ &= \int_{0}^{\infty} g_{\sigma_D}(x+0.5\sigma_D)dx = \int_{0}^{0.5\sigma_D} g_{\sigma_D}(x)dx = \\ &= \Psi_{0,\sigma_D}(0.5\sigma_D) = \Psi_{0,1}(0.5) \approx 0.31 \end{aligned} \tag{6.6}$$

where $\Psi_{\eta,\sigma}(x)$ is the normal cumulative distribution with mean η and standard deviation σ. Here, the assumption that the pixel is centred in $x = 0$ was done so

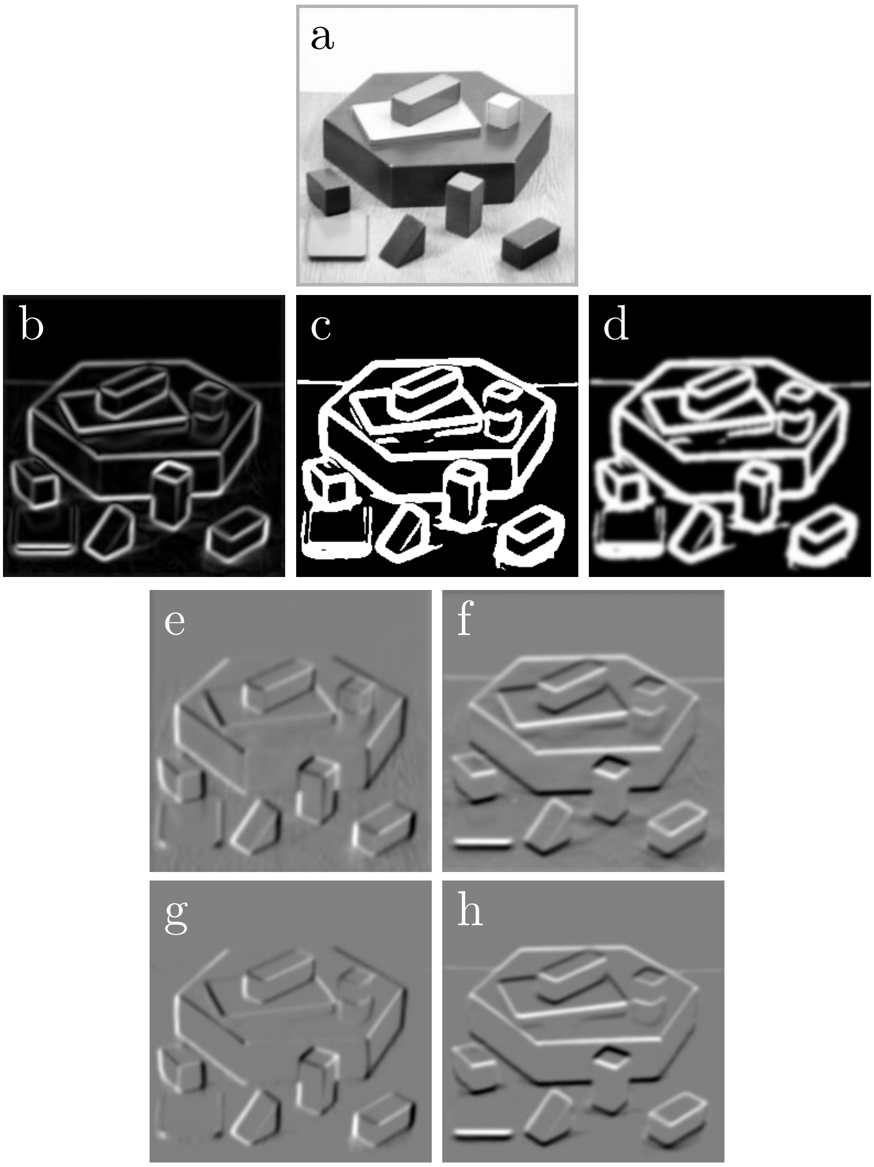


Figure 6.1: Original image (a), gradient magnitude (b), the binary mask K (c) and the final mask M (d). The original derivatives L_x, L_y (e,f) and the final enhanced derivatives $\overline{L_x}$, $\overline{L_y}$ (g,h). The results are obtained for $i = 3$, $\sigma_{I_i} = 1.4^i$ and $\sigma_{D_i} = 0.7\sigma_{I_i}$

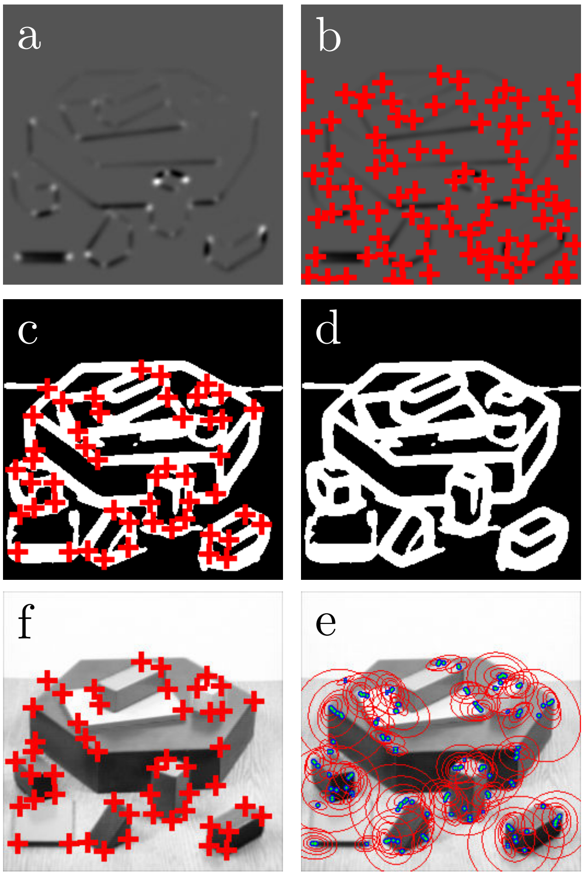


Figure 6.2: The H_z cornerness function (a) with superimposed local maxima (b) and the edge mask $M(\mathbf{x}, \sigma_D) > th_m$ (c) with final selected corners (d). Corners on the image at the current scale (e) and patch ellipses obtained at multiple scales (f). The current scale used is determined by $i = 3$, $\sigma_{I_i} = 1.4^i$ and $\sigma_{D_i} = 0.7\sigma_{I_i}$, while for (f) $i = 3, \ldots, 8$

that the left border of the pixel is in $x = -0.5\sigma_D$. A threshold $th_m = 0.31$ guarantees a general lower bound which does not depend on the position of the step edge neither on σ_D, and also includes the case $th_m = 0.5$, which corresponds to a pixel border in $x = 0$.

6.2.4 Implementation details

The observation scale depends on the detail level required by the image processing task and should be provided by the user. The scales have been set according to $\sigma_{I_i} = \xi^i \sigma_{I_0}$ and $\sigma_{D_i} = s\sigma_{I_i}$, with $\xi = 1.4$, $\sigma_{I_0} = 1$ and $s = 0.7$, as reported in [112]. Coarser scales are obtained by increasing the index i.

Table 6.1 shows the average computational time required to process a single scale in the case of i from 1 to 11 and the respective cumulative time, together with the

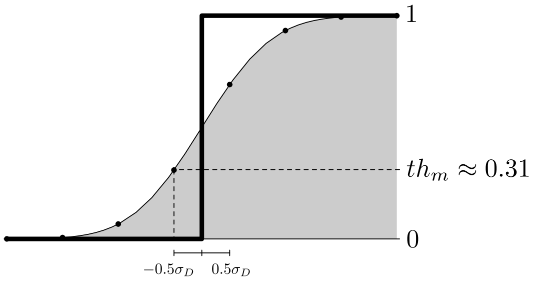


Figure 6.3: An ideal step edge (bold line) and its smoothed version (thin line). If σ_D is used as pixel resolution, a cutoff point is obtained for $th_m \approx 0.31$

percentage of corners extracted with respect to $i = 1$. The tests have been performed on the Oxford image dataset [114]. The algorithm has been implemented on a Linux system with kernel version 2.6.27, running on Intel® Core™2 Duo E8500 CPUs. As the scale increases, the computational time almost doubles and the number of extracted corners halves.

	scale index	corners (%)	time (mm:ss)		cum.time (mm:ss)		
			mean	max	mean	max	
	1	100.0	0:02	0:03	0:03	0:06	
to coarser scales	2	49.3	0:02	0:03	0:05	0:09	
	3	31.5	0:02	0:03	0:08	0:12	best scales
	4	17.1	0:03	0:04	0:10	0:15	
	5	9.7	0:03	0:04	0:13	0:19	
	6	5.3	0:04	0:06	0:17	0:24	
	7	2.9	0:05	0:07	0:22	0:31	
	8	1.6	0:07	0:08	0:28	0:38	
	9	0.8	0:09	0:12	0:37	0:50	
	10	0.4	0:13	0:16	0:50	1:07	
	11	0.3	0:19	0:24	1:09	1:30	

Table 6.1: The average computational time required to process a single scale in the case of i from 1 to 11 and the respective cumulative time, together with the percentage of extracted corners with respect to $i = 1$

To further refine the corner selection, points with $\sqrt{\lambda_2/\lambda_1} < 0.25$ have been discarded. Only 1.9% on average of the initially extracted corners have been removed, underlining the consistency of the new method.

The floating-point computation of the maximum value of H_z (see eq. 6.5) within a circular window of radius $3\sigma_D$ is particularly time consuming for high values of the index i. To speed-up the algorithm, a heap structure [31] was implemented which updates only border elements through auxiliary indexes when the window is shifted (see fig. 6.4).

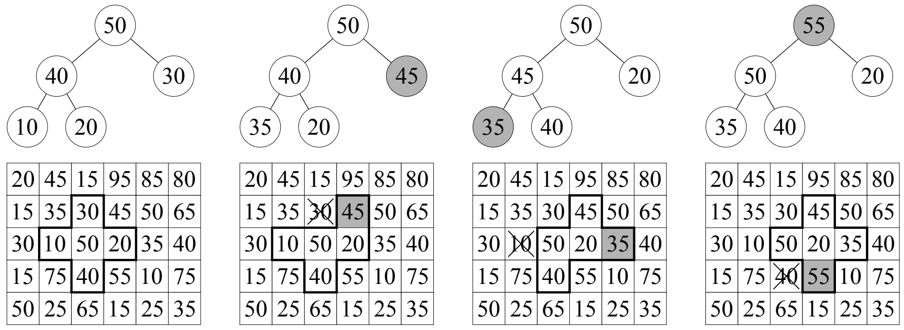


Figure 6.4: When the max filter window is shifted (bottom row) only border elements are updated in the corresponding heap (top row)

The whole absolute time of the detector is quite high compared to other covariant feature detectors [114, 4, 100], but it is still suitable for off-line tasks, moreover a parallel implementation has also been developed through a dynamic scheduler to balance the workload which was shown to perform better with respect to a static scheduler [6]. In any case, it should be noted that no approximations, which usually improve the computational performances, have been made.

Observing the small amount of features corresponding to more time consuming coarser scales, a reasonable compromise between the number of extracted corners and the computational performances (see table 6.1, columns “corners” and “time”) is provided for scale indexes $i < 9$.

6.2.5 Experimental results

The new detector was tested according to the repeatability index (see eq. 5.2) and the matching score (see eq. 5.3) on the Oxford dataset and its extension to three dimensional objects using the setup described in [114]. The SIFT and the recent SFOP detectors (see Sec. 2.2.3) have also been included in the comparison, using the code freely distributed by the authors, with their default parameters. To fairly compare about the same number of points detected for indexes $i = 3, \ldots, 8$, a subset of points, called SIFT*, was considered, obtained by the SIFT with scales $\sigma_I > 1.68$ (see table 6.2).

Results are reported in figs. 6.5–6.10, according to the repeatability index and the matching score, referred not only to the proposed algorithm, but obtained also through the Harris-affine detector [112] (HA in the plots), the SFOP and the SIFT methodologies. The absolute number of correspondences and matches are also reported. Tests reported in [114, 59], also in the case of further detectors, show that our approach is better then the Harris-affine methodology and comparable with the Hessian-affine and MSER detectors. We have experimentally verified that the new algorithm is also comparable with the SFOP and the SIFT approaches, though the

	scale index range				
	1-11	2-8	2-11	3-8	3-11
HA	378%	186%	189%	106%	107%
SFOP	336%	177%	179%	97%	100%
SIFT*	366%	188%	190%	103%	105%
SIFT	99%	55%	56%	34%	35%

Table 6.2: Percentage of extracted corners of the HarrisZ detector for different scale ranges with respect to other corner detectors. HA refers to the Harris-affine detector, while SIFT* is obtained for SIFT with $\sigma_I > 1.68$

latter returns a slightly better matching score in the cases of the "bark" and of the "boat" sequences, which correspond to scale and rotation changes.

The discrimination between features and noise at finer scales is a difficult task, indeed the feature stability increases with coarser scales. This conclusion can also be inferred from fig. 6.11 which shows the average repeatability index with respect to the overlap error on the Oxford database.

Furthermore, for high scale indexes, the results of the HarrisZ detector are almost unchanged, because only a negligible number of corners is extracted and their computation can be avoided, reducing the running time. In order to compare different detectors, both the repeatability index and the matching score require about the same number of extracted points. Actually, our algorithm with $i \geq 3$, the Harris-affine procedure and the SFOP and SIFT* detectors returned a comparable number of points.

Further discussions should be carried out for the "graffiti" sequence, where the new algorithm behaves like the Harris-affine detector for high viewpoint angles. Projective transformations are assumed piecewise locally affine, in the case of small projective distortions. Moreover, since corners rely on contour stability unlike blob-like features, which are based on homogeneous flat regions, corner detectors are more sensitive to relevant viewpoint changes (see Sec. 5.2). Besides, the HarrisZ detector returns good results in the case of non-planar scenes, comparable with those obtained by the MSER detector, thus underlining the stability and validity of our approach (see figs. 6.9–6.10).

6.2.6 Final remarks

As reported in Sec. 6.2.5, tests show the validity of the proposed methodology that returns good results when compared to other recent feature detectors.

In terms of repeatability index and matching score, the new HarrisZ detector provides better results, with respect to the standard Harris-affine detector, which are comparable with respect to the Hessian-affine and MSER approaches under all

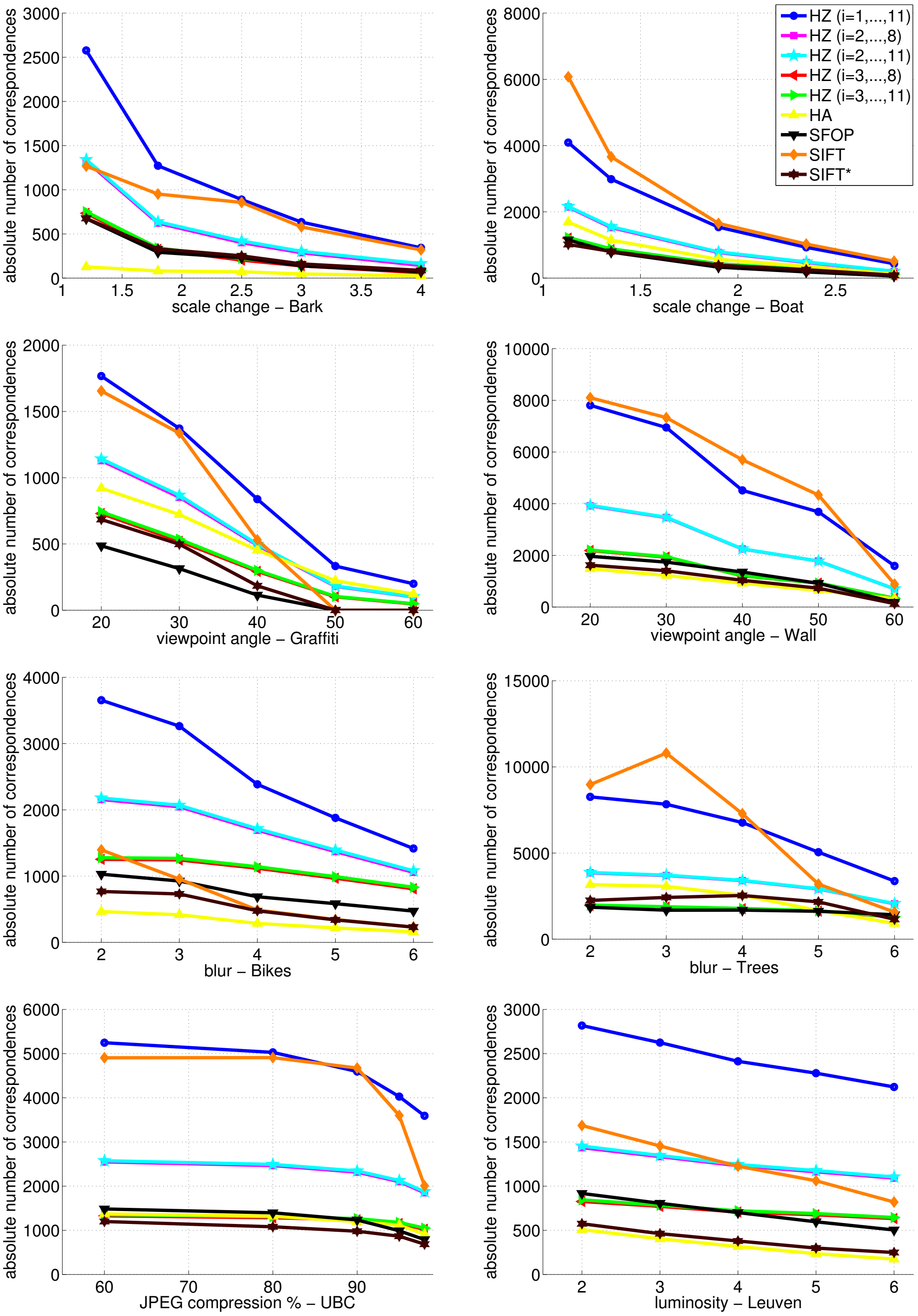


Figure 6.5: Absolute number of correspondences for the Oxford dataset

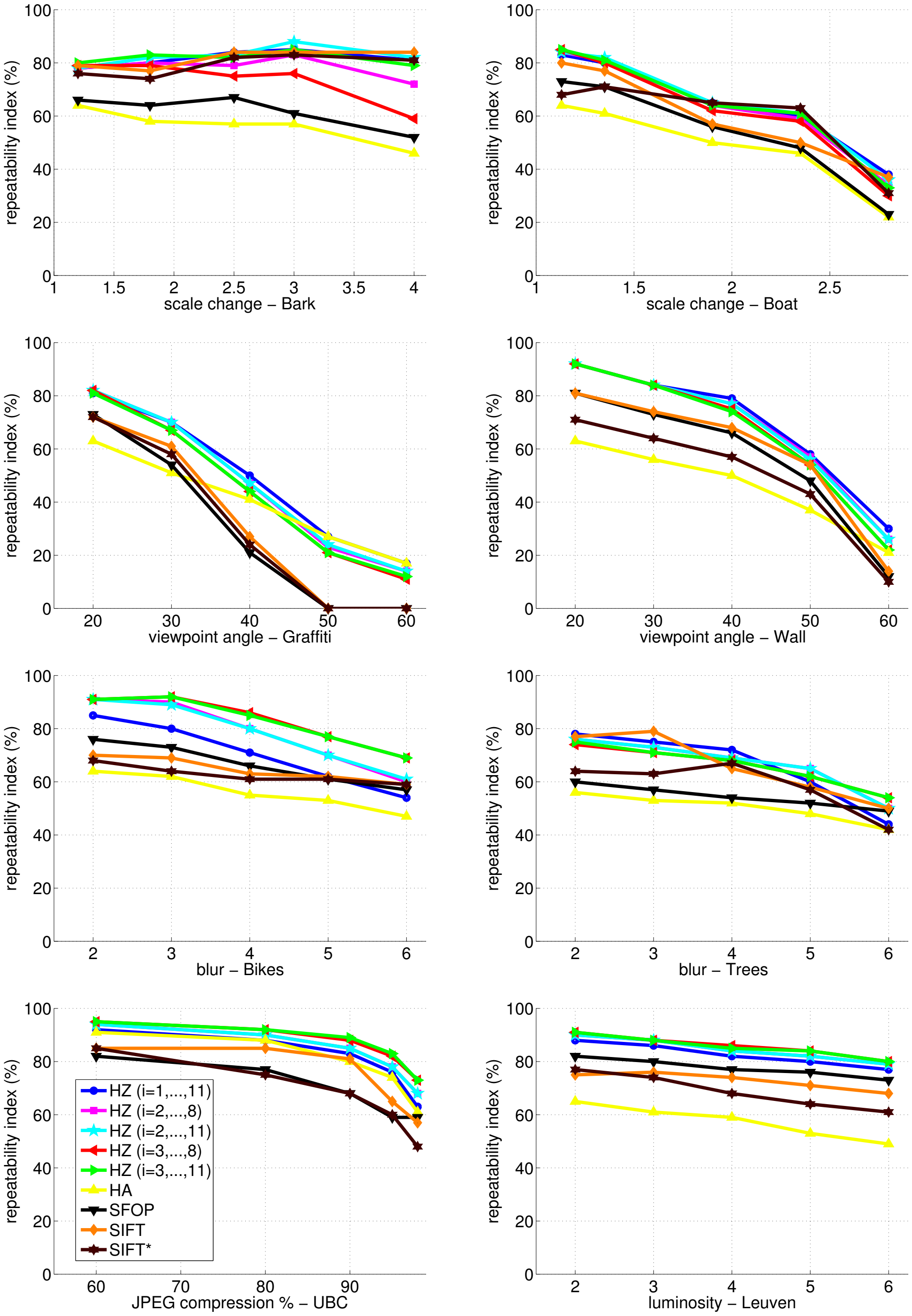


Figure 6.6: Repeatability index for the Oxford dataset

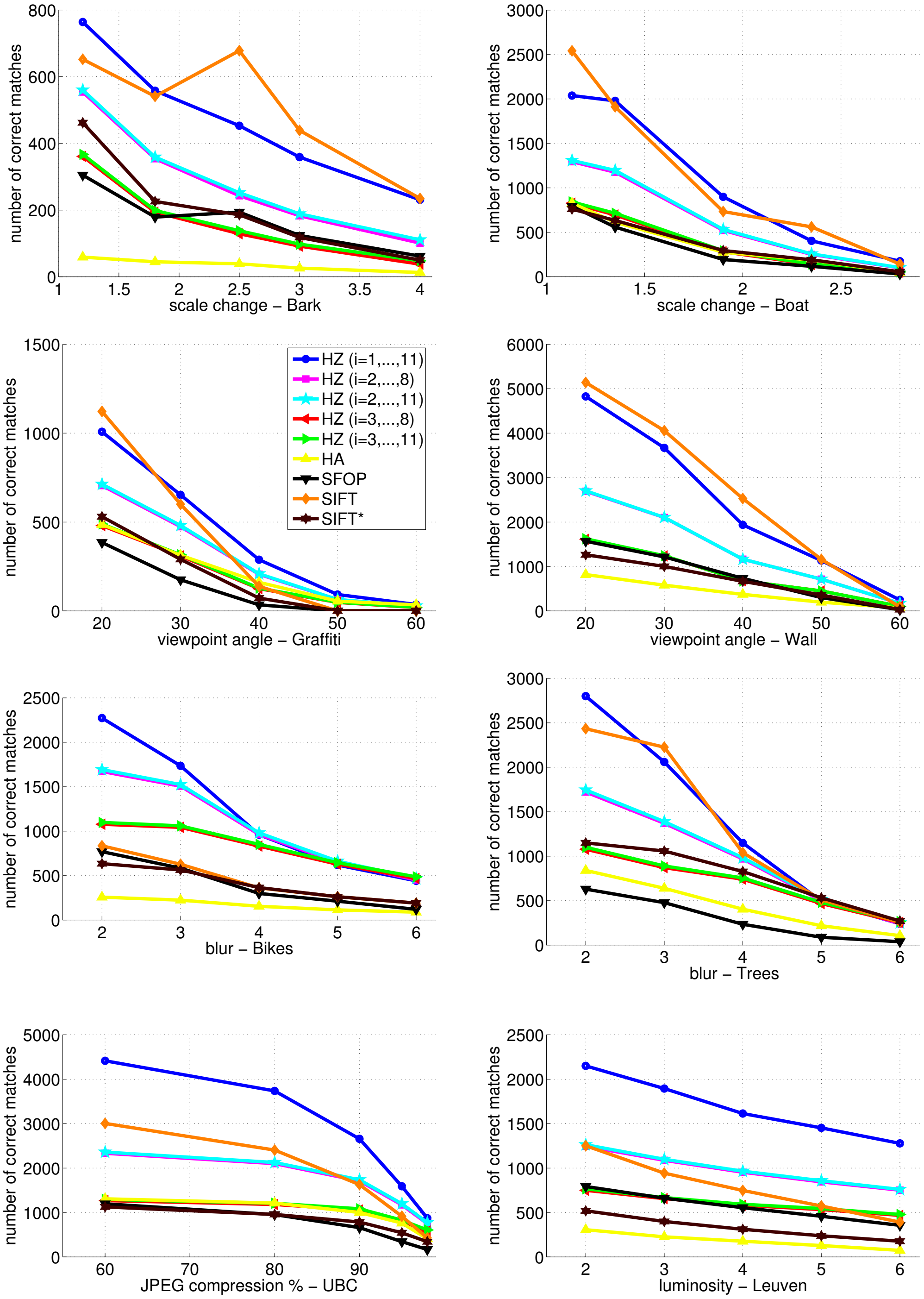


Figure 6.7: Number of correct matches for the Oxford dataset

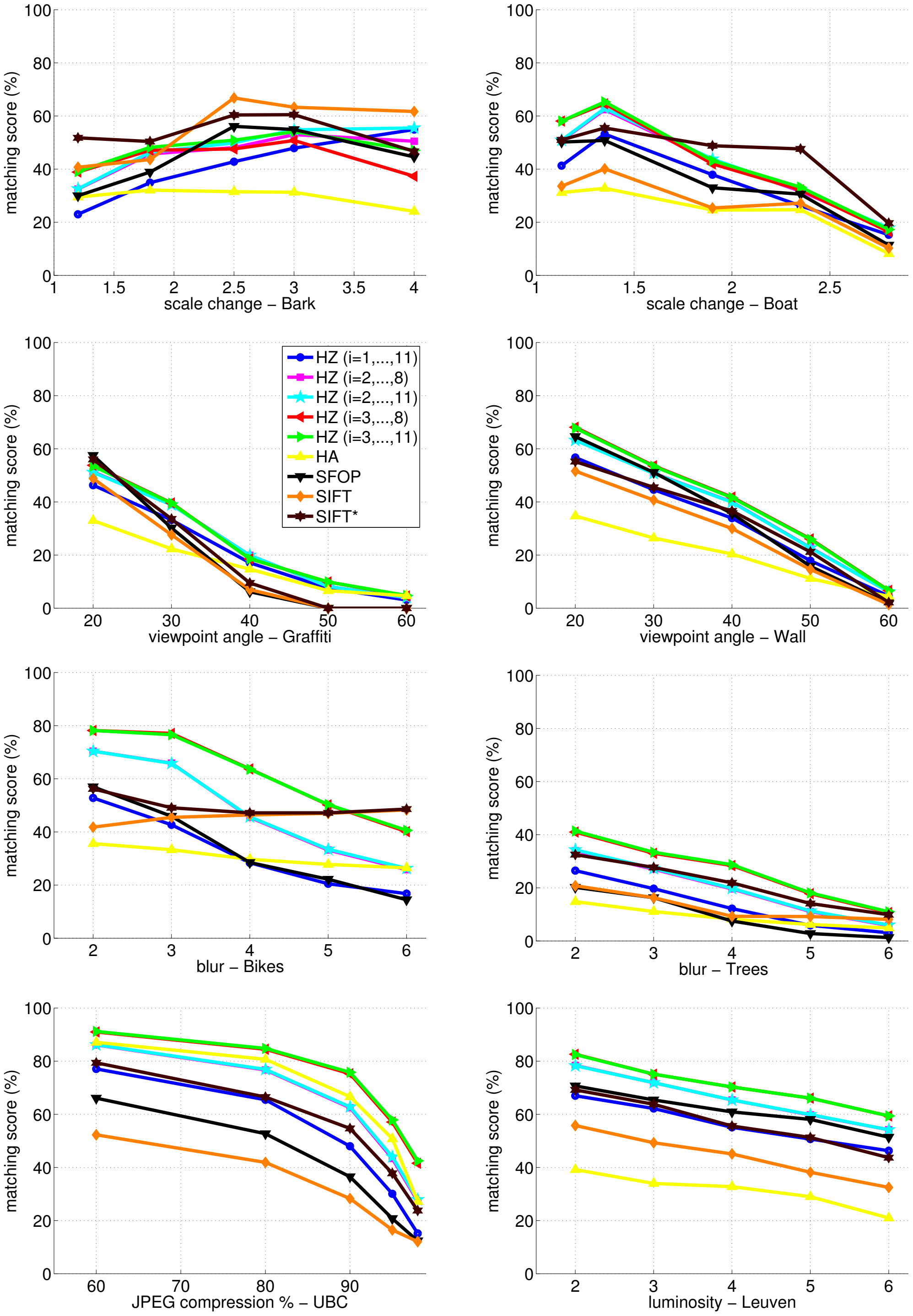


Figure 6.8: Matching score for the Oxford dataset

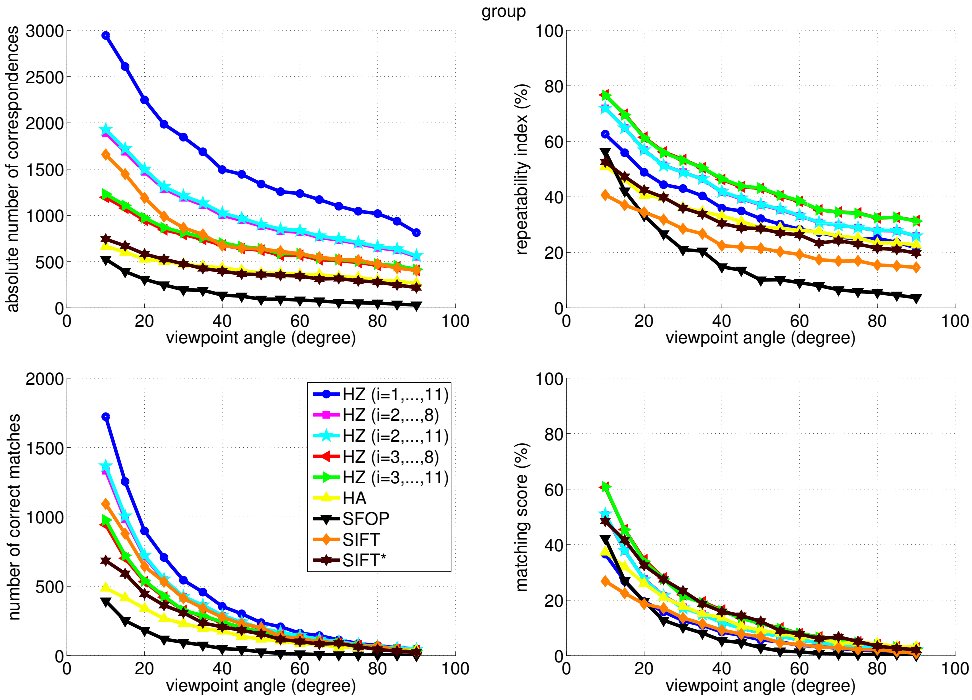


Figure 6.9: Results on the group sequence

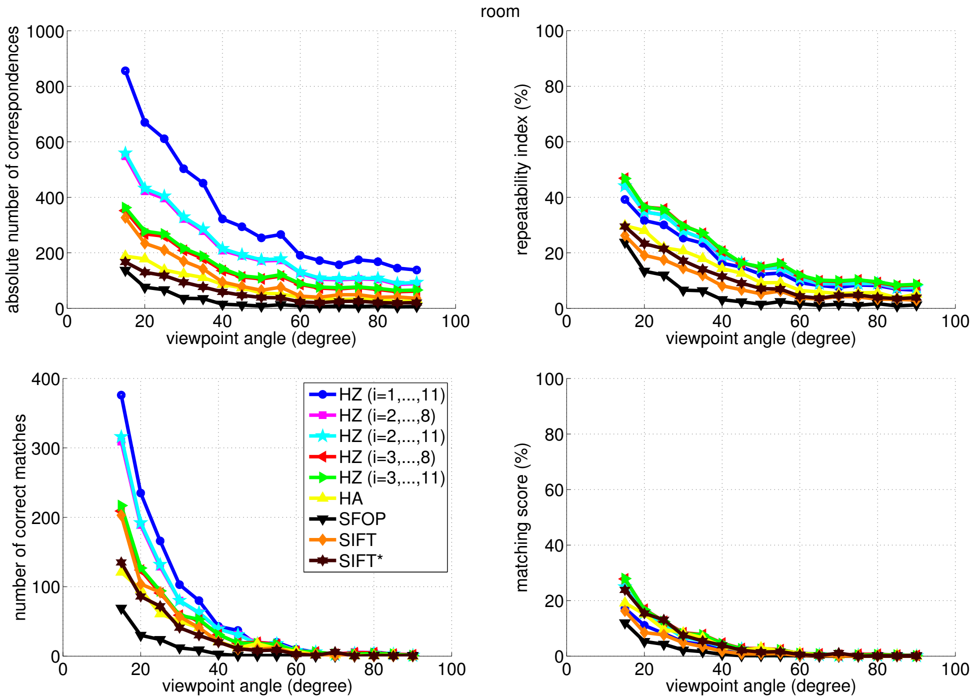


Figure 6.10: Results on the room sequence

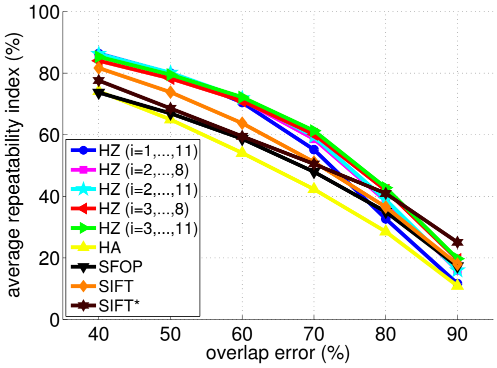


Figure 6.11: Average repeatability index for different overlap errors on the Oxford dataset

kinds of transformations applied in the Oxford dataset. These considerations still hold for the non-planar sequences in the three dimensional extension of the Oxford dataset.

The z-score function normalizes the determinant and the trace used by the Harris function and does not demand the user to set the value of the linear combination coefficient, which depends on the image to analyse. The mask to limit the search only to corners close to the edges returns a stable output, which is no longer affected by the threshold value that relies on the content of the whole image.

It can be noted that corners are complementary to blob-like features, thus they provide additional information that can be combined to obtain better results. We have proved experimentally that a complete pyramidal structure is not required and that just a limited number of scales are sufficient to locate almost all of the corners present in the image.

Though the method is not fast, it is still appropriate for off-line tasks which require high accuracy. A possible application regards the combination of the proposed method with other feature detectors in order to refine the final result.

6.3 The sGLOH descriptor

6.3.1 The main idea

An extension of the GLOH descriptor (see Sec. 3.2.6) is proposed. The new sGLOH (shifting GLOH) descriptor avoids to rotate the feature patch before computing the descriptor vector using a polar grid. Instead of rotating the patch in the estimated dominant orientation, for which an accurate computation can be difficult, the descriptor compares different discrete orientations which can be obtained by shifting the descriptor vector at a reasonable computational cost, and the best is selected.

Though the RIFT descriptor is also rotational invariant (see Sec. 3.2.6), its performances are similar to those provided by the spin image [85], which was shown to be less good than the SIFT in a popular comparison [113]. Though the proposed approach to get the best orientation is not a novelty [178], it has never been extended and evaluated on the SIFT framework.

6.3.2 Descriptor details

The sGLOH descriptor grid is made up of n circular rings centered on the feature point. Each ring contains m regions, equally distributed along m directions, defining a region $\mathcal{R}_{r,d}$ with $r = \{1, 2, \ldots, n\}$ and $d = \{0, 1, \ldots, m-1\}$. The inner circular region can be divided in m radial sectors (see fig. 6.12) defining the single region $\mathcal{R}_{0,0}$ or more regions $\mathcal{R}_{0,d}$. Given a descriptor vector H, a function $\Psi(H)$ is defined as equal to 0 if a single region is defined, 1 otherwise.

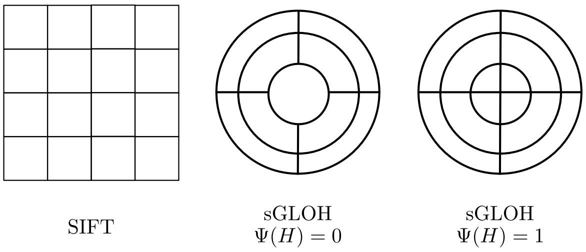


Figure 6.12: Cartesian grid used by the SIFT descriptor (left) and circular grids investigated by the sGLOH descriptor

For each region, the orientation histogram weighted by the gradient magnitude is computed in m quantized orientation. In order to obtain a better estimation of the gradient distribution, for each region the bin value h_i where $i = 0, 1, \ldots, m-1$ is computed by the Gaussian kernel density estimation

$$h^i_{r,d} = \frac{1}{\sqrt{2\pi}\sigma} \sum_{\mathbf{p}\in\mathcal{R}_{r,d}} G_m(\mathbf{p})\, e^{-\frac{(M_{2\pi}(G_d(\mathbf{p})-m_i))^2}{2\sigma^2}} \tag{6.7}$$

where $G_m(\mathbf{p}), G_d(\mathbf{p})$ are respectively the gradient magnitude and orientation of a pixel $\mathbf{p}$ in the region $\mathcal{R}_{r,d}$, with $r = \{0, 1, \ldots, n\}$ and $d = \{0, 1, \ldots, m-1\}$, $m_i = \frac{2\pi}{m} i$ is the i-th bin center, and $\sigma = \frac{2\pi}{m} c$, with $c \in \mathbb{R}^+$, is the standard deviation. The function $M_q(x)$ is used to take into account a periodicity of length q

$$M_q(x) = \begin{cases} x & \text{if} \quad x < \frac{q}{2} \\ q - x & \text{otherwise} \end{cases} \tag{6.8}$$

In modular arithmetic, $[i+d]_m$ shifts cyclically by d positions the element i of a vector of size m, given the congruence modulo m relation $a \equiv b(mod\; m)$, where the congruence class is represented by $[a]_m$. Defining a block histogram

$$H_{r,d} = \bigoplus_{i=0}^{m-1} h^{[d+i]_m}_{r,d}, \tag{6.9}$$

where $\bigoplus$ is the concatenation operator, so that for each block the first bin has direction d, the final descriptor vector H is obtained by concatenating histograms

$$H = \bigoplus_{i=0}^{n} \bigoplus_{j=0}^{m-1} H_{i,j} \tag{6.10}$$

where $H_{0,k}$ for $k = 1, \ldots, m-1$ is not considered if $\Psi(H) = 0$. The final descriptor length is $l = m(mn + 1 + (m-1)\Psi(H))$.

The rotation of the descriptor by a factor αk where $\alpha = \frac{2\pi}{m}$ is obtained by a cyclic shift of the block histogram inside a ring (see fig. 6.13)

$$H_{\alpha k} = \begin{cases} \displaystyle\bigoplus_{i=0}^{n} \bigoplus_{j=0}^{m-1} H_{i,[k+j]_m} & \text{if } \Psi(H) = 1 \\ \\ \displaystyle H_{0,k} \bigoplus_{i=1}^{n} \bigoplus_{j=0}^{m-1} H_{i,[k+j]_m} & \text{otherwise} \end{cases} \tag{6.11}$$

where $H_{0,k} = \bigoplus_{i=0}^{m-1} h_{0,d}^{[i+k+d]_m}$. The distance between two features H and $\overline{H}$ is then given by

$$\widehat{\mathcal{D}}(H, \overline{H}) = \min_{k=0,\ldots,m-1} \mathcal{D}(H, \overline{H}_{\alpha k}) \tag{6.12}$$

where $\mathcal{D}(\cdot, \cdot)$ is a common distance measure (see Sec. 4.1 and Sec. 4.3) and each descriptor vector has been normalized to the unit length.

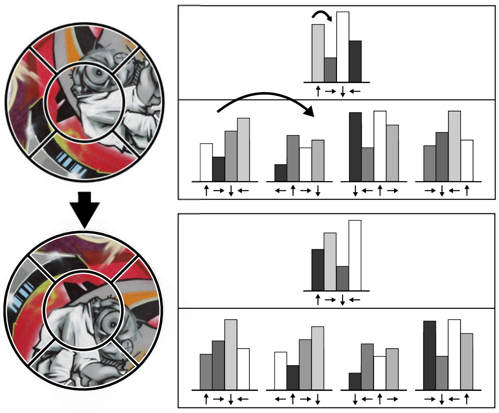


Figure 6.13: A discrete rotation of the feature patch can be obtained for the sGLOH descriptor by shifting the block histograms and the bins inside each block

6.3.3 Experimental result

The sGLOH descriptor has been compared with both the SIFT and the GLOH descriptors on the well-known Oxford database (see Sec. 5.3). The same experimental setup described in [113] has been used, while key points have been extracted with the HarrisZ detector which has been proved to give robust and stable features. The first and fourth images of each sequence in the dataset have been used for the validation.

The number of correct matches and of correspondences is computed according to the overlap error (see Sec. 5.1). As in [113], the overlap error is fixed to $\varepsilon = 0.5$, the feature patch is 41×41 pixels, while the nearest neighbour matching strategy is adopted. The sGLOH descriptor has been tested for $c = 0.7$, $m = 8$, $n = 0, 1, 2$, so that the descriptor grid radii are respectively $\{20\}, \{12, 20\}, \{7, 13, 20\}$ pixels. When $n = 1, 2$, also the different cases $\Psi(H) = 0, 1$ have been examined, obtaining descriptors of lengths $l = 64, 72, 128, 136, 192$.
The best distance measures between L_1, L_2 have been used for each descriptor. The sGLOH descriptor performs better with L_1, while the L_2 distance was used in the case of the SIFT and the GLOH descriptors.

Plots are shown in figs. 6.14–6.15. The results are comparable with those obtained by SIFT (blue surface) and the GLOH (green surface), in particular the sGLOH descriptor with length $l = 128$ represents a good compromise between the length and the performances of the descriptor.

Only in the case of the "bark" sequence, the SIFT performs better, but only for really high precision. This can be due to the kernel density estimation used in the sGLOH descriptor which smooths the patch, because the bark sequence is heavy textured and the smooth can decrease the discriminative power of the method. From the plots it can also be seen that sGLOH detector presents local minima for $l = 72, 136$, underlining better result for the configurations with $\Psi(H) = 1$.

While plots in fig. 6.14 refer to scale, rotation and viewpoint changes and are more relevant in the test, the plots in fig. 6.15 are also helpful. In fact, though the latter image sequences do not present any scale, rotation or viewpoint changes, so clearly the proposed sGLOH detector will give the better estimation of the patch orientation, they provide clues that the dominant gradient orientation estimation used by the SIFT and the GLOH descriptors can lead sometimes to bad patch alignments, not due only to repeated patterns in the images which also affect the sGLOH descriptor.

The sGLOH descriptors does not introduce any relevant computational cost in the generation of the descriptor vector. The new distance measure $\widehat{d}$ is more time consuming, but still acceptable. Plot in fig. 6.16 shows the average cumulative running time for the SIFT, the GLOH and the sGLOH descriptors respectively for each image sequence plus the time to compute the distance for all vector pairs. As it can be seen the increasing time is reasonable for all sequences, also considering that the matching cost is quadratic with respect to the number of descriptors. Only for a very high number of features as in the "wall" sequence and for a descriptor length $l > 128$ the proposed descriptor is slower.

As final remarks, the sGLOH descriptor retains good results in terms of robustness and stability, at a reasonable increase in the computational cost.

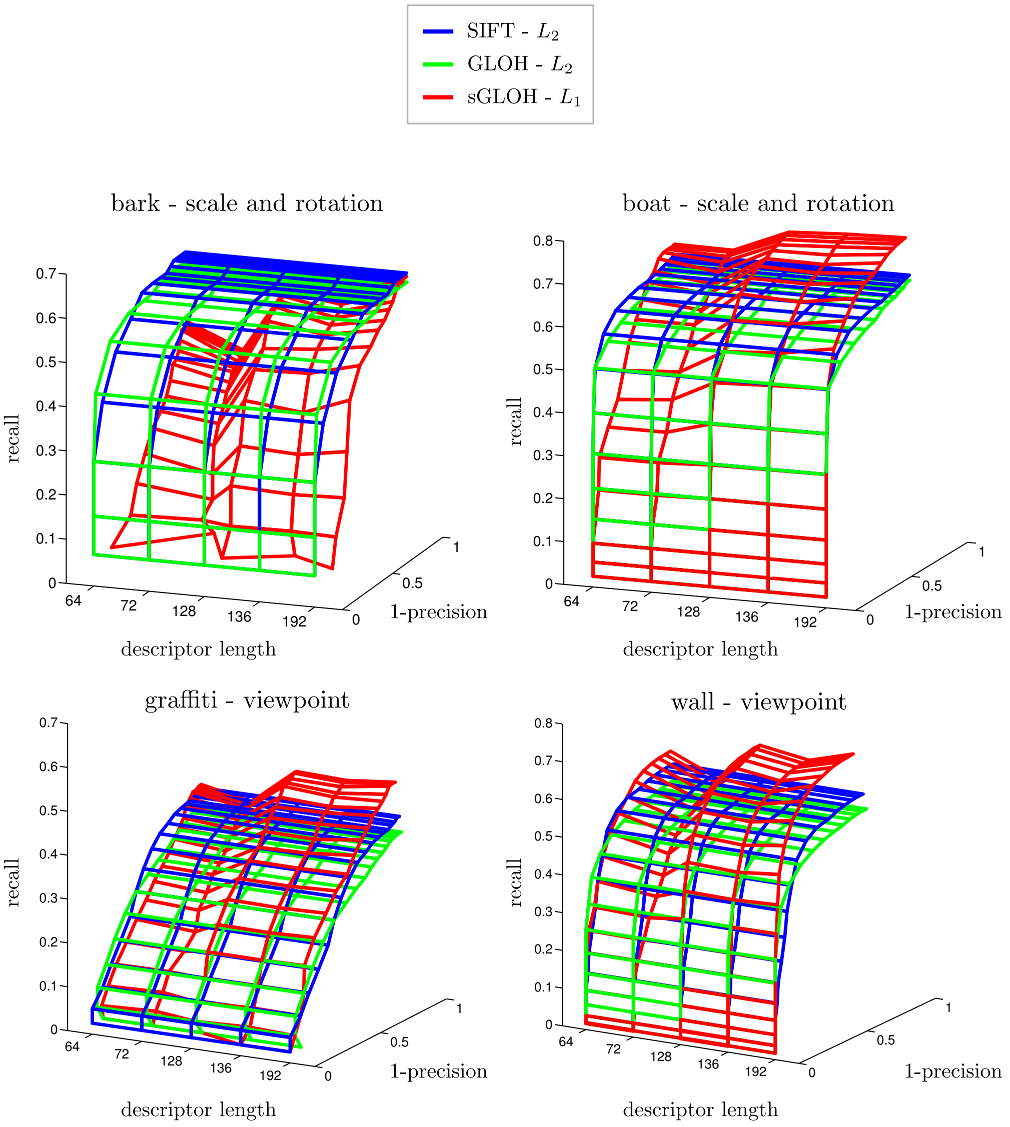


Figure 6.14: 1-precision/recall plots for scale, rotation and viewpoint changes on the Oxford database

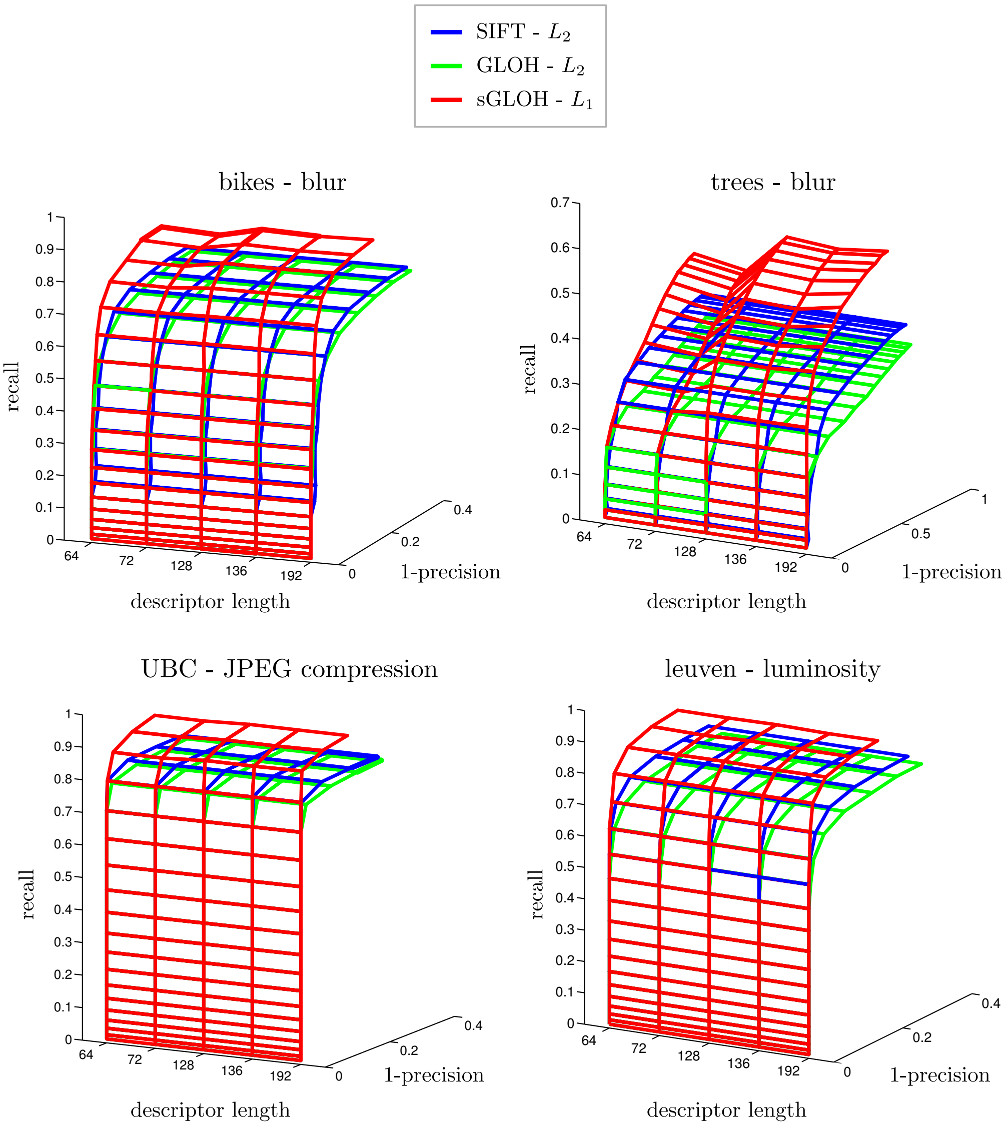


Figure 6.15: 1-precision/recall plots for blur, JPEG compression and illumination changes on the Oxford database

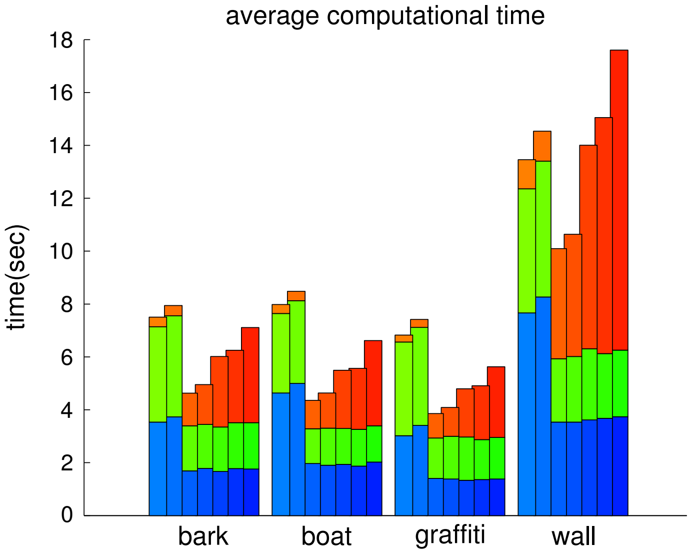


Figure 6.16: The average cumulative time on 15 runs for the “bark”, “boat”, “graffiti” and “wall” sequences. The blue and the green bars refer to the time required by the descriptors, while the red bar refers to the time spent to evaluate the distance. Each cluster reports from left to right the time for the SIFT, GLOH and sGLOH with descriptor length of 64, 72, 128, 136, 192

Chapter 7

Stereo matching

7.1 Overview

Stereo vision is one of the principal topic in computer vision. Given two cameras with known *intrinsic and extrinsic parameters*, it is possible to get the three-dimensional structure of the scene from the point correspondences in these two images by *triangulation* [70]. Moreover, the calibration parameters [70] can be extracted by a relative small amount of point correspondences and their positions in the three-dimensional space [70].

Furthermore, the stereo *epipolar geometry* [70, 44, 174] allows to constrain possible matching candidates to lie on a straight line and a three-dimensional reconstruction can be obtained. The main object of the epipolar geometry is the *fundamental matrix*, which can be computed by an exiguous number of point matches between the stereo pairs [70] and allows a three-dimensional reconstruction up to a projective transformation [70]. When the camera intrinsic parameters are also known the *essential matrix* [70] can be computed and a fully Euclidean reconstruction is possible [70]. In some cases (in particular at least three views are needed) by imposing some constrains on the camera parameters it is also possible to obtain the Euclidean reconstruction with the fundamental matrix by means of *autocalibration* algorithms [70]. Moreover, by combining the matches provided by more than a stereo pair, further matches can be obtained as well as their consistency can be improved [48].

Multiview reconstruction extends these approaches and obtains finer results [70, 161] (see fig. 7.1). Usually a new camera is added at each time, which leads to very time expensive algorithms, the structure is recovered by *factorization* [166, 70] and the *bundle adjustment* is used to minimize the error on the whole scene framework [173, 70, 44, 156]. Recent new methods [155, 64] allow to merge different cameras in an efficient way.

The feature matching is also a relevant topic in the *disparity map estimation.* Nowadays *sparse disparity maps* have been replaced by *dense disparity maps* [161], which allow a more precise three-dimensional reconstruction. Before applying a good algorithm to estimate the dense disparity map, images have to be *rettified* [63, 133], which usually implies the estimation of the fundamental matrix. Matching features obtained by a sparse depth map can also be used as seed points for extending a dense disparity map [89, 191], especially in the case of a *wide-baseline* stereo matching [65]. Wide-baseline stereo is harder than a *narrow-baseline* disparity estimation, where dense disparity algorithms work well [161, 167, 147]. Besides, points corre-

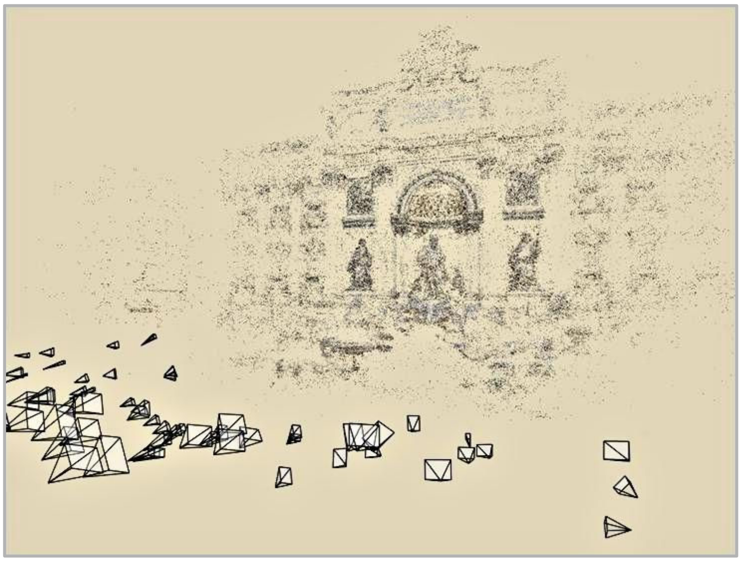


Figure 7.1: A multi view reconstruction of the Trevi Fountain obtained by the Photo Tourism software. Image from [156]

spondences can also be used for *image registration* [161] and for body modelling and tracking [161].

Simply matching strategies described in Sec. 5.3, generate a large number of *outliers* which seriously affect the accuracy of the estimation of a plane homography or of the fundamental matrix. Several algorithms based on *robust statistics* has been developed to find the best set of *inliers*, most of them are based on the RANSAC (RANdom SAmple Consensus) approach [51], which provide a relatively fast and robust inlier estimation.

7.2 The stereo framework

7.2.1 The camera model

The perspective camera is modelled as a matrix $\mathrm{P} \in \mathbb{R}^{3\times4}$ which takes a three-dimensional point $\overline{\mathbf{X}}$ in homogeneous coordinates (see Sec. 1.3.1) and projects it to $\overline{\mathbf{x}}$ in the image plane π

$$\overline{\mathbf{x}} \simeq \mathrm{P}\overline{\mathbf{X}} \tag{7.1}$$

In particular the *pinhole camera model* [70, 174, 44] (see fig. 7.2) is used to describe a standard camera. Under the pinhole camera model the camera matrix P can be decomposed as $\mathrm{K}\,[\mathrm{R}|\mathbf{t}]$ where $\mathrm{K} \in \mathbb{R}^{3\times3}$ represent the camera intrinsic parameters and the matrix $[\mathrm{R}|\mathbf{t}] \in \mathbb{R}^{3\times4}$ embeds the Euclidean transformation from the world to the camera coordinate frames [70], i.e. a three-dimensional point $\mathbf{X}$ is mapped in the camera coordinate frame (see fig. 7.2) as

$$\mathbf{X}_{cam} = \mathrm{R}\mathbf{X} + \mathbf{t} \tag{7.2}$$

or

$$\overline{\mathbf{X}}_{cam} \simeq [\mathrm{R}|\mathbf{t}]\,\overline{\mathbf{X}} \tag{7.3}$$

In the camera coordinate frame the matrix K projects the ray from the *camera centre* $\mathbf{C} = \mathbf{0}$ through a three-dimensional point $\mathbf{X}_{cam}$ into the point $\overline{\mathbf{x}}$ on the *image plane* π at the distance f from the *principal axis* [70] (see fig. 7.2)

$$\overline{\mathbf{x}} \simeq \mathrm{K}\mathbf{X}_{cam} = \begin{bmatrix} f & & c_x \\ & f & c_y \\ & & 1 \end{bmatrix} \mathbf{X}_{cam} \tag{7.4}$$

where c_x and c_y are the coordinates of the *principal point* in the plane image reference frame. More in general, to take into account the CCD grid of a real camera, which cannot be squared or with orthogonal axes, the matrix K is generalized as [70]

$$\mathrm{K} = \begin{bmatrix} a_x & s & c_x \\ & a_y & c_y \\ & & 1 \end{bmatrix} \tag{7.5}$$

i.e. an affine transformation of the image plane coordinates is considered. The 3×4 projective camera P has 11 degrees of freedom because camera matrices are equal up to scale and the camera centre $\overline{\mathbf{C}}$ is given by the right null space of P, i.e. $\mathrm{P}\overline{\mathbf{C}} \simeq \mathbf{0}$. For further information about the camera matrix properties see [70].

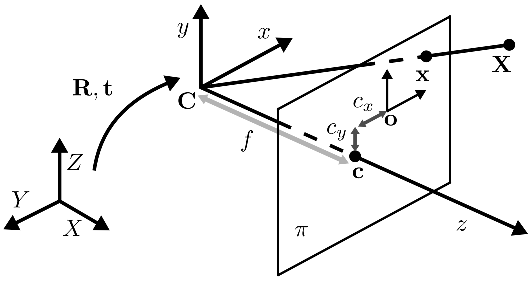


Figure 7.2: Under the pin-hole camera model, a three-dimensional point $\mathbf{X}$, expressed according to the reference frame X, Y, Z, is projected into the point $\mathbf{x}$ on the image plane π through the ray from the camera centre $\mathbf{C}$. The camera reference frame x, y, z is obtained by the rotation matrix R and the translation vector T. The z axis intersect the image plane π in the principal point $\mathbf{c}$, where its distance from the camera centre is given by the focal length f, while its coordinates in the image plane reference frame are $[c_x, c_y]$. Image adapted from [174]

Image coordinates can be *normalized* [70] by pre-multiplying them with $\mathbf{K}^{-1}$, i.e. $\overline{\mathbf{x}}' = \mathrm{K}^{-1}\overline{\mathbf{x}}$ so that only the extrinsic parameters have to be considered because

$$\overline{\mathbf{x}}' \simeq \mathrm{K}^{-1}\overline{\mathbf{x}} \simeq \mathrm{K}^{-1}\mathrm{P}\overline{\mathbf{X}} \simeq \mathrm{K}^{-1}\mathrm{K}\,[\mathrm{R}|\mathbf{t}]\,\overline{\mathbf{X}} \simeq [\mathrm{R}|\mathbf{t}]\,\overline{\mathbf{X}} \tag{7.6}$$

where the *normalized camera matrix* [70] is given by $P' = [R|\mathbf{t}]$.

To better characterized the camera model, the *radial distortion* correction factors [70] can also be incorporated to the model (see fig. 7.3). If $\mathbf{r} = [r_x, r_y]^T$ is the centre of the radial distortion, the correction factor applied to the point $\mathbf{x}_r$ after the projection to the image plane could be modelled as

$$\mathbf{x} = \mathbf{r} + L(\| \mathbf{x}_r - \mathbf{r} \|^2)(\mathbf{x}_r - \mathbf{r}) \tag{7.7}$$

where the error divergence can be approximated by Taylor expansion [70]

$$L(r) = 1 + \kappa_1 r + \ldots + \kappa_n r^n \tag{7.8}$$

It can be noted that the radial distortion error increases with the distance from the radial distortion centre which usually is close to the image centre.

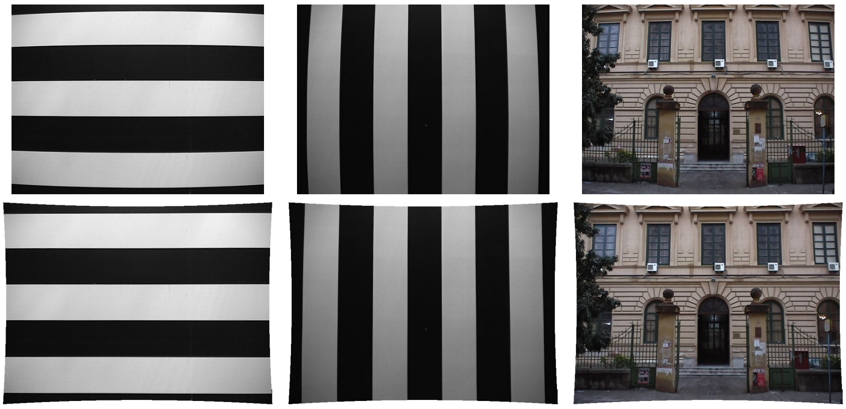


Figure 7.3: Calibration patterns used to estimate the radial distortion and a test image (top). The same images after the correction is applied by the method described in [7] (bottom)

7.2.2 The epipolar stereo geometry

As shown in fig. 7.4, given two cameras that see the same scene, the plane $\pi_{\mathbf{X}}$ through the three-dimensional point $\mathbf{X}$ and the two camera centre $\mathbf{C}_1$, $\mathbf{C}_2$, known as *epipolar plane*, is projected into the straight line $l^1_{\mathbf{X}}$,$l^2_{\mathbf{X}}$, called *epipolar lines* on the two image planes π_1, π_2 [70]. The *baseline*, i.e. the line joining the two camera centres, intersects the image planes into two points e_1,e_2 called *epipoles*, which are common to all the respective epipolar lines for different points $\mathbf{X}$. The epipoles describe the stencils of epipolar lines through them.

7.2.3 Triangulation

A narrow baseline provides a easier estimation of the correspondences because the coordinates of a point on an image should not be searched far in the other view.

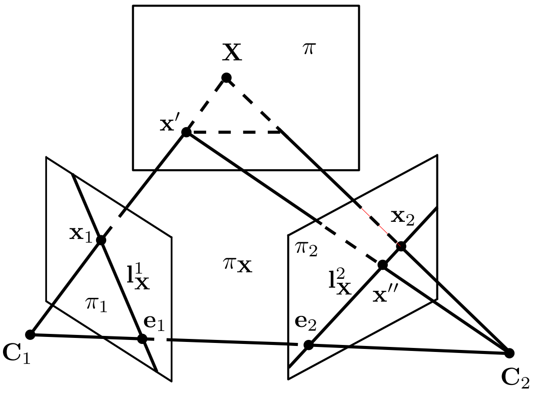


Figure 7.4: Given a stereo camera pairs specified respectively by the the image planes π_1, π_2 and the camera centres $\mathbf{C}_1$, $\mathbf{C}_2$, a three dimensional point $\mathbf{X}$ is projected into the points $\mathbf{x}_1$, $\mathbf{x}_2$ in the image planes, which both lie on the epipolar plane $\pi_{\mathbf{X}}$ given by the point $\mathbf{X}$ an the camera centres $\mathbf{C}_1$, $\mathbf{C}_2$. The line through the camera centres is the baseline, which intersects the image planes in the respective epipoles $\mathbf{e}_1$, $\mathbf{e}_2$, while the lines $\mathbf{l}^1_{\mathbf{X}}$, $\mathbf{l}^2_{\mathbf{X}}$ in the image planes from the epipoles through the point projections $\mathbf{x}_1$, $\mathbf{x}_2$, are known as epipolar lines. A plane π not passing through the baseline induces a plane homography H_π, which projects a point of the ray from $\mathbf{C}_1$ through $\mathbf{x}_1$ into the epipolar line $\mathbf{l}^2_{\mathbf{X}}$ and in similar way for the other camera. Image from [70]

A wide baseline is better to estimate the three-dimensional position of a point in the space by triangulation (see fig. 7.5). For error free camera matrices and point coordinates, the three dimensional location of a point is given by the intersection of the corresponding rays from the cameras (see fig. 7.4), while in real cases an estimation is provided by minimizing an error cost [70, 174, 161]. As it can be seen from fig. 7.5, when corresponding rays are almost parallel in the narrow baseline case, the error in the estimation of the position is greater [70].

Given the projection $\mathbf{x}_1$ of $\mathbf{X}$ on the plane π_1, and the epipoles, the projection $\mathbf{x}_2$ on π_2 is constrained to lay on the epipolar line $l^2_{\mathbf{X}}$, and in a similar way for the projection $\mathbf{x}_1$ on the epipolar line $l^1_{\mathbf{X}}$.

7.2.4 The fundamental matrix

The epipolar line relationship can be expressed by mean of the *fundamental matrix* F [102]. Given a plane π not passing through the two camera centers $\mathbf{C}_1$, $\mathbf{C}_2$, a ray from $\mathbf{C}_1$ to a point $\mathbf{X}$ intersects the plane π in a point $\mathbf{x}'$ and the planes π_1, π_2 in $\mathbf{x}_1$, $\mathbf{x}_2$ respectively. The projection $\mathbf{x}''$ of $\mathbf{x}'$ in the image plane π_2 through the ray passing through $\mathbf{C}_2$ must lay on the epipolar line $l^2_{\mathbf{X}}$ because $\mathbf{x}'$ is on the epipolar plane $\pi_{\mathbf{X}}$ (see fig. 7.4), so that lines are mapped to lines by the transformation

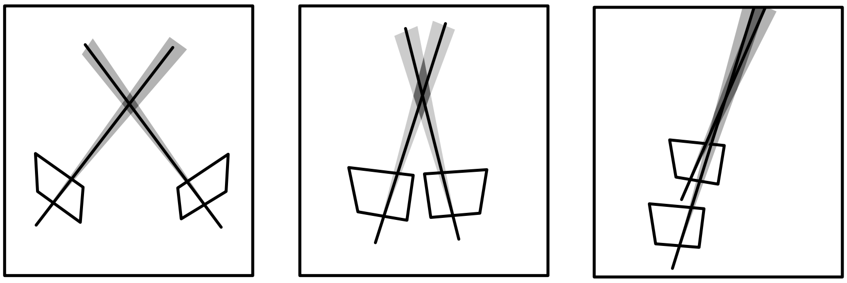


Figure 7.5: Different stereo configurations from a wide baseline to a narrow baseline (left to right). As the baseline distance decreases, though the error in point estimation of the stereo pairs remains the same (light gray cone), the uncertainty in the deep estimation of the three dimensional point increases (dark gray area). Images from [70]

inducted by the plane π, i.e. the plane π induces an homography from $\mathbf{x}_1$ to $\mathbf{x}''$ [70]

$$\overline{\mathbf{x}}'' \simeq \mathrm{H}_\pi \overline{\mathbf{x}}_1 \tag{7.9}$$

In homogeneous coordinates the line through two points $\mathbf{a}$, $\mathbf{b}$ is given by the cross product $\bar{l} \simeq \overline{\mathbf{a}} \times \overline{\mathbf{b}}$, which can be expressed as the matrix product $\bar{l} \simeq [\overline{\mathbf{a}}]_\times \overline{\mathbf{b}}$ where

$$\overline{\mathbf{a}} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} \qquad \text{and} \qquad [\overline{\mathbf{a}}]_\times = \begin{bmatrix} 0 & -z & y \\ z & 0 & -x \\ -y & x & 0 \end{bmatrix} \tag{7.10}$$

moreover $\overline{\mathbf{a}}^T \bar{l} = 0$, i.e. a point lies on a line if their dot product is equal to zero. The epipolar line $l^2_{\mathbf{X}}$ joins the epipole $\mathbf{e}_2$ and the point $\mathbf{x}''$

$$\bar{l}^2_{\mathbf{X}} \simeq \overline{\mathbf{e}}_2 \times \overline{\mathbf{x}}'' \simeq \overline{\mathbf{e}}_2 \times \mathrm{H}_\pi \overline{\mathbf{x}}_1 \simeq [\overline{\mathbf{e}}_2]_\times \mathrm{H}_\pi \overline{\mathbf{x}}_1 \tag{7.11}$$

and since $\mathbf{x}_2$ lies on $l^2_{\mathbf{X}}$ the fundamental matrix relation is obtained as [70]

$$\overline{\mathbf{x}}_2^T \bar{l}^2_{\mathbf{X}} = \overline{\mathbf{x}}_2^T [\overline{\mathbf{e}}_2]_\times \mathrm{H}_\pi \overline{\mathbf{x}}_1 = \overline{\mathbf{x}}_2^T \mathrm{F} \overline{\mathbf{x}}_1 = 0 \tag{7.12}$$

where the fundamental matrix F is

$$\mathrm{F} \simeq [\overline{\mathbf{e}}_2]_{\times \mathrm{H}_\pi} \tag{7.13}$$

A similar derivation can be obtained starting by the epipolar line $l^2_{\mathbf{X}}$. The fundamental matrix F has rank 2 because it raises from the matrix product by the matrix $[\overline{\mathbf{e}}_2]_\times$, which has rank 2 in the non-degenerate case [70], and the full rank matrix H_π. Moreover, it has 7 degrees of freedom since it is not a full rank matrix and it is known up to scale [70]. The epipolar lines are obtained as

$$\begin{aligned} \bar{l}^2_{\mathbf{X}} &\simeq \mathrm{F} \overline{\mathbf{x}}_1 \\ \bar{l}^1_{\mathbf{X}} &\simeq \overline{\mathbf{x}}_2^T \mathrm{F}^T \end{aligned} \tag{7.14}$$

while the epipoles $\overline{\mathbf{e}}_1$, $\overline{\mathbf{e}}_2$ are respectively the right and left null spaces of $\mathbf{F}$, because for all points in the image planes π_1, π_2 the epipolar lines for both the cameras should pass through them [70], i.e.

$$\begin{aligned} \overline{\mathbf{e}}_2^T \mathrm{F} \overline{\mathbf{x}}_1 = 0\,, \forall \mathbf{x}_1 \;&\Rightarrow\; \overline{\mathbf{e}}_2^T \mathrm{F} = \mathbf{0} \\ \overline{\mathbf{x}}_2^T \mathrm{F} \overline{\mathbf{e}}_1 = 0\,, \forall \mathbf{x}_2 \;&\Rightarrow\; \mathrm{F} \overline{\mathbf{e}}_1 \;= \mathbf{0} \end{aligned} \tag{7.15}$$

When the image point coordinates are normalized (see Sec. 7.2.1) the fundamental matrix is called *essential matrix* [70]. The essential matrix $\mathrm{E} = \mathrm{K}_2^T \mathrm{F} \mathrm{K}_1$ can be easily derived from the fundamental matrix F and the matrices K_1, K_2 of the intrinsic parameters because

$$\overline{\mathbf{x}}_2'^T \mathrm{E} \overline{\mathbf{x}}_1' \simeq \overline{\mathbf{x}}_2^T \mathrm{K}_2^{-T} \left(\mathrm{K}_2^T \mathrm{F} \mathrm{K}_1 \right) \mathrm{K}_1^{-1} \overline{\mathbf{x}}_1 \simeq \overline{\mathbf{x}}_2^T \mathrm{F} \overline{\mathbf{x}}_1 \tag{7.16}$$

where $\overline{\mathbf{x}}_1'$, $\overline{\mathbf{x}}_2'$ are the normalized coordinates.

A stereo pair can be transformed by homographies so that corresponding epipolar lines corresponds to the same scanline (see fig. 7.6). This transformation is called *image rectification* [63, 133] and it is commonly used as preprocessing stage for dense disparity algorithms. For more information about the topics of this section see [70, 44, 174, 161].

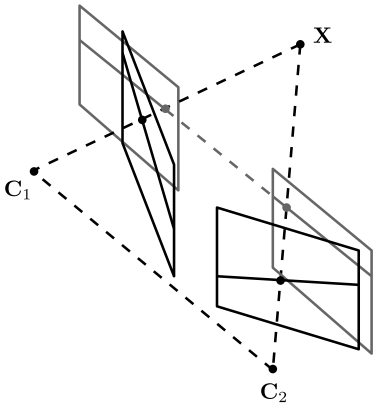


Figure 7.6: In the rectification process the image cameras (black stereo pair) are moved (usually by rotating and scaling the image plane) so that corresponding epipolar lines lie on the same line (gray stereo pair). Image from [174]

7.3 Fundamental matrix computation

Since planar homographies (see Sec. 1.3.1) also play a crucial role in the three-dimensional framework, before discussing the methodologies to compute the fundamental matrix, the computation of planar homographies is introduced.

7.3.1 Planar homography computation

Two homogeneous vectors $\overline{\mathbf{a}}$, $\overline{\mathbf{b}}$ representing the same inhomogeneous points are parallel, i.e. they differ only by a scale factor, if their cross product is equal to zero [70]

$$\overline{\mathbf{a}} \simeq \overline{\mathbf{b}} \Rightarrow \overline{\mathbf{a}} \times \overline{\mathbf{b}} = 0 \tag{7.17}$$

For homographies the following expression then holds

$$\overline{\mathbf{x}}'_i \simeq \mathrm{H}\overline{\mathbf{x}}_i \Rightarrow \overline{\mathbf{x}}'_i \times \mathrm{H}\overline{\mathbf{x}}_i = 0 \tag{7.18}$$

where $\overline{\mathbf{x}}'_i = [x'_i, y'_i, w'_i]^T$ and $\overline{\mathbf{x}}_i = [x_i, y_i, w_i]^T$, which can be written as

$$\mathrm{A}\mathbf{h} = \begin{bmatrix} \mathbf{0}^T & -w'_i\mathbf{x}'^T_i & -y'_i\mathbf{x}'^T_i \\ w'_i\mathbf{x}'^T_i & \mathbf{0}^T & -x'_i\mathbf{x}'^T_i \\ -y'_i\mathbf{x}'^T_i & x'_i\mathbf{x}'^T_i & \mathbf{0}^T \end{bmatrix} \begin{bmatrix} \mathbf{h}^1 \\ \mathbf{h}^2 \\ \mathbf{h}^3 \end{bmatrix} = \mathbf{0} \tag{7.19}$$

where

$$\mathrm{H} = \begin{bmatrix} \mathbf{h}^{1T} \\ \mathbf{h}^{2T} \\ \mathbf{h}^{3T} \end{bmatrix} \tag{7.20}$$

which for each corresponding pairs of points $\overline{\mathbf{x}}'_i$, $\overline{\mathbf{x}}_i$ gives rise to three equations, whose two only are linear independent [70]. The homography matrices have 8 degrees of freedom because they are equal up to scale, thus 4 points suffice in determining the homography by using two of the set of three equations obtained for each point. The solution of the obtained *homogeneous system* [70] is given by the right null space of the matrix A when only 4 points are used.

For an over determined system, i.e. when more than 4 point correspondences are used, more robust to noise, the linear system can be solved by imposing for an element of the matrix A the constrain $h_{ij} = 1$ or by using the *Singular Value Decomposition* (SVD) [70, 161, 174].

The SVD decomposition of a generic matrix $\mathrm{A} \in \mathbb{R}^{m \times n}$ is

$$\mathrm{A} = \mathrm{U}\Sigma\mathrm{V}^T \tag{7.21}$$

where $\mathrm{U} \in \mathbb{R}^{m \times p}$ and $\mathrm{V} \in \mathbb{R}^{p \times n}$ with $p = \min(m, n)$ are orthogonal matrices and $\Sigma \in \mathbb{R}^{p \times p}$ is a diagonal matrix [161]. The diagonal element of Σ are known as *singular values* while the columns of U, V are called respectively the left and right *singular vectors* [161].

It can be proved [174] that finding the vector $\mathbf{h}$ which minimizes $\parallel \mathrm{A}\mathbf{h} \parallel$ subject to $\parallel \mathbf{h} \parallel = 1$, is an approximation of the searched homography matrix. The solution is given by the right singular vector corresponding to the smallest singular value of A [174]. This last method involving the SVD decomposition is called *Discrete Linear Trasform* (DLT) algorithm [70]. The DLT algorithm is not stable, however, pre-normalizing the points on two images so that their centroids are at the origin and the mean distance from the origin is $\sqrt{2}$ using two similarity transformation T_1 and T_2 avoids this issue and makes the DLT algorithm very robust and effective in the homography computation [70]. The final homography matrix $\widehat{\mathrm{H}}$ is obtained by deleting the similarity transformation, i.e. $\widehat{\mathrm{H}} = \mathrm{T}_2^T\mathrm{H}\mathrm{T}_1$ [70].

7.3.2 Error cost functions

The cost function used in the DLT algorithm (see Sec. 7.3.1) is called the *algebraic error* [70]

$$\mathcal{D}_{alg}\left(\overline{\mathbf{x}}'_i, \mathrm{H}\overline{\mathbf{x}}_i\right) = \parallel \mathrm{A}\mathbf{h} \parallel^2 \tag{7.22}$$

and it does not have a geometric or statistical meaning, even if it can be related to the the *geometric error* [70]

$$\mathcal{D}_{geom}\left(\overline{\mathbf{x}}'_i, \mathrm{H}\overline{\mathbf{x}}_i\right) = \parallel \mathbf{x}'_i - \mathrm{H}\mathbf{x}_i \parallel^2 \tag{7.23}$$

where with an abuse of notation $\mathrm{H}\mathbf{x}_i$ is the reprojection by the plane homography H of $\mathbf{x}_i$ expressed in inhomogeneous coordinates. Different geometric errors can be considered, as the *symmetric transfer error* [70] assuming the error in one image only

$$\mathcal{D}_{sym}\left(\overline{\mathbf{x}}'_i, \mathrm{H}\overline{\mathbf{x}}_i\right) = \parallel \mathbf{x}'_i - \mathrm{H}\mathbf{x}_i \parallel^2 + \parallel \mathbf{x}_i - \mathrm{H}^{-1}\mathbf{x}'_i \parallel^2 \tag{7.24}$$

or in both images

$$\mathcal{D}_{reproj}\left(\overline{\mathbf{x}}'_i, \mathrm{H}\overline{\mathbf{x}}_i\right) = \parallel \widehat{\mathbf{x}}_i - \mathbf{x}_i \parallel^2 + \parallel \widehat{\mathbf{x}}'_i - \mathbf{x}'_i \parallel^2 \text{ subject to } \widehat{\mathbf{x}}'_i = \mathrm{H}\widehat{\mathbf{x}}_i \tag{7.25}$$

In the last case in addition to the homography matrix also the true point locations $\widehat{\mathbf{x}}_i$ and $\widehat{\mathbf{x}}'_i$ have to be estimated [70]. A last error measure is the *Sampson error* [70], which is the error distance to the first order approximation of the function. If $\mathbf{x}$ and $\widehat{\mathbf{x}}$ are a point and its estimation, for a given function $\mathcal{F}$ its approximation can be written by Taylor expansion up to the first order as

$$\mathcal{F}(\widehat{\mathbf{x}}) = \mathcal{F}(\mathbf{x}) + \frac{\partial \mathcal{F}}{\partial \mathbf{x}}\delta_{\mathbf{x}} \tag{7.26}$$

where $\delta_{\mathbf{x}} = \widehat{\mathbf{x}} - \mathbf{x}$. The Sampson error is defined as

$$\mathcal{D}_{Samp}\left(\widehat{\mathbf{x}}, \mathcal{F}(\mathbf{x})\right) = \left|\left|\frac{\partial \mathcal{F}}{\partial \mathbf{x}}\delta_{\mathbf{x}}\right|\right|^2 \tag{7.27}$$

It can be proved that in the case of homographies the Sampson error is identical to the corresponding geometric error [70]. Moreover the geometric error is also optimal under the assumption of isotropic Gaussian noise, i.e. the minimization the geometric error is equal to find the Maximum Likehood Estimation (MLE) [70] for homographies [70]. For a non-isotropic Gaussian noise distribution, the geometric error should be modified by substituting the Euclidean distance by the Mahalanobis distance (see Sec. 4.2.1) [70].

Thought minimizing the geometric or the Sampson error provides better results, it requires to use an iterative non linear minimization algorithm, such as the Levenberg-Marquardt method [70], and a good initial solution usually obtained by the DLT algorithm (see Sec. 7.3.1). However these methodologies are more computational expensive and the accuracy of the DLT solution is usually good for common computer vision tasks [70].

7.3.3 The eight point algorithm

The DLT algorithm (see Sec. 7.3.1) can also be applied for estimating the fundamental matrix F (see Sec. 7.2.4). In this case the DLT algorithm is known as the *eight point algorithm* [70]. The fundamental matrix constraint $\overline{\mathbf{x}}_i'\mathrm{F}\overline{\mathbf{x}}_i = 0$ where $\overline{\mathbf{x}}_i' = [x_i', y_i', w_i']^T$ and $\overline{\mathbf{x}}_i = [x_i, y_i, w_i]^T$ are corresponding homogeneous points in the two images, can be written explicitly as

$$\mathbf{A}_i^T\mathbf{f} = [x_i'x_i, x_i'y_i, x_i', y_i'x_i, y_i'y_i, y_i', x_i, y_i, 1]\begin{bmatrix}\mathbf{f}^1\\ \mathbf{f}^2\\ \mathbf{f}^2\end{bmatrix} = 0 \tag{7.28}$$

where

$$\mathrm{F} = \begin{bmatrix}\mathbf{f}^{1T}\\ \mathbf{f}^{2T}\\ \mathbf{f}^{3T}\end{bmatrix} \tag{7.29}$$

By stacking at least 8 point correspondences in the matrix $\mathrm{A} = [\mathbf{A}_1^T, \ldots, \mathbf{A}_n^T]^T$, with $n \geq 8$ because the matrix F is known up to a scale factor, an homogeneous system $\mathrm{A}\mathbf{f} = 0$ is derived whose solution can be computed by using the DLT algorithm (see Sec. 7.3.1). In particular the vector $\mathbf{f}$ is the right singular vector corresponding to the smallest singular value of $\mathbf{A}$.

The fundamental matrix is singular (see Sec. 7.2.4) but the solution found by the eight point algorithm not. It can be proved [70] that in the *Frobenious norm* the singular matrix close to a full-rank matrix can be obtained by replacing in its SVD decomposition the smallest singular value with 0. So, in order to reinforce the singularity constraint on the fundamental matrix F in the last step of the eight point algorithm the smallest singular value of the SVD decomposition of F is replaced by 0. As in Sec. 7.3.1, a normalization step on the two sets of point in the images should be applied in order to obtain a robust estimation of the fundamental matrix [70].

7.3.4 Using less than eight points

Less than 8 points can be used to solve the homogeneous system $\mathbf{A}\mathbf{f} = 0$, which corresponds to find more than one plausible fundamental matrix [70]. In particular if 7 points are used [70], the homogeneous system solution is given by a linear combination of the two basis vectors F_1, F_2 of the right null space of A, i.e. it is of the form $\mathrm{F} = \alpha\mathrm{F}_1 + (1-\alpha)\mathrm{F}_2$. The matrix F is singular so it is possible to impose the constraint $\det(\mathrm{F}) = \det(\alpha\mathrm{F}_1 + (1-\alpha)\mathrm{F}_2) = 0$ which yields to a cubic polynomial equation in α. Solving this equation leads to find one or three real solutions which can be substituted to α to get one or three fundamental matrices. In a similar way six or five point correspondences can be used by imposing more constrains which yield to respectively a maximum of 6 or 10 solutions [127, 138].

7.3.5 Other non linear methods

The algebraic minimization algorithm [70] and the non linear iterative minimization schemes to minimize the geometric or the Sampson errors [70] (see Sec. 7.3.2) can also be used. In particular the three-dimensional point coordinates have to be obtained by triangulation, using camera matrices compatible with the fundamental matrix, to estimate the geometric error. This makes the cost of the solution computationally expensive [70]. More feasible solutions can be found by using the Sampson error which explicitly is [70]

$$\mathcal{D}_{Samp}(\overline{\mathbf{x}}_i'\mathrm{F}\overline{\mathbf{x}}_i) = \frac{(\overline{\mathbf{x}}_i'\mathrm{F}\overline{\mathbf{x}}_i)^2}{(\mathrm{F}\overline{\mathbf{x}}_i)_1^2 + (\mathrm{F}\overline{\mathbf{x}}_i)_2^2 + (\mathrm{F}^T\overline{\mathbf{x}}_i')_1^2 + (\mathrm{F}^T\overline{\mathbf{x}}_i')_2^2} \tag{7.30}$$

where $(\mathbf{a})_k$ means the k-th element of the vector $\mathbf{a}$. The *symmetric epipolar error* [70]

$$\mathcal{D}_{ep}(\overline{\mathbf{x}}_i'\mathrm{F}\overline{\mathbf{x}}_i) = \mathcal{D}_{el}(\overline{\mathbf{x}}_i', \mathrm{F}\overline{\mathbf{x}}_i) + \mathcal{D}_{el}(\overline{\mathbf{x}}_i, \mathrm{F}^T\overline{\mathbf{x}}_i') \tag{7.31}$$

where $\mathcal{D}_{el}(\overline{\mathbf{x}}_i', \mathrm{F}\overline{\mathbf{x}}_i)$ is the squared distance of the point to the corresponding epipolar line

$$\mathcal{D}_{el}(\overline{\mathbf{x}}_i', \mathrm{F}\overline{\mathbf{x}}_i) = \frac{(\overline{\mathbf{x}}_i'\mathrm{F}\overline{\mathbf{x}}_i)^2}{(\mathrm{F}\overline{\mathbf{x}}_i)_1^2 + (\mathrm{F}\overline{\mathbf{x}}_i)_2^2} \tag{7.32}$$

has been reported not to give good results, through it is similar to the Sampson error [70].

The same considerations done in the case of the plane homography estimation (see Sec.7.3.1) hold. The 8 point algorithm is usually enough precise for common applications [70], moreover the algebraic minimization algorithm and non-linear iterative minimization schemes require an initial estimation of the fundamental matrix and they are more computational expensive.

7.4 The RANSAC approach

The simply matching strategies described in Sec. 5.3, generate a large number of *outliers*. Several algorithms based on *robust statistic* have been developed to find the best set of *inliers* while computing the fundamental matrix (or the plane homography in the case of planar scenes), most of them based on the RANSAC (RANdom SAmple Consensus) approach [51].

Given the candidate matching correspondences $(\mathbf{x}, \mathbf{x}')$, RANSAC starts by computing a model $\mathcal{M}$ by using a minimum hypothesis set, i.e. four point correspondences for a plane homography (see Sec. 7.3.1) and eight or less for the estimation of the fundamental matrix (see Secs. 7.3.3–7.3.4). The model $\mathcal{M}$, obtained by a random sampling of the candidate correspondences, is validated using a robust error measure $\rho_{\mathcal{M}}(\mathbf{x}, \mathbf{x}')$ on the whole set of candidate matches.

The error cost function used by RANSAC to validate the model is given by the cardinality of the inlier set $\mathcal{M}$ according to a threshold value th, i.e. the sum over all matches of the function ρ

$$\rho_{\mathcal{M}}(\mathbf{x}, \mathbf{x}') = \begin{cases} 1 \text{ if } \mathcal{D}_{\mathcal{M}}(\mathbf{x}, \mathbf{x}') \leq th \\ 0 \text{ otherwise} \end{cases} \tag{7.33}$$

where $\mathcal{D}$ is an error measure such as one seen in Sec. 7.3.2. Since the model is estimated by a minimum number of matches, the probability to be contaminated by outliers is low and the model check is relatively fast.

In the standard RANSAC approach, the putative matching correspondences are usually obtained by taking for each feature point in the first image its nearest neighbour in the second image according to the similarity of their descriptors [112]. It is a common practice introduced by [100] to score these matches using the nearest neighbour ratio (see Sec. 5.3) and then by applying a threshold equal to 0.8, in order to limit the number of wrong matches.

Moreover, input correspondences are usually limited so that there are no matches which share the same feature point, because it can lead to a degenerate model estimation. This can be accomplished by removing correspondences for points that are not mutual nearest neighbours [128], i.e. $\mathbf{x}$ should be the nearest neighbour of $\mathbf{x}'$ and vice versa for a match $(\mathbf{x}, \mathbf{x}')$. Otherwise the similarity score function $\mathcal{D}$ for match features can be made symmetric, such as [109]

$$\mathcal{D}^{\star}(\mathbf{x}, \mathbf{x}') = \frac{\mathcal{D}(\mathbf{x}, \mathbf{x}') + \mathcal{D}(\mathbf{x}', \mathbf{x})}{2} \tag{7.34}$$

However is was noted [189] that in general the nearest neighbour of a point is not the true corresponding point, which usually lies in the first k nearest neighbours (see fig. 7.7). To deal with this issue, generalizations of the RANSAC for relaxed correspondences have been proposed [189, 16], where the data input contains multiple candidate matches for each feature.

In particular in [189] the generalized RANSAC builds the hypothesis model by sampling points on the first image and then to complete the match it chooses the corresponding point on the second image from the first k nearest neighbours according to a given probability distribution. Clearly, a check is done to avoid that the last sampled correspondence contains no points that were already present in the model. A similar approach can be found in [16].

The number of iterations which the RANSAC should do is estimated by considering the estimated fraction of inliers ξ, given a model $\mathcal{M}$ of cardinality m [70]. Since the samples are drawn independently the probability to get a corrupted model, i.e. with at least an outlier, is $1 - \xi^m$. The probability to obtain at least an outlier free model in N trial is $P = 1 - (1 - \xi^m)^N$. An outlier free model with a confidence given by P, for instance $P = 0.99$ should be found in N iteration, thus

$$N = \frac{\log(1 - p)}{\log(1 - \xi^m)} \tag{7.35}$$

Since the inlier fraction ξ is usually not known a priori, RANSAC can start by imposing an initial value for ξ and by determining adaptively the inlier fraction by updating the value of ξ with the fraction of inliers of the best model obtained so far. A check is performed to test if the number of iterations done is greater that the current estimate of N. [70]. The number of iterations to be done is dependent from the fraction of inliers and on the model size. For complex model or when the number of correct matches is low, RANSAC may become prohibitive [70]. Nowadays, a model obtained by 7 or 5 correspondences is used to reduce the number of iterations for the estimation of the fundamental matrix [128, 138]. In this way, one or more matrices for a single hypothesis set (see Sec. 7.3.4) are obtained and validated on the data.

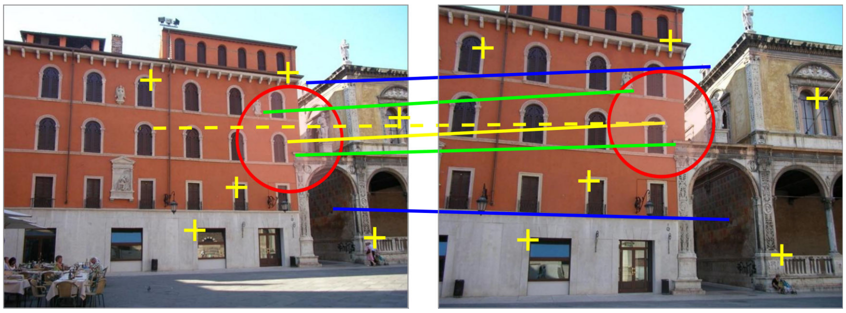


Figure 7.7: When a dominant plane is present, identified in this example by the building facades, almost all matches (yellow crosses) lie on it, leading to a wrong estimation of the fundamental matrix. A better estimation can be obtained by using off-of-the-plane matches (blue segments). Moreover, if repeated patterns are present, the first nearest neighbour match could not be correct (yellow solid and dotted segments) and the consistency given by closed matched pairs (green segments in the red circle) can be used to disambiguate the match

7.4.1 Improving the cost function

Other robust cost functions have been proposed [169, 170]. MSAC (M-estimator SAmple Consensus) [170] uses the *Huber measure* [142]

$$\tau_{\mathcal{M}}(\mathbf{x},\mathbf{x}') = \begin{cases} (\mathcal{D}_{\mathcal{M}}(\mathbf{x},\mathbf{x}'))^2 & \text{if } (\mathcal{D}_{\mathcal{M}}(\mathbf{x},\mathbf{x}'))^2 \leq th^2 \\ th^2 & \text{otherwise} \end{cases} \tag{7.36}$$

while MLESAC (Maximum Likehood Estimation SAmple Consensus) [170] tries to maximize the likelihood of the model. The error distribution is modelled by a mixture of the inlier error distribution and the outlier error distribution. The former is modelled as a Gaussian distribution g_σ with zero mean (see eq. 1.11) and the latter as an uniform distribution in $\langle 0, Z\rangle$. The error function becomes

$$\psi_{\mathcal{M}}(\mathbf{x},\mathbf{x}') = \kappa\frac{1}{\sqrt{2\pi}\sigma}e^{-\frac{(\mathcal{D}_{\mathcal{M}}(\mathbf{x},\mathbf{x}'))^2}{2\sigma^2}} + (1-\kappa)\frac{1}{Z} \tag{7.37}$$

where the mixing parameter κ is estimated by using the *Expectation Maximization* (EM) [71]. AMLESAC (Adaptive MLESAC) [84] is a noise adaptive variant of MLESAC estimator, which simultaneously estimates the mixing parameter κ and the inlier noise level σ.

7.4.2 Improving the sampling strategy

Different sampling strategies have been proposed in order to reduce the running time and to improve the inlier estimation [123, 105, 135, 110, 16, 124, 144, 104, 106, 29, 57].

NAPSAC (N Adjacent Points SAmple Consensus) [123], under the assumption that the inlier correspondences $(\mathbf{x}, \mathbf{x}')$ tend to be close in their four-dimensional space, modifies the sampling strategy. In particular, after the random selection of the first candidate match, the other candidates are chosen within a hypersphere centred in the first putative match with radius r, according to their distance from the centre.

Through in RANSAC it is assumed that a model obtained by an uncontaminated sample is consistent with all inliers, this is not the case in practice [26]. However, by observing that a good model tends to find a significant fraction of inliers, the set of putative inliers so found can be used to achieve a better model estimation. LO-RANSAC (Local Optimization RANSAC) [28], runs an optimization scheme every time a new better model is found. In particular a nested inner RANSAC can be performed, by sampling further models from the inlier set of the best model found and by verifying them against all candidate correspondences [28].

In a similar way Cov-RANSAC [136] computes the covariance error matrix of the best model estimated so far and propagates the model uncertainty to validate the other matches [70]. This allows to incorporate other possible inliers which were excluded due to the noisy of the data (see fig. 7.8). An inner RANSAC is then performed on the obtained set of putative inliers [136].

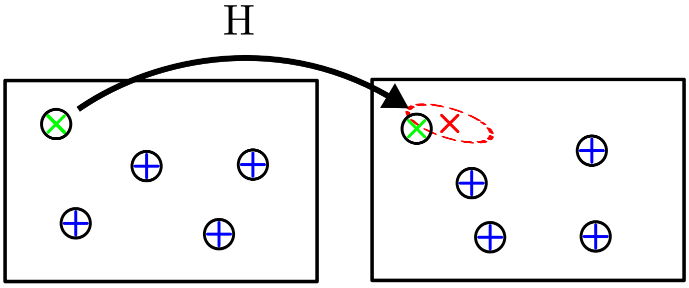


Figure 7.8: An example of the Cov-RANSAC method in the case of the homography estimation. The homography is estimated in the left image from four points (blue crosses), where the circle around them represent the error uncertainty. The error estimation is propagated so that for further points (green cross) the uncertainty of the reprojection (red ellipse) is taken into account. Image from [136]

Observing that a model which produces an high number of outliers contains points which are less probably inliers, the BaySAC [16] reduces their probability to be drawn again in the sampling by using the Bayes' rule. In a similar way SimSAC [16] simulates a status vector of inlier/outliers for all the data according to a prior probability. If the outlier outcomes are compatible with the previous failed hypothesized models, the frequency histograms for each data selected as inlier in the simulation is incremented. After T simulations compatible with the old hypotheses, the new model is formed by the first m peaks of the frequency histogram.

PROgressive SAmple Consensus (PROSAC) [105], assumes that the similarity information of the descriptors provides a good priory estimate of the correctness of the match. The algorithm starts by selecting models formed by high rank matches, because they can speed up the research for a good model. As the number of iterations increases without find a good model, PROSAC modifies its behaviour towards a random sampling because there is an evidence that the assumption done on the descriptor ranking was not correct.

Similar to PROSAC, BetaSAC [110], selects the model sample by preferring high rank matches first. However, by using the beta distribution [61] it offers a selection conditional on the previous data selected for the hypothesis set [110]. BetaSAC converges towards RANSAC in the worst case, moreover it behaves as PROSAC by an appropriate parameter setting.

Assuming that there exists some grouping in the data, obtained for instance by clustering on the optical flow or by image segmentation [66], the GroupSAC [124] drives the sampling process so that the model samples are drawn from different data groups (see fig. 7.9). Defining a configuration as a set of groups, the sampling process starts by exploring the configurations in increasing order of their cardinality, such that each group at least contributes to one data point for a configuration. Among the configurations with the same cardinality, those for which the sum of the cardinality of their groups is higher are preferred, since there is a clue of the data consistency [124].

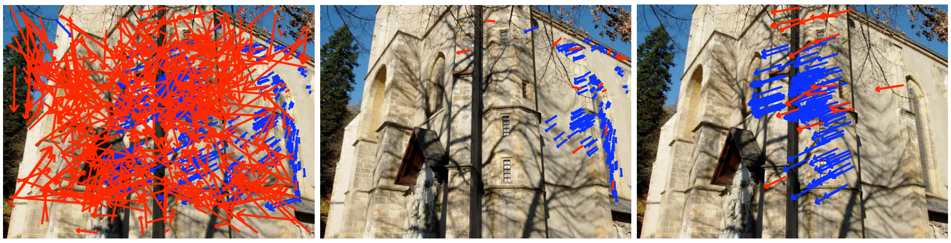


Figure 7.9: Candidate matches (left) with highlighted inliers (blue arrows) and outliers (red arrows). Clustering by the optical flow and taking the first two relevant classes (center, right), it can be noted that almost all elements in the classes are inliers. Images from [124]

SCRAMSAC [144] builds a reduced set of candidate matches on a *Spatial Consis-*

tency Check (SCC) which is used to draw the hypothesis models. For each point which composes a correspondence, a local circular neighbourhood $\mathcal{N}$ of points with similar scale factor is defined where the radius is determined in relation to the the scale of the feature point [144] (see fig. 7.7). The Spatial Consistency Check for a matched pair $(\mathbf{x}, \mathbf{x}')$ is given according to the ratio of the cardinality of the set of correspondences $(\mathbf{x}_i, \mathbf{x}'_j)$ which are in a local neighbourhood of both the points which compose the match, i.e. $\mathbf{x}_i \in \mathcal{N}(\mathbf{x})$ and $\mathbf{x}'_j \in \mathcal{N}(\mathbf{x}')$, divided by the cardinality of the set of the correspondences which are only in the local neighbourhood of one point, i.e $\mathbf{x}_i \in \mathcal{N}(\mathbf{x})$.

7.4.3 Degenerate configurations

Degenerate model configuration may arise in the sampling step in the case of the fundamental matrix estimation if the sampled points are all compatible with a plane homography. This can lead to a wrong estimation of the fundamental matrix, because the null space of the homogeneous system 1.3.1 has a dimension greater than one, which implies that multiple fundamental matrices are compatible with the model. This case is likely to happen when a dominant plane is present in the image (see fig. 7.7), and the points in the hypothesis model are usually sampled from this plane. This may lead to estimate wrongly other inliers which do not lie on the plane as outliers and to obtain a wrong fundamental matrix.

For plane homography estimation the degenerate configuration is provided when the sampled points are collinear. Clearly this is also a degenerate case for the fundamental matrix estimation.

For completeness, if both the camera optical centres and the three-dimensional points which define the correspondences lie on the *critical surface* [70], the fundamental matrix is also degenerate. It is also the case when using only 7 correspondences which is easily handled as described in Sec. 7.3.4.

In [168] the Geometric Robust Information Criterion (GRIC) score is proposed to select the best model

$$GRIC = \sum \rho(e_i^2) + 2dn + 2k \tag{7.38}$$

where n is the number of data points. The parameter d is the dimension of the constrains, with $d = 3$ for the fundamental matrix and $d = 2$ for the plane homography because in the former case all the corresponding three-dimensional points lie in a three-dimensional space, while in the latter on a plane surface [171]. The parameter k is the number of model parameters, where $k = 7$ for the fundamental matrix due to the scale and singularity constrains, and 8 for the plane homography due to scale constrain only [168]. The robust error function ρ on the error measure e_i for the i-th data is defined as

$$\rho(e^2) = \begin{cases} \dfrac{e^2}{\sigma^2} & \text{if } \dfrac{e^2}{\sigma^2} < 2(r-d) \\ 2(r-d) & \text{otherwise} \end{cases} \tag{7.39}$$

where r is the data dimension, with $r = 4$ in both the case of the fundamental matrix and of the plane homography [168].

Later in [29] DEGENSAC was proposed to improve the RANSAC estimation of the fundamental matrix in the case of a dominant plane. DEGENSAC also returns the best plane homography found in the case of degenerate configuration. Whenever a new better model is found, i.e. a model with a greater inlier support set, a test is performed to see if there exist five correspondences related by a homography. Using the same notation of Sec. 7.2.2, given three points correspondences $(\mathbf{x}_i, \mathbf{x}'_i)$, $i = 1, 2, 3$, and the fundamental matrix F, the related homography H is [70]

$$\mathrm{H} = \mathrm{A} - \overline{\mathbf{e}}_2(\mathrm{M}^{-1}\mathbf{b})^T \tag{7.40}$$

where $\mathrm{A} = [\overline{\mathbf{e}}_2]_\times \mathrm{F}$, $\mathrm{M}^T = [\overline{\mathbf{x}}_1, \overline{\mathbf{x}}_2, \overline{\mathbf{x}}_3]^T$ and $\mathbf{b} = [b_1, b_2, b_3]^T$ with

$$b_i = \left(\overline{\mathbf{x}}'_i \times (\mathrm{A}\overline{\mathbf{x}}_i)\right)^T \left(\overline{\mathbf{x}}'_i \times \overline{\mathbf{e}}_2\right) \parallel \overline{\mathbf{x}}'_i \times \overline{\mathbf{e}}_2 \parallel^{-2} \tag{7.41}$$

The possible homographies are computed by using the fundamental matrix F and three correspondences from the five taken from the sampled model [29]. It can be shown that at most five different homographies have to be tested [29]. If the test finds a compatible homography with a greater support set than the last best homography model validated, it is stored. Moreover, using the *plane-and-parallax algorithm* [70] which requires a homography and only two matches, an inner RANSAC is run, paying attention to not select correspondences on the estimated plane of H, i.e. leaving out the homography inliers.

Lastly, QDEGSAC [57], a RANSAC for (Quasi-)Degenerate data, has been presented, which provides a general framework that can be used not only in the case of fundamental matrix estimation [57]. Since in the case of degenerate data the homogeneous system associated with the model estimation (see Sec. 7.3.3) has a null space of dimension greater than one [70], a robust measurement on the rank of the data matrix provided by the inliers can be used to detect degeneracies. After a first RANSAC, successive RANSACs are performed on the inlier support set of the previous one until their ratio is less then a predefined threshold. Each of these RANSAC steps look for degenerate inliers which satisfy the constraints of the model for increasing dimensions of the null space. When the final set of degenerate inliers is found, the correct model which completes the model obtained by the degenerate points is retrieved by running a RANSAC on the outliers set.

The described methodologies have been incorporated to other RANSAC approaches [144, 135, 124].

7.4.4 Faster model check verification

Different strategies have been developed to reduce the time to verify the models.

In [104] the $T_{d,d}$ test is performed, where the test is passed if the evaluation model is consistent with d random selected correspondences. Models which pass the test are then evaluated on the whole set of data. Thought the number of model evaluated increases since good models can be rejected by mistake, the total cost of the method is reduced. This happens because the number of total verification steps is reduced, especially for the value $d = 1$ suggested by the authors [104].

In a similar way the Bail-out test [22] breaks the evaluation of the current model if during the test of the i-th candidate match an approximate probability of estimate a better model is low. The WaldSAC strategy [106] avoids the full validation of the dataset if the *Wald's Sequential Probability Ratio Test* (SPRT) [183] is not satisfied.

The $T_{d,d}$ test and SPRT have been incorporated in many RANSAC based approaches to speed-up the process [135, 144, 124].

The Preemptive RANSAC [126] was designed for real time applications. A fixed number of hypotheses is generated and evaluated on a subset of the data points. The hypotheses are then reordered based on the results of the scoring procedure and only a fraction of the hypotheses are evaluated on the next subset of data until one hypothesis is left.

The Adaptive Real-Time Random SAmple Consensus (ARRSAC) [135] extends the Preemptive RANSAC scheme by generating new hypotheses after retaining a fraction of the evaluated hypotheses on a block of data. The number of the new hypotheses to generate is computed by an estimation of the inlier ratio based on the best ranked hypothesis so far. Clearly the new hypotheses have to be also validated on the previous blocks of data.

7.4.5 Other approaches

Similar to the RANSAC, the LMedS (Least Median Square) algorithm [190, 142], searches for the best model which minimized the median value of the squared error distance $\mathcal{D}$ over the correspondence set, under the assumption that $\xi \geq 0.5$ [26], while the MINPRAN algorithm selects the model that MINimizes the Probability of RANdomness of the solution [158]. The RANSAC method has been also combined with the Hough transform [66] to estimate the fundamental matrix. This allows a fast model estimation under the presence of outliers, obtained by sampling a smaller than the minimal subset, followed by a voting process of the remaining data [39]. Also genetic algorithms [117] have been employed in GANSAC [139], where a gene codifies the inlier/outlier statuses of the data and the usual operations of crossover and mutation are applied [117]. The pbM-estimator (projection based M-estimator) algorithm [159] instead reframes the regression problem of the best model estimation into a projection pursuit framework [25].

Chapter 8

A new sparse soft matching algorithm

8.1 Overview

In the following a matching algorithm based on RANSAC (see Sec. 7.4) is described. Its main features are a image-guided selection of the error threshold and a *soft matching* strategy in contrast to the one-to-one matching usually adopted by RANSAC (see Sec. 7.4), which increases the absolute matches. Moreover, the sampling process (see Sec. 7.4.2) is guided by a global-to-local constraint generation. The final inlier matches are homogeneously better distributed along the image, resulting in a more stable estimation of the homography or the fundamental matrix associated to the stereo pair. As a weak point, it is computationally more expensive than RANSAC and it considerably relies on the descriptor similarity.

An evaluation framework similar to that proposed by Moreels and Perona in [119] is also presented. It is pure geometrical, i.e. it only uses the feature position, not any further information provided by the feature detector such as the scale, the orientation and the shape of the patch. Moreover, it allows a slight computation of the ground truth data.

8.2 Algorithm description

8.2.1 The main idea

The main idea of the method resembles the simulated annealing process [14]. Different RANSACs are repeated as in the inner RANSAC scheme (see Sec. 7.4.2). At each RANSAC run the set $\mathcal{S}$ of putative matches from where the candidate model $\mathcal{M}$ is sampled and the set $\mathcal{T} \supset \mathcal{S}$ used to validate the model, increase their cardinalities. New candidate matches are added to $\mathcal{S}$ and $\mathcal{T}$ according to the increment of the percentage of the estimated inliers, the rank of the matches provided by the feature descriptor similarities and to a resolution factor l. At the same time, the error threshold th used to validate the inliers decreases to obtain a finer model at every RANSAC.

If a model $\mathcal{M}$, represented by the set of matches $\mathcal{I}$ which agree with it, found in the current RANSAC run is better than the best model $\overline{\mathcal{I}}$ found in all previous runs, both $\mathcal{S}$ and $\mathcal{T}$ grow slowly. On the other hand, $|\mathcal{S}|$ and $|\mathcal{T}|$ increase faster because the use of the similarity to rank the matches does not give a good clues on the true inliers and a random selection should achieve a better result. Moreover, the probability of sampling a correspondence, represented by a table $\mathcal{W}$, is updated by considering the history of the previous RANSAC runs.

The sampling strategy is similar to that proposed by other guided sampling approaches (see Sec. 7.4.2), however the selection also depends on the image resolution as will be described next. The threshold error th, related to the current resolution, makes the process similar to the inner RANSAC scheme with iterations [28], where nested RANSACs are repeated on smaller inliers subset, by decreasing a threshold value. However, in the new proposed method, the validation set is not provided by the initial input set of all matches $\mathcal{P}$, but it varies with the iterations. Moreover, the process provides sets of increasing cardinality which incorporate the previous ones in contrast to nested subsets.

As other characteristics, soft matches are used, i.e. pairs can share a common point, as done for some RANSAC based approaches (see Sec. 7.4), but differently from other existent methodologies, to estimate the model $\mathcal{M}$ the sampling tries to force points which are distant from each others. It allows a better estimation (see fig. 8.1), though RANSAC-based approaches where matches are chosen close to each others have been used successfully (see Sec. 7.4.2).

Before an effective description of the algorithm, the selection of the candidate matches, the sampling strategy and the model validation are described in more details in the next subsections.

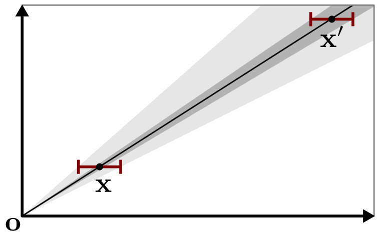


Figure 8.1: Given the same measurement error (red line), a better estimation on the line through $\mathbf{o}$ is obtained if a far points $\mathbf{x}'$ is used (dark gray cone) instead of a close point $\mathbf{x}$ (light gray cone)

8.2.2 Putative matches selection

The selection strategy is based on subsets $\mathcal{U}_l$ of the set $\mathcal{P} = \{(\mathbf{x}^1, \mathbf{x}^2)\}$ of the initial input candidate matches, where $\mathbf{x}^1$ and $\mathbf{x}^2$ are respectively in the first and in the second image of the stereo pair I_i, $i = \{1, 2\}$. The set $\mathcal{U}_l$ depends on the scale resolution parameter l. Each image of the stereo pair, respectively of size $m_i \times n_i$, is divided into a squared grid where the stride is $s_i = \min(m_i, n_i)/l$.

A putative match $(\mathbf{x}^1, \mathbf{x}^2)$ belongs to the k-th block in the i-th image $\mathcal{B}^i_{lk}$ for a scale parameter l if the respective point $\mathbf{x}^i$ of the pair is inside the region delimited by that block.

For each block $\mathcal{B}^i_{lk}$, the best putative match $p^i_{lk} \in \mathcal{P}$ inside the block is chosen according to the similarity $\mathcal{D}$ of the corresponding descriptors (see Chapter 4)

$$p^i_{lk} = \underset{(\mathbf{x}^1,\mathbf{x}^2)\in\mathcal{B}^i_{lk}}{\arg\min}\ \mathcal{D}\left(\mathbf{x}^1,\mathbf{x}^2\right) \tag{8.1}$$

Clearly, $\{p^1_{lk}\} \neq \{p^2_{lk}\}$ in general for the two sets $\{p^1_{lk}\}$,$\{p^2_{lk}\}$, because the number of blocks in the two images are not equal and the elements of a match pair cannot be mutually neighbours (see fig. 7.4). The union of the two sets

$$\mathcal{U}_l = \{p^1_{lk}\} \cup \{p^2_{lk}\} \tag{8.2}$$

gives the final set $\mathcal{U}_l$ from which to sample the model (see fig. 8.2). As other methods described in Sec. 2.2.5 it allows a better distribution of the points on the images and provides *soft matches*, in the sense that there can be pairs which share one element. This can alleviate the issue of a bad input set of matches which can occur when pairs of candidate matches are not allowed to share an element.

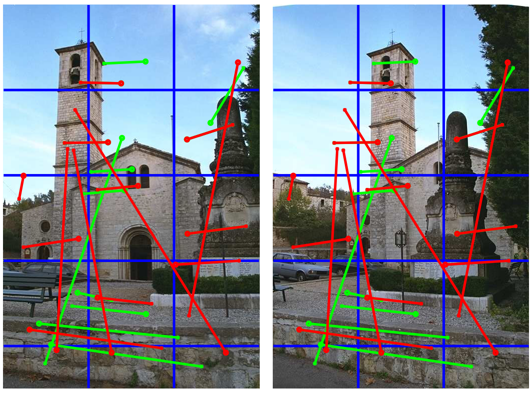


Figure 8.2: The set $\mathcal{U}_3$ of the correspondences, superimposed on the stereo image pair I_1, I_2 (left, right image). In particular the red matches represent the set $\{p^1_{3k}\}$, while the green one gives the set $\{p^2_{3k}\}$

8.2.3 Model sampling

The seven or four matches from which to build the model $\mathcal{M}$, respectively in the case of a plane homography or in the case of the fundamental matrix, are obtained from a set $\mathcal{S} = \{s_j\}$ of the candidate inliers as follows. Let $\mathcal{T}$ be the set used to validate the model $\mathcal{M}$, $|\mathcal{S}| = n$, so that $\mathcal{S} \subset \mathcal{T} \subset \mathcal{P}$.

Matched pairs $\overline{s}_k \in \mathcal{S}$, where $k = 4$ or $k = 7$ respectively in the case of a plane homography or the fundamental matrix, are sampled in order, according to a probability distribution p_{kj} associated to the pair $\overline{s}_j$ at the selection of the k-th pair. The probability p_{kj} is updated to $p_{(k+1)j}$ after the k-th sampled pair is selected.

The probability p_{1j} associated to $s_j = \left(\mathbf{x}^1, \mathbf{x}^2\right)$ for the first sampled pair is

$$p_{1j} = \frac{w(s_j)d(s_j)q(s_j)z(s_j)}{b_1} \tag{8.3}$$

It is the product of different weights. The term b_1 is chosen so that $\sum_{j=1}^{n} p_{1j} = 1$, the weight $w(s_j)$ indicates the goodness of the pair over the RANSAC history and it will be discussed later.

The weight $d(s_j)$ defines the *inner similarity* of the descriptor associated with $\mathbf{x}^i$, i.e. the uniqueness in the respective images of the features associated with the pair. In particular given the ratio

$$A_v^i(s_j) = \frac{\mathcal{D}\left(\mathbf{x}^i, \widehat{\mathbf{x}}_2^i\right)}{\mathcal{D}\left(\mathbf{x}^i, \widehat{\mathbf{x}}_v^i\right)} \tag{8.4}$$

where $\widehat{\mathbf{x}}_r^i$ is the r-th nearest neighbour of $\mathbf{x}^i$ among all features detected in I_i and v is a fixed parameter, the inner similarity is

$$d(s_j) = 1 - A_v^1(s_j)A_v^2(s_j) \tag{8.5}$$

Pairs for which $d(s_j)$ is very close to 1, are formed of features which correspond to repeated patterns in the images, so they are more ambiguous than others (see fig. 8.3). The measure can be seen as an extension of the nearest neighbour ratio (see Sec. 5.1.5), however by varying v it allows to take into the account the fact that the same feature is often selected at different but very close scales.

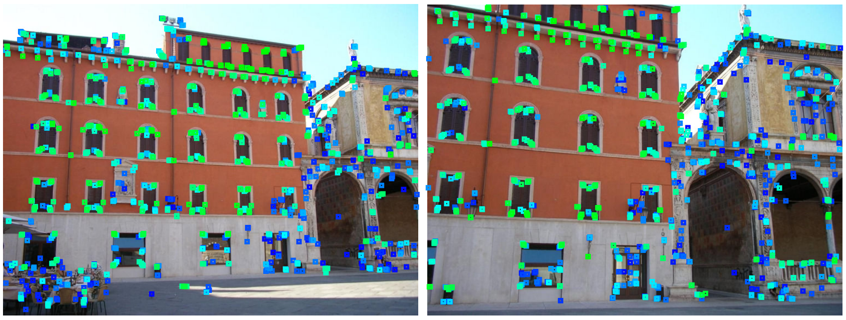


Figure 8.3: A stereo pair with superimposed detected features. The colour of the features s_j vary from green to blue as $d(s_j)$ decreases toward zero

The weight $z(s_j)$ defines the *cross similarity* of the pair in the set $\mathcal{T}$, i.e. how many matches in $\mathcal{T}$ share common points,

$$z(s_j) = \frac{1}{\sqrt{B^1(\mathbf{x}^1)B^2(\mathbf{x}^2)}} \tag{8.6}$$

where $B^j(\mathbf{x}^i)$ is the number of pairs in $\mathcal{T}$ which share the same element $\mathbf{x}^i$. The cross similarity also indicates how ambiguous is the match, but in contrast to the inner similarity it works on the relationship between the two images and not inside of them.

The last weight $q(s_j)$ defines a *consistency similarity* and indicates the geometric consistency of the pair. If $\mathcal{C}_r^i(\mathbf{x}^i)$ is the set of match pairs where the corresponding element in I_i has a distance less than r from $\mathbf{x}^i$ and $r = th$

$$q(s_j) = \frac{\mathcal{C}_r^1(s_j) \cap \mathcal{C}_r^2(s_j)}{\mathcal{C}_r^1(s_j) \cup \mathcal{C}_r^2(s_j)} \tag{8.7}$$

This formulation is similar to SCC (see Sec. 7.4.2), however to be independent from the feature descriptor it does not take into account the scale given by the feature and it is symmetric.

After the k-th pair $\widehat{s}_k = s_j$ is sampled and added to the model $\mathcal{M}$, the probability $p_{(k+1)i}$ of the pairs in $\mathcal{C}_r^f(\widehat{s}_k)$, with $f = \mathrm{mod}(k, 2) + 1$, and of the pairs which share a point with $\widehat{s}_k$ is set to zero. The remaining values are normalized to the unit to get a new consistent probability distribution (see fig. 8.4). This strategy removes a random neighbourhood of the points $\mathcal{C}_r^f(\widehat{s}_k)$ and forces the model to be composed by distant points, though as shown by NAPSAC and GroupSAC (see Sec. 7.4.2), inliers are often close to each other. As shown in fig. 8.1 in the case of a straight line, since the estimation by close points of geometric entities subject to noise is less accurate, this strategy can enhance the selection of the inliers.

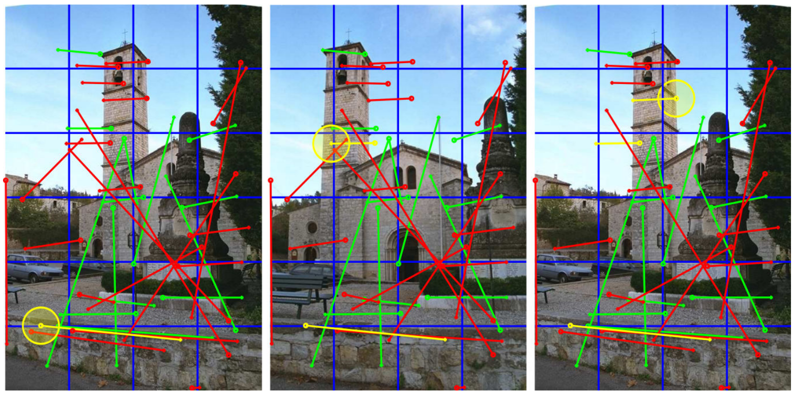


Figure 8.4: Successive steps of the model sampling for the sparse soft matching. A match is selected (yellow segment) and all matches in the first image inside the corresponding neighbour (yellow circle) are removed. The process is repeated for the next sampled matches by alternating the image for the deletion of neighbours

8.2.4 Model validation

In order to validate the model $\mathcal{M}$, the error cost used is given by the maximum between the symmetric reprojection errors. In particular in the case of the fundamental matrix F, the error cost ρ_{F} is the maximum of the distances $\mathcal{D}_{el}$ of the corresponding points in the pair $t_j = \left(\mathbf{x}^1, \mathbf{x}^2\right) \in \mathcal{T}$ from the respective epipolar lines (see Sec. 7.3.5)

$$\rho_{\mathrm{F}}(t_j) = \max\left(\mathcal{D}_{el}(\overline{\mathbf{x}}^2, \mathrm{F}\overline{\mathbf{x}}_1), \mathcal{D}_{el}(\overline{\mathbf{x}}^1, \mathrm{F}^T\overline{\mathbf{x}}^2)\right) \tag{8.8}$$

For a planar homography H, the error cost ρ_{H} uses the geometric error $\mathcal{D}_{geom}$ (see Sec. 7.3.2)

$$\rho_{\mathrm{H}}(t_j) = \max\left(\mathcal{D}_{geom}\left(\overline{\mathbf{x}}^2, \mathrm{H}\overline{\mathbf{x}}^1\right), \mathcal{D}_{geom}\left(\overline{\mathbf{x}}^1, \mathrm{H}^{-1}\overline{\mathbf{x}}^2\right)\right) \tag{8.9}$$

The use of the maximum value instead of the symmetric error (see Sec. 7.3.2) imposes a more rigid constraint on the error. Pairs for which the error is less than a threshold value th are included in the set $\mathcal{T}' = \{t'_j\}$, $j = 1, \ldots, m$ of matches which can be considered inliers. This set can contain pairs which share a same element so, in order to disambiguate the matches, the pairs t'_j are ranked according to a function $\mathcal{K}$ and the final set $\mathcal{I} = \mathcal{Q}_{\mathcal{K}}$ of hard match inliers is obtained as described in Sec. 5.1.1. The value of the function $\mathcal{K}$ for a pair $t'_j = (\mathbf{x}^1, \mathbf{x}^2)$ is

$$\mathcal{K}(t'_j) = \rho(t'_j)c(t'_j)\left(1 - d(t'_j)\right)\left(1 - q(t'_j)\right)\left(1 - z(t'_j)\right) \tag{8.10}$$

The value ρ is the error cost from ρ_{F}, ρ_{H} normalized so that the sum over the set $\mathcal{T}'$ gives the unit, $q(t'_j)$ and $d(t'_j)$ are respectively the consistency similarity and the inner similarity indexes described in Sec. 8.2.3, while $z(t'_j)$ is the cross similarity over the set $\mathcal{T}'$. Lastly, the value $c(t'_j)$ is the *descriptor similarity*, i.e. the distance computed on the feature descriptors associated to the pair, normalized on the set $\mathcal{T}'$.

An ordering relation $\mathcal{I} > \mathcal{I}'$ between inliers set $\mathcal{I}$, $\mathcal{I}'$ is defined so that $\mathcal{I} > \mathcal{I}'$ if $|\mathcal{I}| > |\mathcal{I}'|$ or, in the case their cardinality are equals, if the median value of ρ for $\mathcal{I}$ is less than that of $\mathcal{I}'$.

8.2.5 Merging all together

Algorithm 1 shows the main body of the method. The threshold value th in line 9 is computed as

$$th = \kappa\frac{\min(m_1, m_2, n_1, n_2)}{k + k_{step}} \tag{8.11}$$

where κ is a constant factor. The threshold th is a fraction of the minimum block size used to generate the validation set $\mathcal{T}$ (see Sec. 8.2.2) where κ was set to $\kappa = 1/4$ in the test, th represents the radius of a block with half size with respect to the block $\mathcal{B}^i_{lk}$. The number of iterations k'_{max} (line 16) inside a RANSAC run depends on the current outer iteration k, so that more iterations are required as the sampling set $\mathcal{S}$ and the validation set $\mathcal{T}$ grow.

In particular an empirical linear function has been used

$$k'_{max}(k) = 8 + \frac{5}{2}k \tag{8.12}$$

As an annealing process (as well as other non-linear minimization algorithms such as the Levemberg-Marquadt iteration), when the algorithm is near a good inlier set,

Algorithm 1 pseudocode for the soft sparse matching

```
// initiate the parameters
$k \leftarrow 2$
$k_{step} \leftarrow 1$
$\mathcal{W} \leftarrow \mathbf{1}$
$\mathcal{I}_{best} \leftarrow \varnothing$
// main loop
while $k < k_{max}$ do
    // set the threshold
    $th \leftarrow threshold(k + k_{step})$
    // generate the sampling set
    $\mathcal{S} \leftarrow \mathcal{U}_k$
    // generate the validation set
    $\mathcal{T} \leftarrow \mathcal{U}_k \bigcup \mathcal{U}_{k+k_{step}}$
    $flag \leftarrow 0$
    // execute the RANSAC
    for $k' \leftarrow 1$ to $k'_{max}(k)$ do
        $\mathcal{M} \leftarrow sample(\mathcal{S}, \mathcal{T}, \mathcal{W}, th)$
        $\mathcal{I} \leftarrow eval(\mathcal{S}, \mathcal{T}, th)$
        if $\mathcal{I} > \mathcal{I}_{best}$ then
            $flag \leftarrow 1$
            $\mathcal{I}_{best} \leftarrow \mathcal{I}$
        end if
    end for
    // update the parameters
    if $flag$ then
        $k_{step} \leftarrow \max(1, k_{step})$
        $\mathcal{W} \leftarrow update(\mathcal{W}, \mathcal{I})$
    else
        $k_{step} \leftarrow 2k_{step}$
    end if
    if $k_{step} > k_{maxstep}$ then
        return $\mathcal{I}_{best}$
    end if
    $k \leftarrow k + k_{step}$
end while
return $\mathcal{I}_{best}$
```

it tries to slowly increase the inlier set for a better estimation. Otherwise, it looks for a bigger, and probably random, set so an estimation between the errors of the point matches is also performed. Moreover, if the difference, given by k_{step}, between the sampling set $\mathcal{S}$ and the validation set $\mathcal{T}$ is large, the algorithm breaks because this means that the guided sampling does not perform well.

In the experiments the max allowed k_{step} is $k_{maxstep} = 8$, the rank value v used to compute the inner similarity (see Sec. 8.2.3) was set to 16, while $k_{max} = 64$.

The table $\mathcal{W}$ has an entry $w'(\mathbf{x}^1, \mathbf{x}^2)$ set to 1 at line 4 for each possible input match pairs in $\mathcal{P}$. Whenever a new inlier set $\mathcal{I}_{best}$ is found, $\mathcal{W}$ is updated at line 27 and $w'(p) = w'(p) + 2q(p)$ for each pair $p \in \mathcal{I}_{best}$, where q is the consistency similarity (see Sec. 8.2.3). The value $w(s_j)$ used in Secs. 8.2.3,8.2.4 is the entry value of $w'(s_j)$ normalized over the set $\mathcal{S}$

$$w(s_j) = \frac{w'(s_j)}{\sum\limits_{s \in \mathcal{S}} w'(s)} \tag{8.13}$$

Clearly, the soft sparse matching requires more computational time than the RANSAC approach. Example of the results are shown in figs. 8.5–8.7 in comparison with a standard RANSAC. For RANSAC, the maximum number of iterations is set to 5000 and the Sampson error is used with a threshold of 3.5 pixels.

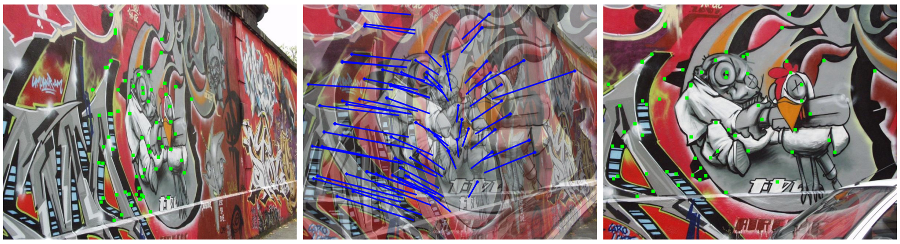


Figure 8.5: A planar homography estimation by the sparse soft matching method. Matched features (green dots) are superimposed on the stereo pair (left, right), and matches are superimposed on both merged images (centre)

8.2.6 Dominant plane and repeated patterns

A treacherous configuration involving dominant planes which cannot be handled by RANSAC (see Sec. 7.4.3), is represented by a dominant plane π with repeated patterns lying on same lines. These lines intersect the images I_i respectively in two points, finite or infinite, denoted by $\mathbf{b}_i$, as shown in fig. 8.10(top row).

The plane induces an homography H_π and this can lead to a wrong estimation of the fundamental matrix F_π when the model $\mathcal{M}$ is sampled by pairs of points lying on the dominant plane π. It happens frequently because the dominant plane extends among all the image surfaces. Moreover the epipoles are wrongly found

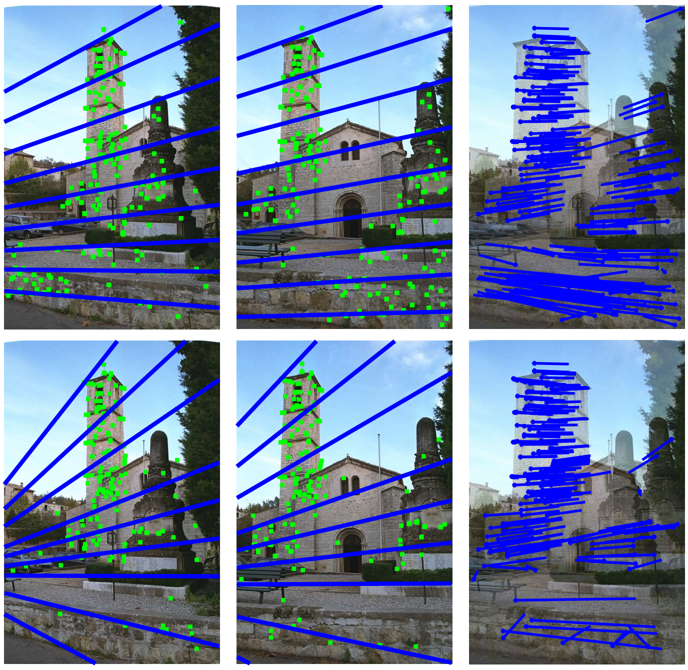


Figure 8.6: Fundamental matrix estimation by the sparse soft matching (top) and by RANSAC (bottom). Corresponding epipolar lines (blue lines) and matched features (green dots) between the stereo pair (left, central columns) and superimposed images with matches (blue segments, right). More and better distributed features are obtained with the proposed method, as well as a better estimation of the epipolar lines

into the points $\mathbf{b}_i$ and the error function ρ_{F} becomes disadvantageous, because the wrong matches are on the incorrect estimate epipolar lines (see fig. 8.10).

In order to deal with this not so infrequent configuration, the following simple strategy can be applied. First a trivial check is performed to establish the presence of a dominant plane. The sparse soft matching algorithm is applied to estimate the dominant plane π with a sample model of 4 matches, obtaining the plane homography H_π and an inliers set $\mathcal{I}_{\mathrm{H}_\pi}$, but also for estimating a fundamental matrix F by using a model of 7 pairs, obtaining an inlier set $\mathcal{I}_{\mathrm{F}}$. If $|\mathcal{I}_{\mathrm{H}_\pi}| > \alpha|\mathcal{I}_{\mathrm{F}}|$ where for instance $\alpha = 0.4$, the fundamental matrix has to be estimated again by using H_π.

In particular the set $\mathcal{P}$ of all possible pairs is updated so that the pairs in $\mathcal{P}$ which share one corresponding point with a pair in $\mathcal{I}_{\mathrm{H}_\pi}$ are removed.

The set $\mathcal{S}$ and $\mathcal{T}$ become $\mathcal{S} = \mathcal{S} \cup \mathcal{I}_{\mathrm{H}_\pi}$ and $\mathcal{T} = \mathcal{T} \cup \mathcal{I}_{\mathrm{H}_\pi}$. Moreover, the model $\mathcal{M}$ is forced to have at maximum 3 pairs from $\mathcal{I}_{\mathrm{H}_\pi}$, by setting the probability of the pairs in $\mathcal{S}$ which are in $\mathcal{I}_{\mathrm{H}_\pi}$ equal to 0 after the third pair in $\mathcal{I}_{\mathrm{H}_\pi}$ was found. Even if the plane-and-parallax algorithm [70] requires only the homography H_π and two off-on-the-plane pairs to estimate the fundamental matrix F, the proposed method forces a model estimation more independent from H_π.

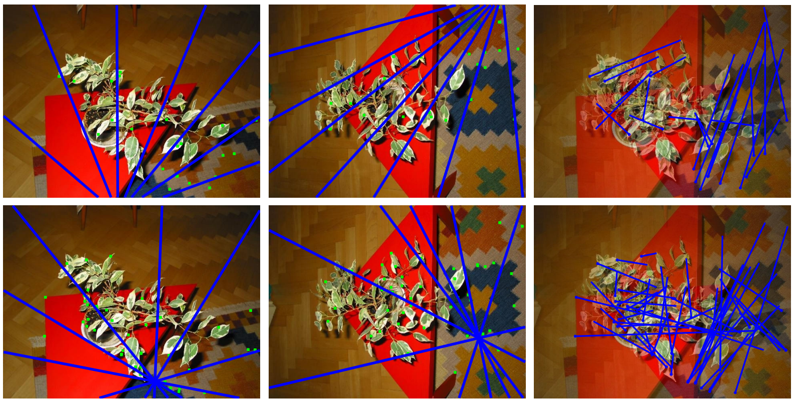


Figure 8.7: Fundamental matrix estimation by the sparse soft matching (top) and by RANSAC (bottom). Corresponding epipolar lines (blue lines) and matched features (red dots) between the stereo pair (left, central columns) and superimposed images with matches (blue segments, right). Both RANSAC and the proposed method fail to estimate the correct epipolar geometry

The entries of the table $\mathcal{W}$ for pairs $(\mathbf{x}^1, \mathbf{x}^2) \in \mathcal{I}_{\mathrm{H}_\pi}$ are updated in line 27 so that $w'(\mathbf{x}^1, \mathbf{x}^2) = w'(\mathbf{x}^1, \mathbf{x}^2) + 1$, while more insight is used in the model validation. In particular the corresponding *Voronoi diagrams* [38] in the images I_i are built for the pairs $(\mathbf{x}^1, \mathbf{x}^2) \in \mathcal{I}_{\mathrm{H}_\pi}$, and the set $\mathcal{T}'$ (see Sec. 8.2.4) is further refined so that point in the pairs $(\mathbf{x}^1, \mathbf{x}^2) \in \mathcal{T}'$ should lie in the same corresponding Voronoi regions (see fig. 8.10, central row). This is often the common case, but for very wide baseline stereo pairs it can lead to drop some inliers.

The time required to compute the correct matches, though good result are obtained, is higher. An example of the final result is shown in fig. 8.10 (bottom).

8.3 Algorithm evaluation

8.3.1 Test setup

In order to test the proposed method in the case of the estimation of the fundamental matrix, a strategy similar to that proposed in [119] was adopted (see Sec. 5.1.5). This strategy does not work for rectified images but it is only geometric and the ground truth can be estimated easily. Three images I_0, I_1, and I_2 are used, where the ground truth fundamental matrices F_{01}, F_{12} and F_{02} are estimated by taking point correspondences by hands using the 8 point algorithm (see Sec. 7.3.3). To be more fair, the correspondences has been estimated by three different people for each stereo pairs.

The check is performed as follows, given three inliers sets $\mathcal{I}_{01}$, $\mathcal{I}_{12}$ and $\mathcal{I}_{02}$ as input. The *chain set* $\mathcal{C}_{ikj}$ of quadruplets of two inlier sets $\mathcal{I}_{ik}$ and $\mathcal{I}_{kj}$ is formed so that $(\mathbf{x}^i, \mathbf{x}^k, \mathbf{x}'^k, \mathbf{x}^j) \in \mathcal{C}_{ikj}$ if two pairs $(\mathbf{x}^i, \mathbf{x}^k) \in \mathcal{I}_{ik}$ and $(\mathbf{x}'^k, \mathbf{x}^j) \in \mathcal{I}_{kj}$ exist for

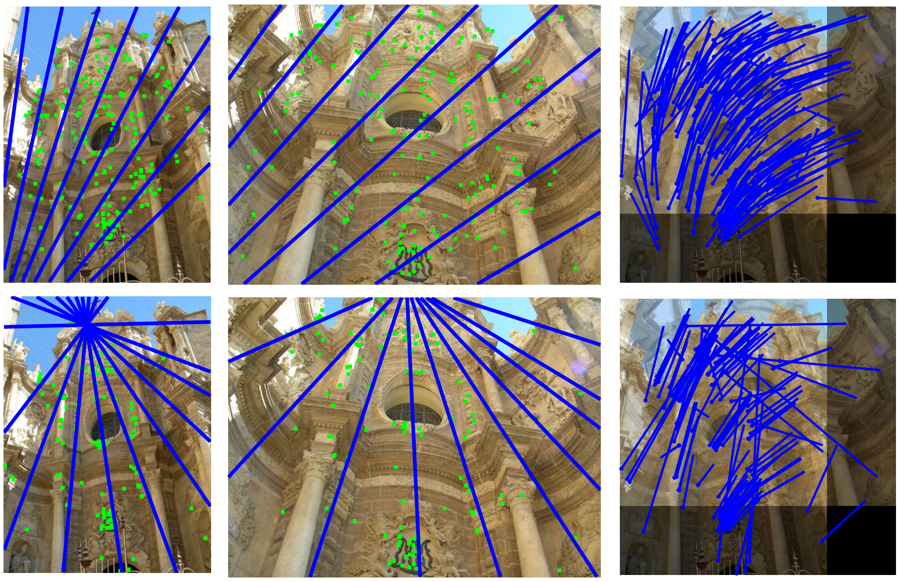


Figure 8.8: Fundamental matrix estimation by the sparse soft matching (top) and by RANSAC (bottom). Corresponding epipolar lines (blue lines) and matched features (green dots) between the stereo pair (left, central columns) and superimposed images with matches (blue segments, right). Also in this case a better feature distribution and a better fundamental matrix estimation are obtained by the sparse soft matching

which the Euclidean distance of the point locations in the same image I_k is less than a threshold th

$$L_2(\mathbf{x}^k, \mathbf{x}'^k) < th \tag{8.14}$$

Three different sets $\mathcal{C}_{012}$, $\mathcal{C}_{120}$ and $\mathcal{C}_{201}$ are obtained for the pairs $\mathcal{I}_{01}$ and $\mathcal{I}_{12}$, $\mathcal{I}_{12}$ and $\mathcal{I}_{20}$, $\mathcal{I}_{10}$ and $\mathcal{I}_{02}$ respectively, where the set $\mathcal{I}_{mn} = \mathcal{I}_{nm}$ is obtained by swapping the points inside a pair.

The chain set $\mathcal{C}_{ikj}$ is tested to check if the two pairs which form a quadruplet are inliers, obtaining two inlier sets $\mathcal{C}_{ikj}^{ik}$ and $\mathcal{C}_{ikj}^{kj}$, one for each stereo pair of the chain.

If all point composing the quadruplet are at the maximum distance th from the epipolar lines, obtained by projecting the corresponding points through the respective fundamental matrices, both the pairs are considered correct matches, i.e. if the test is positive from $(\mathbf{x}^i, \mathbf{x}^k, \mathbf{x}'^k, \mathbf{x}^j) \in \mathcal{C}_{ikj}$ it follows that $(\mathbf{x}^i, \mathbf{x}^k) \in \mathcal{C}_{ikj}^{ik}$ and $(\mathbf{x}'^k, \mathbf{x}^j) \in \mathcal{C}_{ikj}^{kj}$. More in detail, three epipolar lines correspond to $\mathbf{x}^i$ and $\mathbf{x}^j$, while only two for the points $\mathbf{x}^k$ and $\mathbf{x}'^k$, so a total of 10 point-to-epipolar-line distances are checked (see fig.8.11). Clearly, there can be some false positive, especially when the epipolar lines corresponding to a point are almost parallel, however as it was verified in the next tests and as it was noted in [119], it happens with a relatively low probability. For each initial inlier set $\mathcal{I}_{qz}$, the set of correct matches according to the proposed test is $\mathcal{G}_{qz} = \bigcup_{i,j,k} \mathcal{C}_{ijk}^{qz}$.

The main issue with this approach is that some matches can be wrongly discarded if

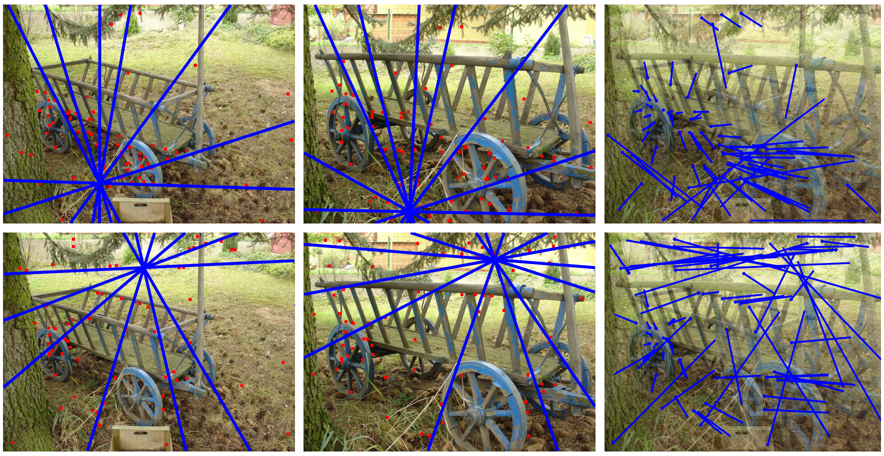


Figure 8.9: Fundamental matrix estimation by the sparse soft matching (top) and by RANSAC (bottom). Corresponding epipolar lines (blue lines) and matched features (red dots) between the stereo pair (left, central columns) and superimposed images with matches (blue segments, right). A better epipolar line estimation is obtained for the proposed method

the point is not present in all images, which can happen for high degree of transformation between images and due to a detector failure. A good measure to validate the consistency of the method is provided by the inlier percentage

$$\mathcal{Z}_{ij} = \frac{|\mathcal{I}_{ij} \bigcap \mathcal{G}_{ij}|}{|\mathcal{I}_{ij}|} \tag{8.15}$$

If the triplets of images I_i are sorted according to the image transformation degree, it can be easily seen that the stereo pair I_0, I_2 should have the high percentage of correct matches because it contains matches obtained from strongly repeated features.

Since it not easy to order the transformations in general, in the proposed framework only the inlier set $\mathcal{G}$ which has the higher inlier percentage $\mathcal{Z}_{ij}$ is used from the three possible choices $\mathcal{G}_{ij}$, i.e.

$$\mathcal{G} = \arg\max_{i,j} \mathcal{Z}_{ij} \tag{8.16}$$

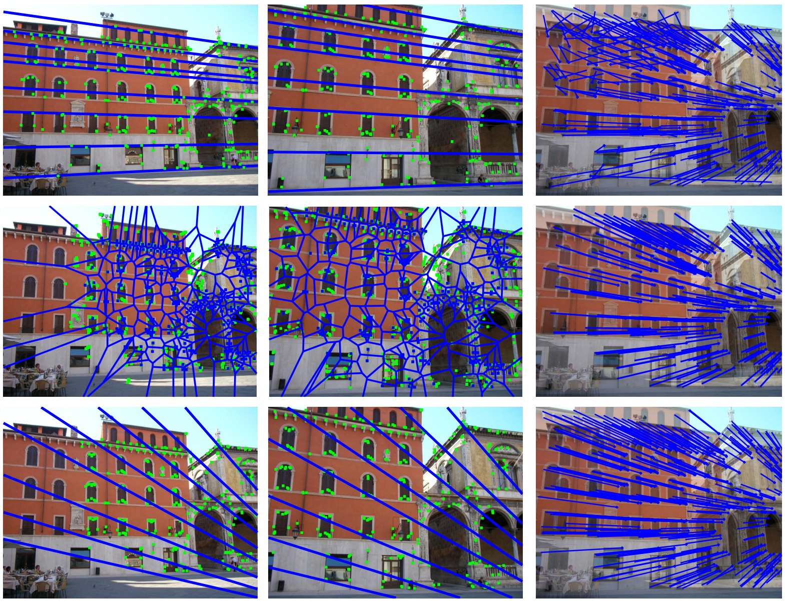


Figure 8.10: Fundamental matrix estimation without taking into account the dominant plane (top row), the stereo pair with the matched features (left, central columns) and the matches on the superimposed images (right column). The epipoles are badly located in the vertical direction on the build facades. Corresponding Voronoi cells (central row) used to locate matches (left, central columns) according to the plane homography estimation (right column). The final fundamental matrix estimation obtained by using the Voronoi cell to constrain the matches (bottom row). To note the estimation of the epipolar lines, close to the true solution

8.3.2 Experimental result

To validate the soft sparse matching, four different measures are considered

- the percentage of inliers $\mathcal{I}$;
- the absolute number of inliers $\mathcal{I}$;
- the percentage of the maximal convex hull area $\mathcal{H}$ between the two stereo images corresponding to $\mathcal{G}$;
- the percentage of the maximal area coverage $\mathcal{V}$ obtained by the relative descriptors.

The error threshold th was set to 5 pixels. In order to compare the matching strategy RANSAC has also been included in the test with different measure errors. The different methodologies have been tested on different triplets of images, reported in figs. 8.12–8.22.

In particular, RANSAC, MLESAC, MSAC and LMedS (see Sec. 7.4.1) have been tested using the Sampson error and the point-to-epipolar-line distance (see

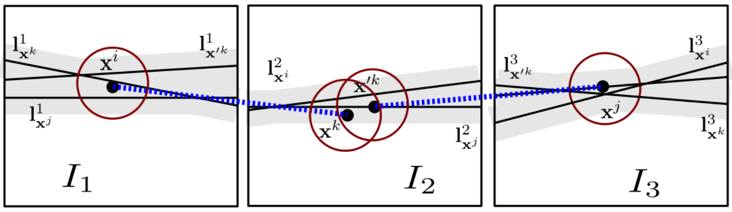

$\mathbf{l}^1_{\mathbf{x}^k}$
$\mathbf{l}^1_{\mathbf{x}'^k}$
$\mathbf{x}^i$
$\mathbf{l}^1_{\mathbf{x}^j}$
I_1
$\mathbf{l}^2_{\mathbf{x}^i}$
$\mathbf{x}'^k$
$\mathbf{x}^k$
$\mathbf{l}^2_{\mathbf{x}^j}$
I_2
$\mathbf{l}^3_{\mathbf{x}'^k}$
$\mathbf{l}^3_{\mathbf{x}^i}$
$\mathbf{x}^j$
$\mathbf{l}^3_{\mathbf{x}^k}$
I_3

Figure 8.11: The matched pairs $(\mathbf{x}^i, \mathbf{x}^k)$, $(\mathbf{x}'^k, \mathbf{x}^j)$ (blue dotted lines) are considered as inliers if the distance between $\mathbf{x}^k$, $\mathbf{x}'^k$ and the distance from a matched point to any epipolar line in the corresponding image are less than th (the circle radius)

Sec. 7.3.2), for an error threshold from 1 to 9 pixels. To merge the errors obtained by the point-to-epipolar-line distance in both the images, three different strategies were adopted. If $\varepsilon = [\varepsilon_1, \varepsilon_2]$ is the error vector for a match pair, the *symmetric error* is defined as $(\varepsilon_1 + \varepsilon_2)/2$, the *geometric error* as $\sqrt{\varepsilon_1^2 + \varepsilon_2^2}$ and the *max error* as $\max(\varepsilon_1, \varepsilon_2)$. A total of 4×4 different RANSAC methodologies were applied.

The maximum number of iterations was set to 5000. For RANSAC methodologies the nearest neighbour selection on a hard match set (see Sec. 5.1.1) was adopted as the input set $\mathcal{P}$, while the soft sparse matching takes as input all the possible matches. The HarrisZ detector was used to detect features, while the feature descriptor vectors were extracted by the sGLOH descriptor (see Sec. 6).

Since all RANSAC based algorithms do not output the same inlier set at each run, a total of five runs has been executed for each stereo pair obtaining $5^3 = 125$ different validation sets. The stereo pair which maximized the mean value of $\mathcal{G}$ (see eq. 8.16) among all different methods has been selected from the three possible choices.

The average value for each measure, as well as the maximum, the minimum and the standard deviation are reported, the same statistics about the running time are also shown. The average inlier percentage is used to rank the algorithms and, among various RANSACs for a fixed methodology and a fixed error measure, only the best result is reported from all the possible choices of the threshold error.

Evaluation results are reported in tables 8.1–8.11, according to the inlier percentage $\mathcal{I}(\%)$, the absolute number of inliers $\mathcal{I}(\#)$, the max convex hull area percentage $\mathcal{H}(\%)$ and the maximal feature area coverage percentage $\mathcal{V}(\%)$. The errors in the human estimation of the ground truth fundamental matrix is also reported as well as the running time of each method, implemented in Matlab.

As it can be seen from the tables, the difference in the average inlier percentage is quite similar for all the image sequences, however the soft sparse matching is ranked in the first positions in most cases. Moreover the absolute number of matches in $\mathcal{I}$, as well as the maximal convex hull area $\mathcal{H}$ and coverage area $\mathcal{V}$ for the soft sparse matching are the highest almost in all sequences. It should also be noted that the input set is larger for the soft sparse matching so that it must contains an higher percentage of outliers.

In figs. 8.12–8.22 the best results for each sequence are reported. In particular the upper rows show the best result obtained for the sparse soft matching, while the lower rows show the best ranked method according to the average inlier percentage reported in the corresponding table. For each method, in the first row the stereo pair with superimposed inlier feature ellipses and the intermediate image are shown (left to right). The max convex hull and the corresponding hull in the other image are also reported (blue boundaries). Moreover, the second row for each method shows the detected inlier matches (red) on the first image and the outliers on the second image (green), according to the evaluation criterion. The last image of each second row shows the images of the sequence superimposed with the correct chains, according to the threshold of 5 pixels.

By inspecting the standard deviation of the absolute number of inliers it can be seen that the proposed algorithm is quite stable. There is also no RANSAC method that is better among the others.

As a main drawback, the computational time is higher, however the use of faster model check (see Sec. 7.4.4) in the soft sparse method was not investigated. As strong point, the proposed method does not require any error threshold in contrast to the other RANSAC methods. If no information about the errors in the images is given, more than one RANSAC runs are required.

The soft sparse matching is strongly guided by the feature descriptor similarities and it can fail when the feature descriptors do not provided good clues on the matches, as shown in fig. 8.7 for the “plant” scene. In this case also RANSAC fails, while PROSAC has been reported to be successful [105], but on a different setup.

The evaluation framework is promising. However some matches which can look correct to human inspections are wrong in the sense of the pure geometrical test performed, which gives an accurate estimation of the inliers. Future works may include further information provided by the feature patch, such as the scale and the shape, to alleviate this issue.

8.3.3 Final remarks

The proposed soft sparse matching is very promising, in particular for the high number of correct matches and the high coverage of the image. Some aspects have to be further investigated, as the use of a fast model check to reduce the running time, and a stop criterion based on statistical hypothesis.

The proposed validation framework seems effective. New tests to validate not only other matching strategies but also feature detectors and descriptors can be performed. Further measures can be included to obtain more insights in the accuracy of the algorithms to test.

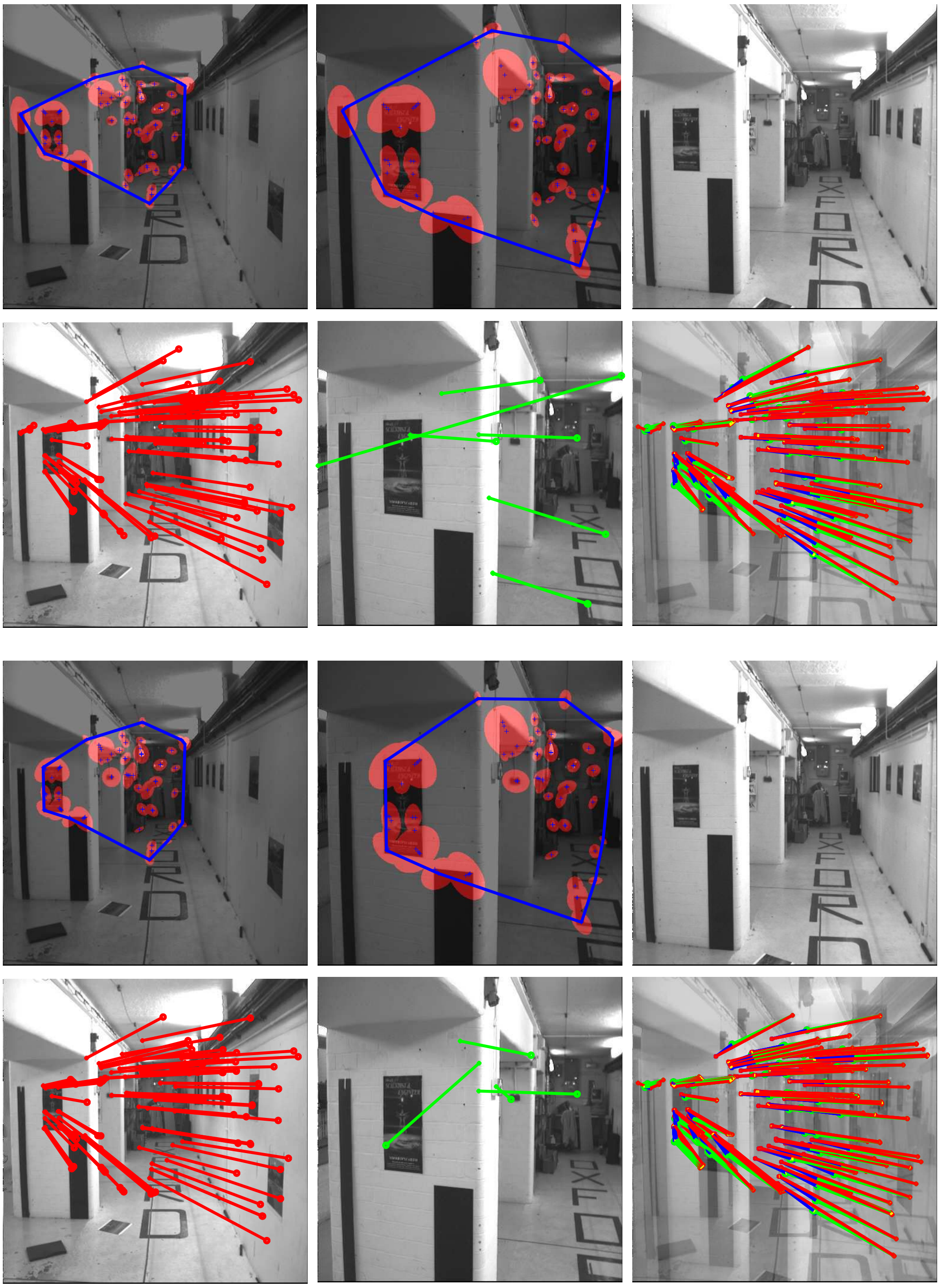


Figure 8.12: The corridor image sequence with the best results (see text for details)

Sequence: corridor
Best image pairs 0 2 with a 5 pixel chain error threshold

Groundtruth fund. matrix estimation pixel reprojection error

Image			1^{st} image		2^{nd} image			
pair	min	mean	max	std	min	mean	max	std
0–1	0.00	1.28	5.90	1.12	0.00	1.02	4.30	0.90
1–2	0.00	1.08	5.64	1.04	0.00	0.65	4.29	0.62
1–2	0.02	0.98	5.13	0.93	0.01	0.73	4.70	0.68

Algorithm statistics **Running time**

	$\mathcal{I}$(%)	$\mathcal{I}$(#)	$\mathcal{H}$(%)	$\mathcal{V}$(%)		$\mathcal{I}$(%)	$\mathcal{I}$(#)	$\mathcal{H}$(%)	$\mathcal{V}$(%)		time(s)		time(s)
SPARSE SOFT MATCHING													
mean	87.56	69	45.30	15.97	best	91.25	73	45.43	15.96	mean	196	max	334
std	2.85	1	1.25	0.64	worst	81.70	67	43.25	16.13	std	119	min	72
RANSAC WITH GEOMETRIC ERROR DISTANCE AND 4 PIXEL THRESHOLD													
mean	87.47	60	33.18	11.50	best	95.18	79	41.16	15.01	mean	28	max	37
std	6.04	11	5.30	2.59	worst	76.92	50	26.82	7.80	std	14	min	8
MSAC WITH SYMMETRIC ERROR DISTANCE AND 3 PIXEL THRESHOLD													
mean	87.13	62	31.96	10.92	best	93.93	62	34.79	11.95	mean	27	max	37
std	8.00	11	5.01	3.40	worst	70.49	43	22.09	5.13	std	13	min	8
MLESAC WITH SAMPSON ERROR DISTANCE AND 1 PIXEL THRESHOLD													
mean	86.16	38	22.33	7.58	best	92.85	39	32.26	5.07	mean	38	max	62
std	5.18	3	5.17	1.65	worst	71.73	33	20.46	7.04	std	27	min	12
MSAC WITH GEOMETRIC ERROR DISTANCE AND 4 PIXEL THRESHOLD													
mean	85.93	49	28.39	9.08	best	92.42	61	38.09	10.27	mean	27	max	37
std	4.26	6	5.25	1.29	worst	78.94	45	29.13	8.38	std	14	min	8
RANSAC WITH MAX ERROR DISTANCE AND 4 PIXEL THRESHOLD													
mean	85.26	63	33.22	11.49	best	91.30	84	44.74	14.79	mean	26	max	37
std	6.50	13	7.72	2.64	worst	74.24	49	26.46	9.22	std	11	min	10
MLESAC WITH MAX ERROR DISTANCE AND 2 PIXEL THRESHOLD													
mean	85.18	66	34.97	12.04	best	92.85	91	45.62	17.19	mean	28	max	39
std	6.18	16	9.80	3.79	worst	75.80	47	22.93	8.32	std	13	min	9
MSAC WITH MAX ERROR DISTANCE AND 4 PIXEL THRESHOLD													
mean	84.80	53	26.69	8.33	best	91.93	57	28.87	10.88	mean	25	max	38
std	5.64	3	2.80	1.30	worst	73.84	48	22.96	7.83	std	11	min	10
RANSAC WITH SYMMETRIC ERROR DISTANCE AND 4 PIXEL THRESHOLD													
mean	84.79	62	30.62	10.70	best	91.66	88	45.09	16.39	mean	26	max	37
std	4.22	12	7.33	2.97	worst	80.00	56	26.92	8.95	std	10	min	10
MSAC WITH SAMPSON ERROR DISTANCE AND 3 PIXEL THRESHOLD													
mean	84.15	71	33.99	12.68	best	92.04	81	38.65	14.80	mean	26	max	38
std	5.22	12	6.71	2.92	worst	78.08	57	32.33	11.38	std	10	min	10
RANSAC WITH SAMPSON ERROR DISTANCE AND 2 PIXEL THRESHOLD													
mean	81.88	55	29.77	11.35	best	94.44	68	39.55	12.99	mean	27	max	38
std	10.96	11	5.56	1.82	worst	57.62	34	25.67	7.22	std	12	min	9
MLESAC WITH GEOMETRIC ERROR DISTANCE AND 2 PIXEL THRESHOLD													
mean	80.36	43	23.41	7.39	best	84.90	45	19.72	7.65	mean	29	max	39
std	5.48	3	4.08	1.55	worst	69.81	37	22.56	5.44	std	14	min	9
MLESAC WITH SYMMETRIC ERROR DISTANCE AND 2 PIXEL THRESHOLD													
mean	75.71	52	28.55	8.56	best	83.82	57	31.47	10.45	mean	26	max	38
std	6.78	6	2.74	1.08	worst	67.79	40	30.34	7.39	std	11	min	10
LMEDS WITH SYMMETRIC ERROR DISTANCE AND 8 PIXEL THRESHOLD													
mean	68.75	46	23.35	8.19	best	74.24	49	22.96	6.93	mean	24	max	38
std	4.07	4	5.36	2.04	worst	63.29	50	31.38	10.64	std	6	min	12
LMEDS WITH MAX ERROR DISTANCE AND 6 PIXEL THRESHOLD													
mean	68.44	39	20.62	7.55	best	79.48	62	35.11	9.37	mean	29	max	51
std	5.70	11	11.35	2.10	worst	57.81	37	26.80	8.27	std	9	min	11
LMEDS WITH SAMPSON ERROR DISTANCE AND 7 PIXEL THRESHOLD													
mean	67.99	64	27.09	11.53	best	71.27	67	30.93	13.42	mean	23	max	38
std	2.27	7	3.36	2.62	worst	63.00	63	23.64	11.35	std	5	min	13
LMEDS WITH GEOMETRIC ERROR DISTANCE AND 9 PIXEL THRESHOLD													
mean	63.19	40	22.86	7.43	best	70.37	38	18.22	9.74	mean	26	max	38
std	3.70	4	6.86	2.04	worst	56.96	45	27.90	8.61	std	7	min	10

Table 8.1: Evaluation results on the corridor sequence (see the text for details)

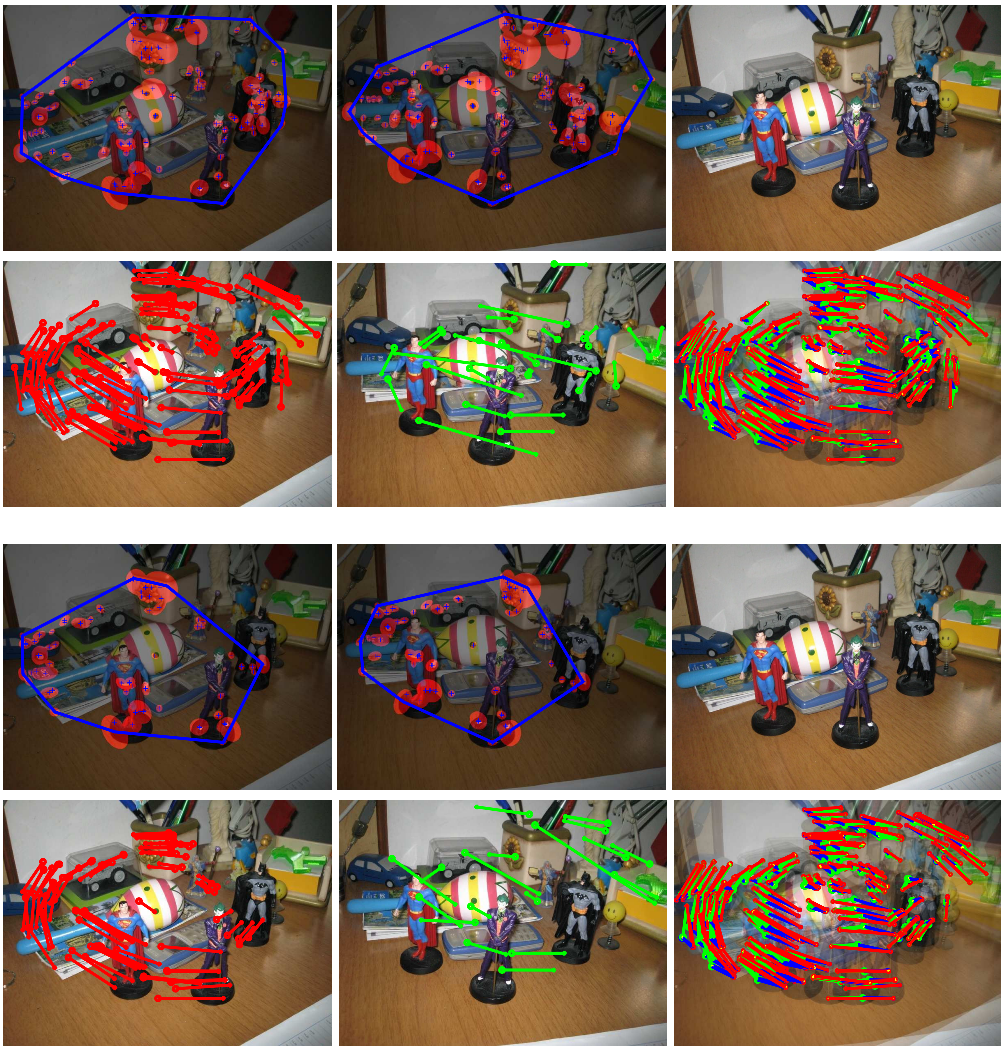


Figure 8.13: The DC image sequence with the best results (see text for details)

Sequence: DC
Best image pairs 0 2 with a 5 pixel chain error threshold

Groundtruth fund. matrix estimation pixel reprojection error

Image			1^{st} image		2^{nd} image			
pair	min	mean	max	std	min	mean	max	std
0–1	0.00	0.83	4.97	0.81	0.00	0.80	4.36	0.78
1–2	0.00	0.94	5.86	0.85	0.00	0.92	5.89	0.82
1–2	0.00	0.75	3.10	0.66	0.00	0.76	4.17	0.66

Algorithm statistics													**Running time**
	$\mathcal{I}$(%)	$\mathcal{I}$(#)	$\mathcal{H}$(%)	$\mathcal{V}$(%)		$\mathcal{I}$(%)	$\mathcal{I}$(#)	$\mathcal{H}$(%)	$\mathcal{V}$(%)		time(s)		time(s)
SPARSE SOFT MATCHING													
mean	79.41	128	45.81	11.05	best	83.43	141	47.33	12.46	mean	812	max	1041
std	1.89	6	1.18	0.90	worst	76.04	127	46.21	11.18	std	648	min	120
MSAC WITH SAMPSON ERROR DISTANCE AND 2 PIXEL THRESHOLD													
mean	76.92	90	36.12	7.30	best	83.18	94	33.00	7.49	mean	74	max	150
std	4.23	8	4.54	1.24	worst	71.69	76	29.99	5.48	std	16	min	51
MSAC WITH MAX ERROR DISTANCE AND 4 PIXEL THRESHOLD													
mean	76.63	94	36.80	7.52	best	80.70	92	40.61	7.51	mean	73	max	158
std	3.71	13	3.58	0.98	worst	68.64	81	29.44	6.56	std	10	min	58
MLESAC WITH MAX ERROR DISTANCE AND 2 PIXEL THRESHOLD													
mean	76.11	107	39.74	8.19	best	83.21	119	40.53	8.77	mean	76	max	157
std	5.06	18	3.13	1.14	worst	68.91	102	44.75	7.39	std	12	min	58
RANSAC WITH MAX ERROR DISTANCE AND 3 PIXEL THRESHOLD													
mean	75.53	90	37.78	6.85	best	81.32	135	44.67	10.26	mean	74	max	155
std	3.77	22	5.76	1.98	worst	65.45	72	33.16	5.00	std	17	min	50
MSAC WITH SYMMETRIC ERROR DISTANCE AND 4 PIXEL THRESHOLD													
mean	74.64	115	37.10	8.93	best	82.82	135	30.38	9.09	mean	73	max	151
std	5.77	22	6.42	1.61	worst	67.88	93	32.10	6.64	std	10	min	58
MSAC WITH GEOMETRIC ERROR DISTANCE AND 5 PIXEL THRESHOLD													
mean	74.51	95	32.88	7.05	best	77.34	99	34.77	8.16	mean	71	max	159
std	1.76	8	4.73	0.75	worst	71.54	88	26.82	7.74	std	16	min	53
RANSAC WITH GEOMETRIC ERROR DISTANCE AND 6 PIXEL THRESHOLD													
mean	74.40	109	39.19	8.51	best	80.39	164	44.70	12.20	mean	72	max	154
std	4.04	28	5.68	2.20	worst	67.47	83	38.73	5.43	std	9	min	59
MLESAC WITH GEOMETRIC ERROR DISTANCE AND 2 PIXEL THRESHOLD													
mean	73.53	85	37.68	7.07	best	79.83	95	40.89	7.99	mean	73	max	155
std	3.36	9	2.15	0.98	worst	69.09	76	38.61	5.78	std	16	min	54
RANSAC WITH SAMPSON ERROR DISTANCE AND 4 PIXEL THRESHOLD													
mean	73.28	139	42.94	10.52	best	79.90	167	46.73	12.07	mean	64	max	158
std	4.01	18	5.15	1.17	worst	68.62	140	46.70	10.66	std	6	min	50
RANSAC WITH SYMMETRIC ERROR DISTANCE AND 3 PIXEL THRESHOLD													
mean	72.68	88	33.13	6.70	best	77.88	81	31.71	5.20	mean	76	max	152
std	3.59	18	4.15	1.45	worst	65.55	59	29.42	5.36	std	14	min	55
MLESAC WITH SYMMETRIC ERROR DISTANCE AND 2 PIXEL THRESHOLD													
mean	71.33	106	36.75	7.87	best	79.33	119	37.60	9.32	mean	73	max	156
std	8.43	21	4.00	1.87	worst	56.15	73	31.97	4.72	std	13	min	56
MLESAC WITH SAMPSON ERROR DISTANCE AND 2 PIXEL THRESHOLD													
mean	70.51	146	42.05	11.04	best	73.36	135	35.23	8.99	mean	60	max	152
std	2.27	11	4.25	1.09	worst	67.13	143	47.02	11.15	std	6	min	45
LMEDS WITH SAMPSON ERROR DISTANCE AND 2 PIXEL THRESHOLD													
mean	68.23	57	33.64	4.44	best	76.13	67	35.56	4.82	mean	77	max	157
std	5.00	6	3.03	0.55	worst	57.47	50	30.91	4.56	std	16	min	51
LMEDS WITH GEOMETRIC ERROR DISTANCE AND 6 PIXEL THRESHOLD													
mean	65.47	82	30.48	5.72	best	69.49	82	29.93	6.39	mean	69	max	162
std	2.96	16	5.63	1.48	worst	58.62	51	22.65	2.92	std	9	min	57
LMEDS WITH SYMMETRIC ERROR DISTANCE AND 7 PIXEL THRESHOLD													
mean	64.98	108	34.62	8.32	best	70.06	103	38.21	7.91	mean	62	max	152
std	2.82	17	4.27	1.46	worst	60.47	101	28.64	7.88	std	6	min	54
LMEDS WITH MAX ERROR DISTANCE AND 4 PIXEL THRESHOLD													
mean	61.51	65	27.41	5.81	best	76.28	119	40.33	9.85	mean	76	max	159
std	7.33	29	10.16	2.50	worst	53.84	42	25.81	3.59	std	11	min	58

Table 8.2: Evaluation results on the DC sequence (see the text for details)

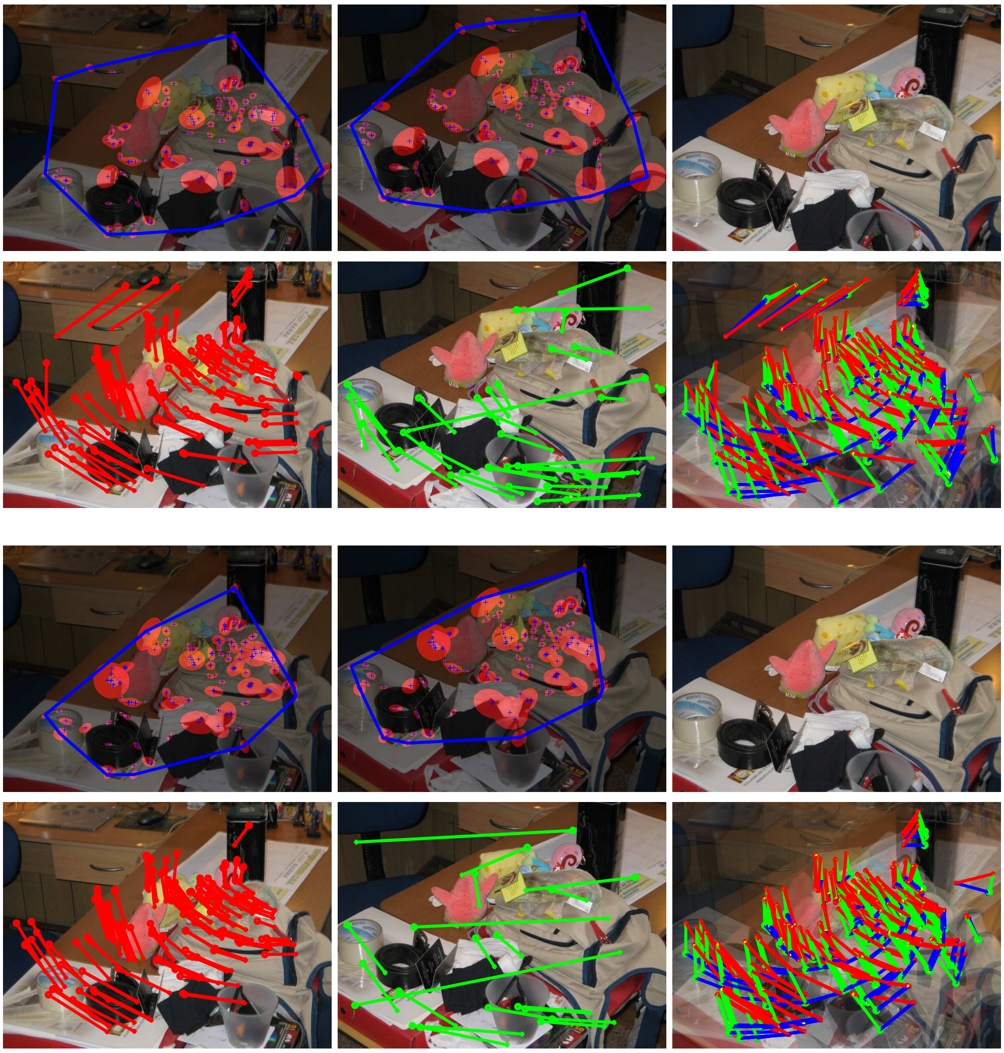


Figure 8.14: The desk image sequence with the best results (see text for details)

Sequence: desk

Best image pairs 0 2 with a 5 pixel chain error threshold

Groundtruth fund. matrix estimation pixel reprojection error

Image			1^{st} image		2^{nd} image			
pair	min	mean	max	std	min	mean	max	std
0–1	0.00	1.29	16.80	1.67	0.00	1.17	15.83	1.54
1–2	0.01	1.12	6.32	1.23	0.01	1.09	6.23	1.19
1–2	0.00	1.21	9.84	1.59	0.00	1.28	15.44	1.68

Algorithm statistics **Running time**

	$\mathcal{I}$(%)	$\mathcal{I}$(#)	$\mathcal{H}$(%)	$\mathcal{V}$(%)		$\mathcal{I}$(%)	$\mathcal{I}$(#)	$\mathcal{H}$(%)	$\mathcal{V}$(%)		time(s)		time(s)
msac with symmetric error distance and 4 pixel threshold													
mean	84.80	130	31.73	9.75	best	89.20	157	36.10	11.62	mean	54	max	79
std	2.27	19	3.79	1.49	worst	82.22	111	26.15	8.15	std	32	min	16
ransac with geometric error distance and 7 pixel threshold													
mean	84.65	151	37.35	12.40	best	86.41	140	36.12	12.39	mean	51	max	79
std	1.24	12	3.50	0.62	worst	82.95	146	42.62	11.82	std	20	min	20
msac with geometric error distance and 5 pixel threshold													
mean	84.41	98	24.69	8.20	best	89.65	130	35.53	10.99	mean	60	max	80
std	3.43	17	7.63	1.67	worst	78.99	94	16.48	6.18	std	45	min	11
mlesac with sampson error distance and 2 pixel threshold													
mean	84.07	149	38.03	12.04	best	87.62	170	39.93	12.89	mean	50	max	81
std	2.66	18	1.29	0.56	worst	80.66	121	35.97	11.33	std	25	min	17
msac with max error distance and 4 pixel threshold													
mean	84.01	108	29.01	8.82	best	87.00	154	31.38	12.71	mean	60	max	80
std	2.01	23	5.11	2.04	worst	79.82	91	27.08	6.58	std	40	min	13
ransac with sampson error distance and 2 pixel threshold													
mean	83.93	104	27.16	8.57	best	87.05	121	28.21	10.31	mean	68	max	119
std	2.08	23	8.85	2.09	worst	79.43	85	16.99	6.14	std	46	min	19
msac with sampson error distance and 2 pixel threshold													
mean	83.89	108	27.78	9.14	best	86.29	107	28.76	8.48	mean	69	max	130
std	1.55	6	3.51	1.28	worst	80.83	97	21.64	7.72	std	45	min	23
ransac with max error distance and 3 pixel threshold													
mean	83.65	92	23.37	7.10	best	85.85	85	27.16	7.98	mean	64	max	102
std	1.97	14	5.44	1.41	worst	78.33	94	22.54	7.81	std	45	min	17
ransac with symmetric error distance and 3 pixel threshold													
mean	83.05	104	27.41	8.80	best	89.16	107	23.08	7.91	mean	73	max	149
std	2.97	9	4.45	1.61	worst	79.19	118	33.29	10.74	std	42	min	26
mlesac with geometric error distance and 2 pixel threshold													
mean	82.30	95	27.72	7.45	best	86.58	142	33.76	10.18	mean	72	max	141
std	4.12	22	4.92	1.56	worst	73.58	78	33.61	6.38	std	46	min	27
mlesac with symmetric error distance and 2 pixel threshold													
mean	80.59	109	28.06	8.41	best	84.26	150	29.64	11.00	mean	56	max	83
std	3.34	21	3.04	1.63	worst	74.56	85	29.04	5.88	std	30	min	16
mlesac with max error distance and 2 pixel threshold													
mean	80.31	104	32.36	8.66	best	88.11	126	38.10	9.35	mean	57	max	84
std	4.75	17	3.21	1.08	worst	74.00	74	34.53	7.48	std	35	min	15
sparse soft matching													
mean	74.90	117	48.61	10.52	best	77.70	122	51.83	11.85	mean	682	max	850
std	1.32	5	3.16	1.27	worst	71.89	110	49.63	8.00	std	519	min	108
lmeds with max error distance and 7 pixel threshold													
mean	73.34	70	22.12	6.04	best	82.88	92	21.26	7.44	mean	50	max	82
std	8.93	22	5.95	1.27	worst	58.18	32	17.89	4.37	std	19	min	22
lmeds with symmetric error distance and 4 pixel threshold													
mean	72.54	51	26.48	4.34	best	81.81	63	12.78	3.91	mean	63	max	98
std	8.26	11	9.52	0.74	worst	56.06	37	27.16	3.90	std	30	min	17
lmeds with geometric error distance and 6 pixel threshold													
mean	70.91	56	19.95	4.56	best	79.26	65	13.76	4.18	mean	55	max	82
std	6.25	12	6.13	0.72	worst	60.63	57	20.34	4.88	std	31	min	16
lmeds with sampson error distance and 3 pixel threshold													
mean	63.79	55	23.29	4.70	best	75.00	105	29.15	9.60	mean	57	max	79
std	11.01	29	6.24	2.77	worst	43.75	14	15.44	0.81	std	27	min	12

Table 8.3: Evaluation results on the desk sequence (see the text for details)

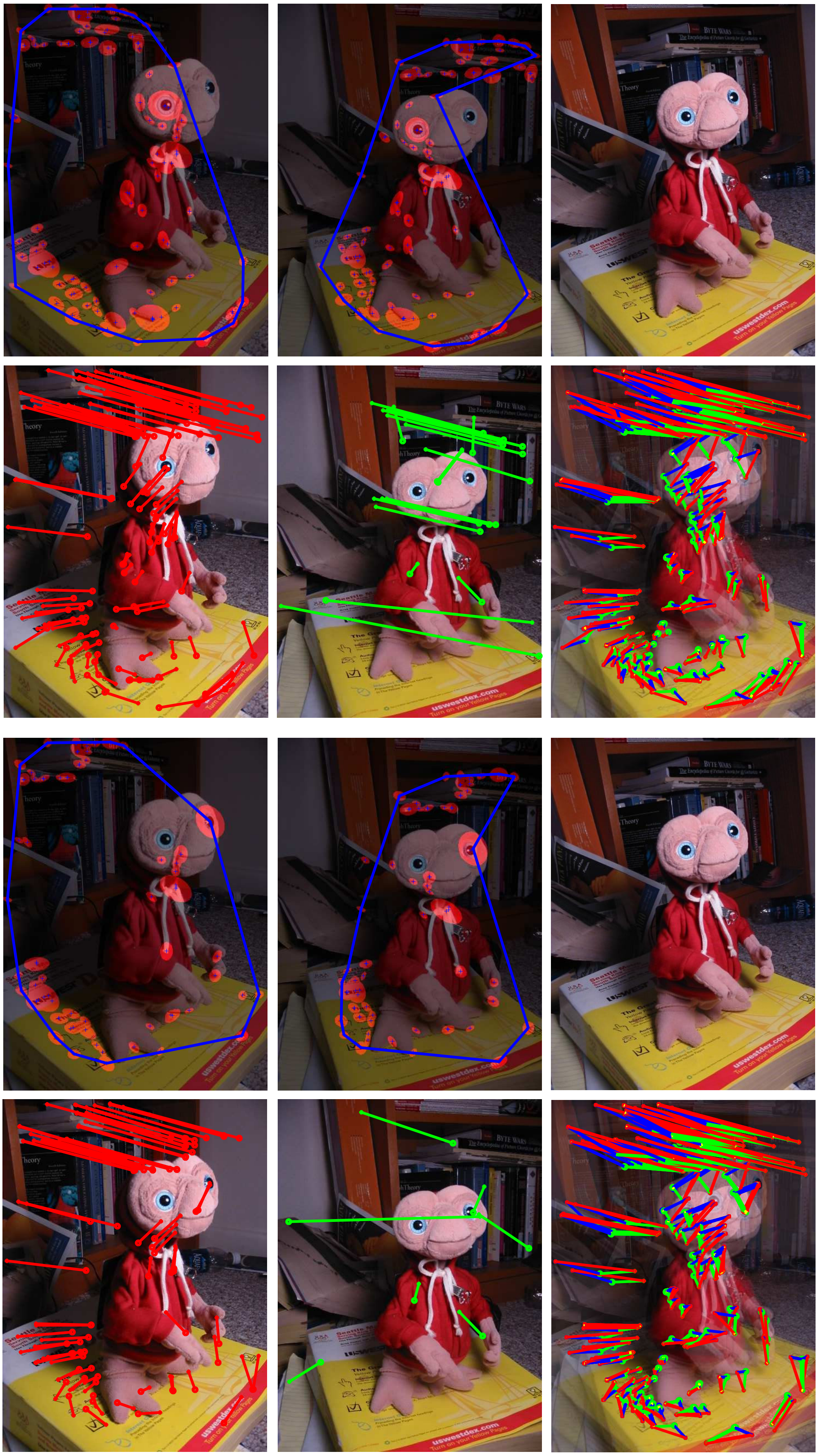


Figure 8.15: The ET image sequence with the best results (see text for details)

Sequence: ET
Best image pairs 0 2 with a 5 pixel chain error threshold

Groundtruth fund. matrix estimation pixel reprojection error

Image			1^{st} image		2^{nd} image			
pair	min	mean	max	std	min	mean	max	std
0–1	0.00	0.63	4.59	0.58	0.00	0.69	4.99	0.63
1–2	0.00	0.82	4.40	0.74	0.00	0.91	5.10	0.84
1–2	0.00	0.87	8.32	1.09	0.01	0.89	4.28	1.11

Algorithm statistics													**Running time**
	$\mathcal{I}(\%)$	$\mathcal{I}(\#)$	$\mathcal{H}(\%)$	$\mathcal{V}(\%)$		$\mathcal{I}(\%)$	$\mathcal{I}(\#)$	$\mathcal{H}(\%)$	$\mathcal{V}(\%)$		time(s)		time(s)
MSAC WITH SYMMETRIC ERROR DISTANCE AND 3 PIXEL THRESHOLD													
mean	84.19	83	57.42	8.02	best	91.56	76	65.70	7.08	mean	36	max	66
std	4.49	10	6.82	1.43	worst	77.65	73	49.87	6.63	std	8	min	20
SPARSE SOFT MATCHING													
mean	82.83	100	65.66	9.49	best	84.42	103	64.92	10.59	mean	294	max	382
std	1.15	1	1.58	0.62	worst	80.16	97	64.92	8.81	std	249	min	38
RANSAC WITH MAX ERROR DISTANCE AND 5 PIXEL THRESHOLD													
mean	82.77	88	56.77	8.10	best	88.67	94	53.89	9.13	mean	28	max	53
std	5.48	19	7.70	1.81	worst	73.75	59	43.51	5.28	std	6	min	15
MSAC WITH GEOMETRIC ERROR DISTANCE AND 9 PIXEL THRESHOLD													
mean	82.25	139	68.25	12.75	best	84.30	145	69.18	13.10	mean	22	max	41
std	1.07	3	1.88	0.53	worst	80.12	133	69.54	12.78	std	5	min	14
RANSAC WITH GEOMETRIC ERROR DISTANCE AND 7 PIXEL THRESHOLD													
mean	81.76	102	59.88	9.12	best	88.51	131	65.90	10.58	mean	24	max	41
std	4.49	23	11.68	2.11	worst	77.06	84	37.06	7.24	std	5	min	14
MSAC WITH MAX ERROR DISTANCE AND 7 PIXEL THRESHOLD													
mean	81.53	127	66.02	11.04	best	83.95	136	68.56	13.18	mean	20	max	42
std	2.20	19	2.73	1.99	worst	77.58	90	61.65	7.42	std	4	min	15
MSAC WITH SAMPSON ERROR DISTANCE AND 1 PIXEL THRESHOLD													
mean	80.91	48	36.13	3.97	best	88.57	62	51.87	6.13	mean	58	max	81
std	6.64	9	13.96	1.15	worst	67.27	37	26.84	2.72	std	40	min	14
MLESAC WITH SYMMETRIC ERROR DISTANCE AND 2 PIXEL THRESHOLD													
mean	80.56	84	59.27	6.56	best	89.65	130	63.73	10.44	mean	30	max	54
std	4.83	23	3.42	1.97	worst	75.24	76	54.82	6.05	std	5	min	15
RANSAC WITH SYMMETRIC ERROR DISTANCE AND 3 PIXEL THRESHOLD													
mean	80.31	76	46.79	6.52	best	83.94	115	65.70	10.33	mean	37	max	105
std	3.27	21	11.16	2.21	worst	74.24	49	32.00	3.86	std	9	min	24
MLESAC WITH GEOMETRIC ERROR DISTANCE AND 2 PIXEL THRESHOLD													
mean	79.45	65	48.54	6.26	best	89.70	61	52.33	6.07	mean	41	max	68
std	7.62	18	9.77	1.60	worst	64.38	47	32.59	5.47	std	9	min	19
MLESAC WITH MAX ERROR DISTANCE AND 2 PIXEL THRESHOLD													
mean	78.52	83	56.68	7.09	best	80.00	104	63.61	8.27	mean	32	max	68
std	0.78	15	6.59	1.55	worst	76.62	59	55.86	4.56	std	7	min	19
MLESAC WITH SAMPSON ERROR DISTANCE AND 2 PIXEL THRESHOLD													
mean	78.36	91	51.55	8.18	best	86.40	89	47.50	7.32	mean	22	max	42
std	6.28	15	7.70	1.60	worst	67.79	80	40.16	7.21	std	5	min	14
RANSAC WITH SAMPSON ERROR DISTANCE AND 4 PIXEL THRESHOLD													
mean	76.96	99	53.00	8.51	best	84.51	131	64.78	10.85	mean	22	max	41
std	5.92	21	15.76	2.46	worst	66.36	73	28.72	5.11	std	4	min	15
LMEDS WITH SYMMETRIC ERROR DISTANCE AND 3 PIXEL THRESHOLD													
mean	74.14	41	37.40	3.49	best	88.57	62	40.25	4.32	mean	41	max	92
std	10.43	14	8.92	0.83	worst	61.36	27	20.84	2.55	std	10	min	24
LMEDS WITH GEOMETRIC ERROR DISTANCE AND 6 PIXEL THRESHOLD													
mean	71.81	47	39.42	4.65	best	81.39	70	53.24	6.96	mean	33	max	66
std	7.44	17	19.34	1.88	worst	55.00	33	21.10	3.32	std	6	min	19
LMEDS WITH SAMPSON ERROR DISTANCE AND 4 PIXEL THRESHOLD													
mean	71.56	59	48.19	5.32	best	79.80	83	57.67	7.19	mean	22	max	41
std	8.57	16	11.76	1.39	worst	55.35	31	26.14	2.93	std	4	min	15
LMEDS WITH MAX ERROR DISTANCE AND 6 PIXEL THRESHOLD													
mean	69.54	61	44.38	6.14	best	83.83	140	69.12	13.44	mean	21	max	41
std	14.20	42	15.95	3.87	worst	42.22	19	29.89	3.01	std	5	min	14

Table 8.4: Evaluation results on the ET sequence (see the text for details)

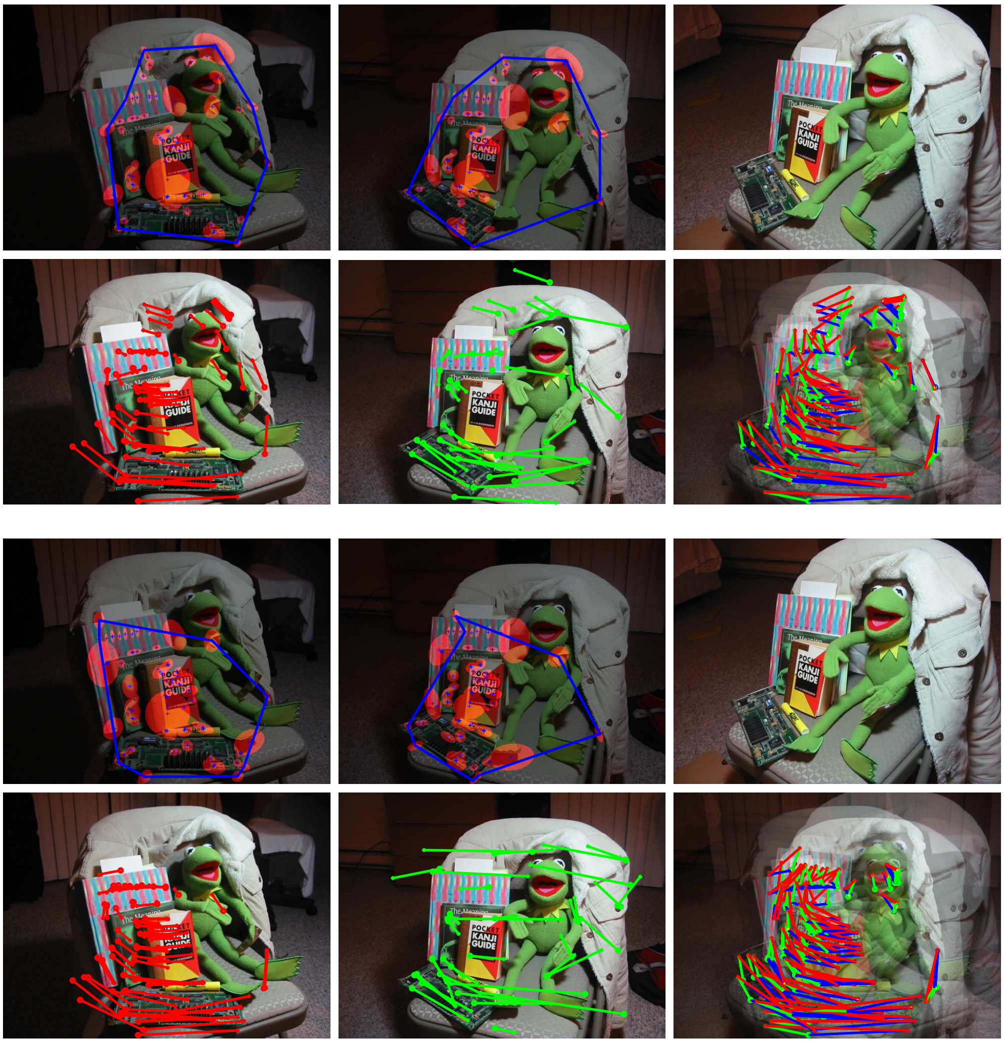


Figure 8.16: The Kermit image sequence with the best results (see text for details)

Sequence: Kermit
Best image pairs 0 1 with a 5 pixel chain error threshold

Groundtruth fund. matrix estimation pixel reprojection error

Image	1^{st} image				2^{nd} image			
pair	min	mean	max	std	min	mean	max	std
0–1	0.00	0.66	4.55	0.68	0.01	0.69	4.02	0.69
1–2	0.00	0.74	4.99	0.81	0.00	0.70	4.52	0.75
1–2	0.00	0.69	3.61	0.70	0.00	0.61	4.15	0.60

Algorithm statistics **Running time**

	$\mathcal{I}(\%)$	$\mathcal{I}(\#)$	$\mathcal{H}(\%)$	$\mathcal{V}(\%)$		$\mathcal{I}(\%)$	$\mathcal{I}(\#)$	$\mathcal{H}(\%)$	$\mathcal{V}(\%)$		time(s)		time(s)
ransac with max error distance and 7 pixel threshold													
mean	61.86	71	22.84	8.26	best	68.96	80	24.64	9.86	mean	41	max	55
std	3.47	6	3.72	0.98	worst	55.81	72	20.99	8.56	std	30	min	10
ransac with geometric error distance and 9 pixel threshold													
mean	61.79	70	24.86	8.74	best	67.56	75	24.68	9.06	mean	41	max	55
std	3.98	10	4.66	1.04	worst	53.33	56	14.95	7.26	std	30	min	9
msac with geometric error distance and 7 pixel threshold													
mean	61.53	58	17.82	6.95	best	67.59	73	22.27	8.63	mean	40	max	55
std	3.91	10	3.35	1.45	worst	55.20	53	19.46	6.00	std	30	min	9
msac with max error distance and 5 pixel threshold													
mean	61.32	64	21.87	7.71	best	67.24	78	25.84	10.01	mean	41	max	57
std	4.60	9	5.18	1.65	worst	53.57	45	13.46	5.21	std	30	min	10
ransac with symmetric error distance and 6 pixel threshold													
mean	61.09	61	20.69	7.46	best	66.66	76	27.13	8.84	mean	40	max	55
std	4.31	7	3.34	1.30	worst	50.52	48	17.38	5.11	std	30	min	10
msac with symmetric error distance and 7 pixel threshold													
mean	60.96	70	25.53	8.51	best	63.38	90	32.96	11.02	mean	40	max	55
std	1.68	11	5.04	1.79	worst	55.14	59	17.43	7.15	std	29	min	10
mlesac with geometric error distance and 3 pixel threshold													
mean	60.42	64	19.52	7.33	best	64.60	73	21.80	7.77	mean	41	max	55
std	3.23	7	4.65	0.65	worst	49.50	50	13.01	5.40	std	29	min	10
msac with sampson error distance and 5 pixel threshold													
mean	60.00	80	27.63	9.88	best	63.84	83	27.16	8.68	mean	40	max	55
std	1.89	6	3.89	1.15	worst	56.19	68	27.20	8.79	std	29	min	10
sparse soft matching													
mean	59.80	54	30.49	7.52	best	65.65	65	31.55	8.27	mean	184	max	278
std	2.81	4	2.25	0.90	worst	52.87	46	29.38	5.93	std	64	min	75
mlesac with max error distance and 2 pixel threshold													
mean	58.58	47	17.93	5.57	best	65.78	50	19.78	5.34	mean	41	max	56
std	4.40	9	5.45	1.20	worst	47.61	30	4.78	3.38	std	27	min	10
ransac with sampson error distance and 3 pixel threshold													
mean	58.27	53	19.86	6.35	best	67.50	54	20.62	6.09	mean	41	max	55
std	4.50	7	4.42	1.40	worst	50.61	41	15.18	4.46	std	30	min	10
mlesac with sampson error distance and 2 pixel threshold													
mean	57.73	64	26.44	8.77	best	64.48	69	29.21	9.23	mean	41	max	57
std	3.74	11	4.92	1.22	worst	51.85	56	15.20	6.83	std	30	min	10
mlesac with symmetric error distance and 3 pixel threshold													
mean	55.50	74	26.59	9.44	best	61.86	73	29.57	9.59	mean	39	max	55
std	3.73	5	3.13	0.69	worst	49.25	66	23.94	8.69	std	30	min	8
lmeds with max error distance and 9 pixel threshold													
mean	46.16	47	13.80	5.95	best	56.60	60	12.41	7.38	mean	40	max	56
std	7.35	11	7.01	1.40	worst	27.35	29	10.63	2.49	std	29	min	10
lmeds with symmetric error distance and 9 pixel threshold													
mean	43.44	41	19.13	5.41	best	59.25	64	18.67	8.45	mean	42	max	57
std	8.24	14	5.01	1.75	worst	21.91	16	15.07	1.54	std	30	min	10
lmeds with geometric error distance and 9 pixel threshold													
mean	40.29	29	9.78	3.18	best	56.86	29	5.94	1.71	mean	40	max	55
std	16.93	17	8.09	2.15	worst	4.16	1	0.00	0.05	std	29	min	10
lmeds with sampson error distance and 8 pixel threshold													
mean	38.09	38	14.82	4.63	best	45.61	52	20.33	6.73	mean	40	max	55
std	6.63	12	3.86	1.50	worst	24.13	21	13.54	3.50	std	29	min	9

Table 8.5: Evaluation results on the Kermit sequence (see the text for details)

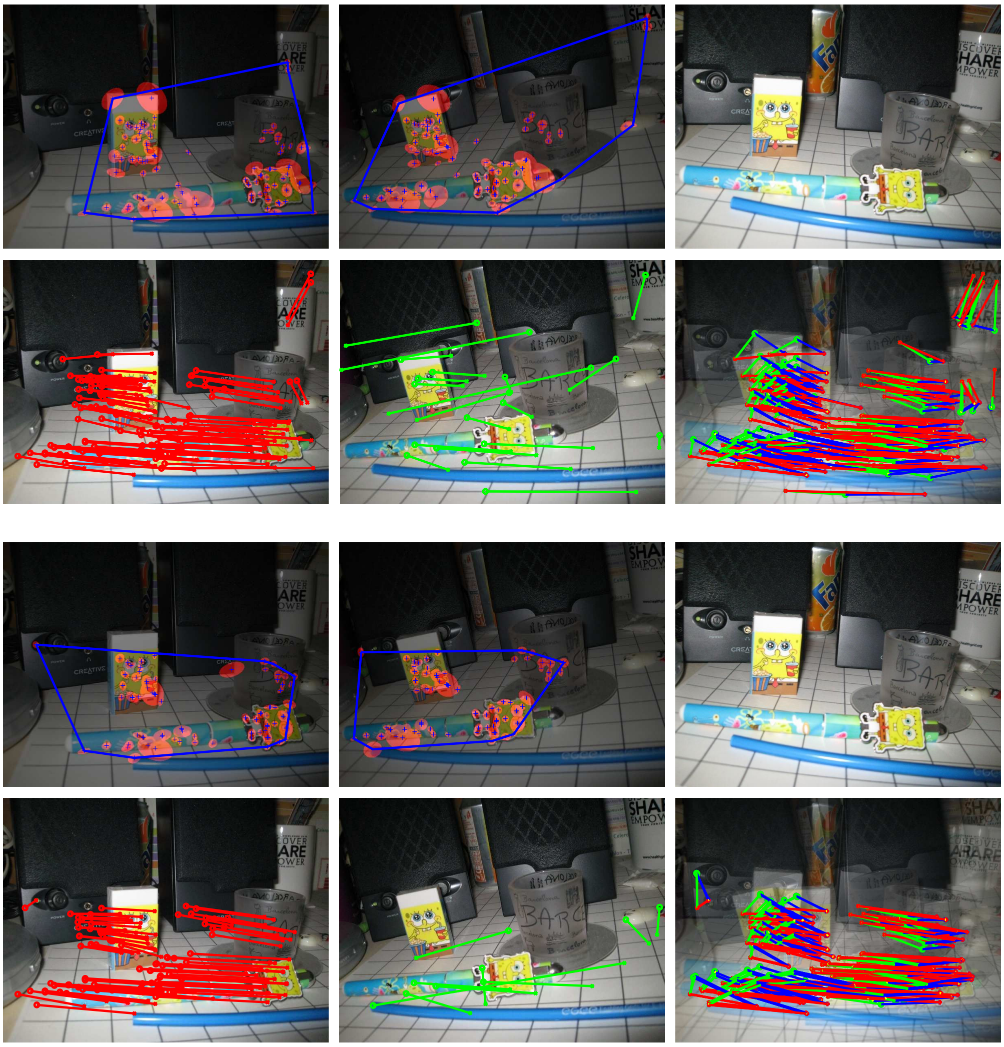


Figure 8.17: The pen image sequence with the best results (see text for details)

Sequence: pen
Best image pairs 0 2 with a 5 pixel chain error threshold

Groundtruth fund. matrix estimation pixel reprojection error

Image			1^{st} image		2^{nd} image			
pair	min	mean	max	std	min	mean	max	std
0–1	0.00	1.73	22.31	2.36	0.00	1.69	21.90	2.33
1–2	0.00	0.96	6.51	1.00	0.00	1.00	6.54	1.04
1–2	0.00	0.90	5.46	0.95	0.00	0.96	21.62	1.02

Algorithm statistics **Running time**

	$\mathcal{I}$(%)	$\mathcal{I}$(#)	$\mathcal{H}$(%)	$\mathcal{V}$(%)		$\mathcal{I}$(%)	$\mathcal{I}$(#)	$\mathcal{H}$(%)	$\mathcal{V}$(%)		time(s)		time(s)
MSAC WITH MAX ERROR DISTANCE AND 3 PIXEL THRESHOLD													
mean	82.73	84	24.33	5.14	best	91.42	96	27.91	5.44	mean	59	max	85
std	5.03	9	5.70	0.73	worst	68.69	79	20.67	5.64	std	43	min	14
MSAC WITH GEOMETRIC ERROR DISTANCE AND 2 PIXEL THRESHOLD													
mean	80.55	50	11.90	2.65	best	87.50	77	21.55	4.05	mean	62	max	86
std	6.70	15	4.33	0.77	worst	65.21	30	8.19	1.55	std	44	min	16
RANSAC WITH MAX ERROR DISTANCE AND 3 PIXEL THRESHOLD													
mean	79.64	71	18.67	4.14	best	86.17	81	25.76	4.74	mean	59	max	82
std	6.05	11	6.54	0.83	worst	68.62	70	19.33	4.12	std	42	min	14
RANSAC WITH GEOMETRIC ERROR DISTANCE AND 6 PIXEL THRESHOLD													
mean	79.31	116	26.19	7.04	best	83.12	133	25.25	7.81	mean	51	max	91
std	3.23	16	6.45	1.22	worst	73.43	94	21.36	5.25	std	35	min	13
MSAC WITH SAMPSON ERROR DISTANCE AND 2 PIXEL THRESHOLD													
mean	78.96	87	20.25	4.87	best	87.21	116	23.32	6.67	mean	60	max	83
std	7.79	19	3.78	1.00	worst	64.36	56	13.87	3.29	std	43	min	14
RANSAC WITH SYMMETRIC ERROR DISTANCE AND 4 PIXEL THRESHOLD													
mean	78.67	97	20.05	5.77	best	87.23	123	28.91	7.37	mean	48	max	59
std	4.55	17	5.62	1.05	worst	74.25	75	11.61	4.40	std	38	min	6
MSAC WITH SYMMETRIC ERROR DISTANCE AND 2 PIXEL THRESHOLD													
mean	78.12	62	15.40	3.89	best	82.47	80	22.51	4.30	mean	66	max	134
std	3.24	12	5.38	0.53	worst	71.66	43	7.84	3.29	std	43	min	24
MLESAC WITH SYMMETRIC ERROR DISTANCE AND 2 PIXEL THRESHOLD													
mean	77.57	93	21.42	5.35	best	81.74	103	23.21	6.52	mean	52	max	60
std	2.00	5	3.05	0.53	worst	74.33	84	21.47	4.76	std	43	min	6
MLESAC WITH GEOMETRIC ERROR DISTANCE AND 2 PIXEL THRESHOLD													
mean	75.95	81	17.93	4.93	best	85.41	123	25.23	7.62	mean	61	max	85
std	8.74	27	4.64	1.48	worst	62.65	52	16.07	3.81	std	44	min	15
MLESAC WITH MAX ERROR DISTANCE AND 2 PIXEL THRESHOLD													
mean	75.79	82	18.54	4.79	best	81.08	90	14.17	4.77	mean	53	max	83
std	2.92	4	3.94	0.22	worst	71.81	79	23.77	4.77	std	38	min	11
MLESAC WITH SAMPSON ERROR DISTANCE AND 2 PIXEL THRESHOLD													
mean	74.99	117	34.32	7.06	best	79.11	125	42.61	8.26	mean	45	max	61
std	2.51	7	10.00	0.61	worst	72.22	104	19.20	6.65	std	17	min	13
RANSAC WITH SAMPSON ERROR DISTANCE AND 3 PIXEL THRESHOLD													
mean	74.58	95	22.07	5.65	best	78.46	102	25.07	6.05	mean	48	max	60
std	2.77	7	3.51	0.59	worst	70.39	88	19.70	4.92	std	33	min	8
SPARSE SOFT MATCHING													
mean	72.82	85	36.36	6.76	best	83.47	101	41.27	8.21	mean	549	max	755
std	5.38	8	6.68	0.67	worst	65.48	74	39.80	5.93	std	345	min	134
LMEDS WITH SYMMETRIC ERROR DISTANCE AND 5 PIXEL THRESHOLD													
mean	71.36	83	17.58	5.01	best	78.23	115	20.08	6.35	mean	47	max	59
std	5.30	21	3.65	0.92	worst	64.55	51	12.10	3.95	std	25	min	9
LMEDS WITH GEOMETRIC ERROR DISTANCE AND 9 PIXEL THRESHOLD													
mean	68.59	95	26.90	6.26	best	74.73	139	45.63	8.33	mean	43	max	59
std	5.11	23	10.99	1.38	worst	56.92	74	14.35	4.87	std	19	min	12
LMEDS WITH SAMPSON ERROR DISTANCE AND 5 PIXEL THRESHOLD													
mean	67.98	106	32.21	6.59	best	71.95	118	41.29	7.10	mean	42	max	61
std	3.35	18	8.80	1.18	worst	62.34	101	30.45	6.03	std	20	min	15
LMEDS WITH MAX ERROR DISTANCE AND 9 PIXEL THRESHOLD													
mean	61.76	88	23.23	5.31	best	69.51	130	32.50	7.61	mean	40	max	60
std	5.96	28	6.98	1.78	worst	54.23	64	12.75	3.96	std	15	min	17

Table 8.6: Evaluation results on the pen sequence (see the text for details)

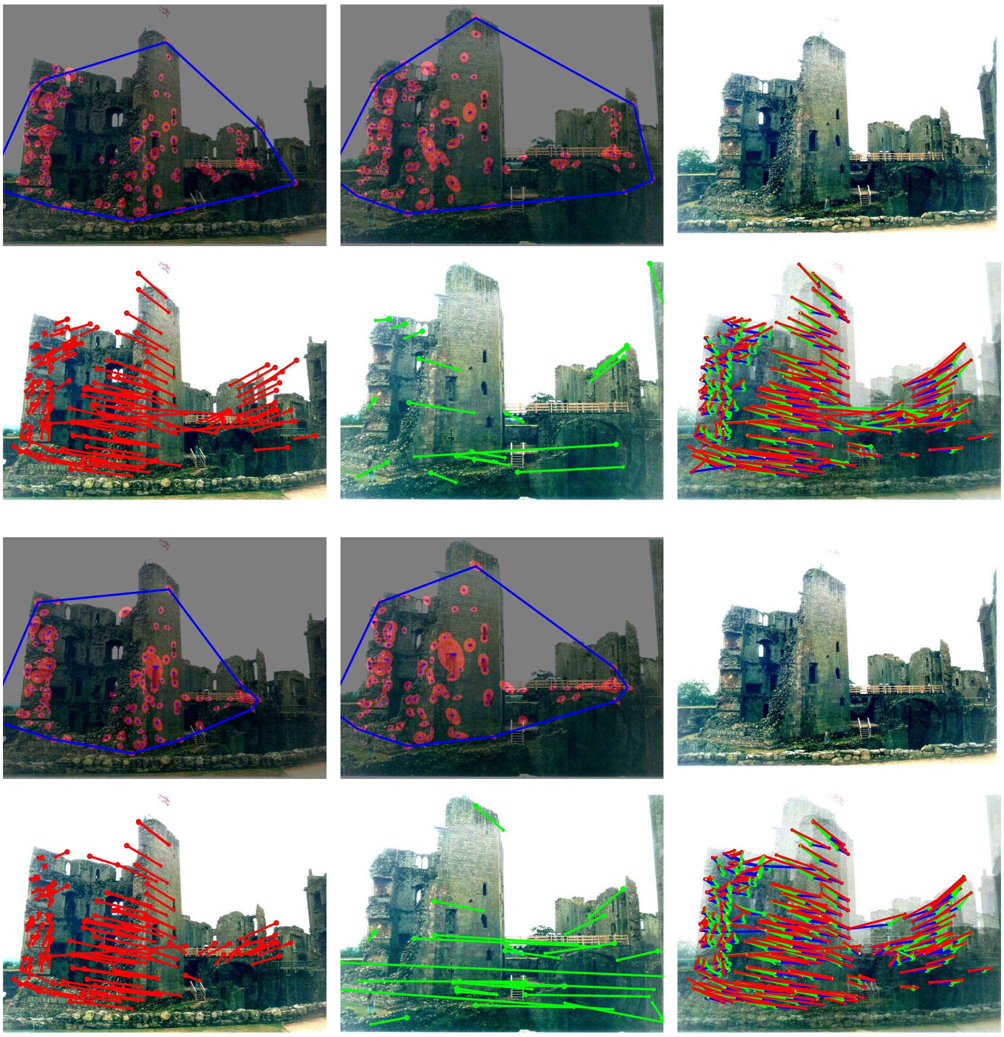


Figure 8.18: The shelf image sequence with the best results (see text for details)

Sequence: shelf
Best image pairs 0 1 with a 5 pixel chain error threshold

Groundtruth fund. matrix estimation pixel reprojection error

Image	1^{st} image				2^{nd} image			
pair	min	mean	max	std	min	mean	max	std
0–1	0.00	0.78	3.63	0.72	0.00	0.80	3.73	0.74
1–2	0.00	0.98	4.03	0.78	0.00	0.91	3.82	0.71
1–2	0.00	0.88	5.82	0.84	0.00	0.81	3.13	0.77

Algorithm statistics **Running time**

	$\mathcal{I}$(%)	$\mathcal{I}$(#)	$\mathcal{H}$(%)	$\mathcal{V}$(%)		$\mathcal{I}$(%)	$\mathcal{I}$(#)	$\mathcal{H}$(%)	$\mathcal{V}$(%)		time(s)		time(s)
SPARSE SOFT MATCHING													
mean	46.77	63	40.18	4.73	best	52.20	71	40.25	5.55	mean	754	max	973
std	2.47	4	2.28	0.47	worst	40.97	59	39.37	3.95	std	602	min	122
RANSAC WITH SYMMETRIC ERROR DISTANCE AND 9 PIXEL THRESHOLD													
mean	41.84	69	30.31	5.51	best	49.64	69	27.75	4.74	mean	45	max	52
std	4.67	10	5.45	0.70	worst	31.07	55	22.55	5.09	std	40	min	4
RANSAC WITH MAX ERROR DISTANCE AND 9 PIXEL THRESHOLD													
mean	39.59	68	32.93	5.21	best	47.69	93	38.74	6.70	mean	45	max	51
std	5.64	20	6.91	1.25	worst	30.20	45	28.27	3.36	std	41	min	3
MSAC WITH SAMPSON ERROR DISTANCE AND 8 PIXEL THRESHOLD													
mean	39.57	76	34.31	5.68	best	42.50	85	36.03	6.42	mean	46	max	51
std	1.84	6	4.32	0.70	worst	35.26	67	25.70	4.68	std	40	min	4
MSAC WITH MAX ERROR DISTANCE AND 9 PIXEL THRESHOLD													
mean	39.09	59	28.11	5.03	best	46.66	63	36.02	5.49	mean	45	max	51
std	4.45	17	7.96	1.07	worst	31.34	42	26.21	4.16	std	40	min	3
RANSAC WITH SAMPSON ERROR DISTANCE AND 9 PIXEL THRESHOLD													
mean	38.57	82	38.58	6.13	best	43.10	100	43.59	7.14	mean	46	max	51
std	4.47	16	4.26	0.93	worst	28.88	52	31.16	4.33	std	41	min	3
MLESAC WITH SYMMETRIC ERROR DISTANCE AND 3 PIXEL THRESHOLD													
mean	38.52	63	29.35	5.13	best	42.66	64	30.51	4.36	mean	46	max	52
std	2.11	3	4.16	0.44	worst	34.31	58	28.17	5.23	std	42	min	3
MLESAC WITH GEOMETRIC ERROR DISTANCE AND 4 PIXEL THRESHOLD													
mean	38.50	73	34.08	5.79	best	42.79	98	39.90	6.88	mean	47	max	53
std	3.28	12	3.69	0.60	worst	31.21	54	30.02	5.14	std	42	min	4
MLESAC WITH MAX ERROR DISTANCE AND 4 PIXEL THRESHOLD													
mean	38.19	83	38.36	6.01	best	44.95	98	43.59	7.09	mean	46	max	51
std	4.31	14	4.55	1.39	worst	30.33	54	30.22	3.14	std	40	min	3
MSAC WITH SYMMETRIC ERROR DISTANCE AND 7 PIXEL THRESHOLD													
mean	38.09	53	25.97	4.52	best	49.16	88	37.23	6.54	mean	45	max	51
std	6.13	15	9.72	1.02	worst	26.36	29	4.98	3.26	std	39	min	4
RANSAC WITH GEOMETRIC ERROR DISTANCE AND 8 PIXEL THRESHOLD													
mean	37.81	42	27.15	3.38	best	49.48	48	31.65	4.12	mean	45	max	51
std	6.62	8	5.94	1.05	worst	25.00	26	12.05	1.21	std	41	min	3
MSAC WITH GEOMETRIC ERROR DISTANCE AND 9 PIXEL THRESHOLD													
mean	37.20	47	28.86	3.69	best	50.93	82	43.38	6.65	mean	46	max	52
std	7.13	14	8.91	1.48	worst	25.22	28	18.04	1.73	std	40	min	4
MLESAC WITH SAMPSON ERROR DISTANCE AND 3 PIXEL THRESHOLD													
mean	36.60	71	33.74	5.46	best	44.00	99	43.09	6.85	mean	46	max	53
std	4.95	14	5.31	0.77	worst	29.44	58	30.52	4.45	std	41	min	3
LMEDS WITH MAX ERROR DISTANCE AND 9 PIXEL THRESHOLD													
mean	25.27	19	10.76	1.53	best	37.33	28	17.86	3.46	mean	45	max	51
std	7.64	7	6.36	0.75	worst	11.39	9	0.29	0.83	std	41	min	4
LMEDS WITH SYMMETRIC ERROR DISTANCE AND 5 PIXEL THRESHOLD													
mean	16.94	9	13.39	0.73	best	31.74	20	22.94	0.89	mean	45	max	52
std	7.82	4	7.73	0.38	worst	3.70	2	0.00	1.05	std	41	min	4
LMEDS WITH SAMPSON ERROR DISTANCE AND 7 PIXEL THRESHOLD													
mean	16.09	16	13.87	1.39	best	26.31	30	24.32	2.50	mean	46	max	53
std	4.57	5	6.49	0.56	worst	9.09	11	14.61	0.81	std	39	min	4
LMEDS WITH GEOMETRIC ERROR DISTANCE AND 5 PIXEL THRESHOLD													
mean	15.17	6	4.99	0.44	best	30.76	12	19.08	0.86	mean	45	max	52
std	6.47	3	5.88	0.22	worst	0.00	0	0.00	0.00	std	41	min	3

Table 8.7: Evaluation results on the shelf sequence (see the text for details)

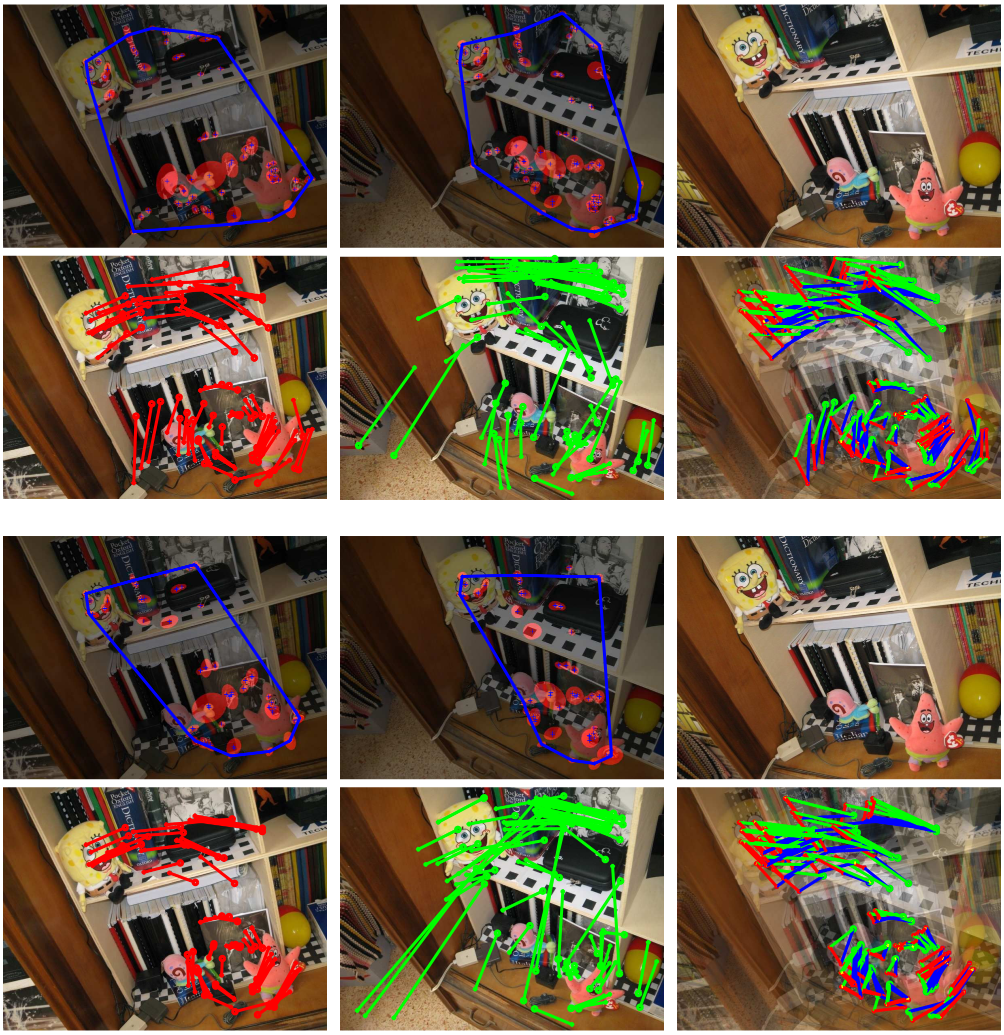


Figure 8.19: The Spongebob image sequence with the best results (see text for details)

Sequence: Spongebob
Best image pairs 0 2 with a 5 pixel chain error threshold

Groundtruth fund. matrix estimation pixel reprojection error

Image			1^{st} image		2^{nd} image			
pair	min	mean	max	std	min	mean	max	std
0–1	0.03	1.83	29.68	2.39	0.03	1.87	31.66	2.54
1–2	0.00	1.10	11.43	1.37	0.00	1.11	11.41	1.37
1–2	0.00	1.05	4.26	1.02	0.00	1.04	31.49	1.01

Algorithm statistics										**Running time**			
	$\mathcal{I}$(%)	$\mathcal{I}$(#)	$\mathcal{H}$(%)	$\mathcal{V}$(%)		$\mathcal{I}$(%)	$\mathcal{I}$(#)	$\mathcal{H}$(%)	$\mathcal{V}$(%)		time(s)		time(s)
ransac with geometric error distance and 4 pixel threshold													
mean	87.75	143	34.07	9.20	best	90.36	150	30.13	11.02	mean	65	max	117
std	1.87	12	2.91	1.49	worst	84.72	122	34.25	6.66	std	33	min	21
mlesac with max error distance and 2 pixel threshold													
mean	87.64	156	34.87	10.18	best	90.44	161	33.51	10.64	mean	47	max	84
std	2.81	15	1.22	1.41	worst	83.63	138	34.95	8.90	std	18	min	20
msac with sampson error distance and 2 pixel threshold													
mean	86.60	144	32.07	9.55	best	91.83	135	35.51	9.88	mean	70	max	116
std	2.83	14	3.25	1.16	worst	81.93	127	32.13	9.40	std	37	min	25
mlesac with geometric error distance and 2 pixel threshold													
mean	86.49	132	35.81	9.45	best	89.74	140	33.88	9.10	mean	72	max	109
std	3.16	26	3.59	1.19	worst	78.44	91	32.02	7.48	std	47	min	18
ransac with symmetric error distance and 2 pixel threshold													
mean	86.20	122	35.02	8.27	best	90.69	156	37.72	9.96	mean	99	max	123
std	2.79	19	2.68	1.20	worst	80.17	93	31.31	6.19	std	60	min	27
msac with max error distance and 2 pixel threshold													
mean	85.82	109	30.18	6.99	best	89.78	123	33.41	7.47	mean	100	max	123
std	3.21	18	7.54	0.84	worst	78.76	89	32.39	6.20	std	61	min	27
mlesac with sampson error distance and 1 pixel threshold													
mean	85.82	82	27.09	4.81	best	93.25	83	25.55	5.05	mean	104	max	127
std	5.59	17	5.50	1.19	worst	74.35	58	27.34	3.54	std	61	min	27
msac with symmetric error distance and 2 pixel threshold													
mean	85.51	109	27.66	7.66	best	92.05	139	31.41	8.72	mean	97	max	125
std	3.36	19	4.72	0.85	worst	80.18	89	31.52	7.72	std	60	min	27
ransac with max error distance and 3 pixel threshold													
mean	85.20	135	31.28	8.99	best	89.54	137	24.59	9.59	mean	73	max	125
std	2.79	19	5.41	1.04	worst	80.39	123	27.43	9.11	std	39	min	25
mlesac with symmetric error distance and 2 pixel threshold													
mean	85.10	150	34.69	9.73	best	86.70	137	30.92	8.47	mean	47	max	66
std	1.22	12	2.34	1.50	worst	83.15	158	37.17	10.76	std	21	min	19
msac with geometric error distance and 4 pixel threshold													
mean	84.08	120	30.08	7.44	best	87.67	128	35.84	7.11	mean	73	max	120
std	3.38	7	5.38	0.96	worst	78.94	120	23.55	7.90	std	35	min	20
ransac with sampson error distance and 3 pixel threshold													
mean	82.93	175	34.00	11.01	best	88.35	220	53.42	13.82	mean	49	max	103
std	2.67	22	6.14	1.68	worst	78.60	147	26.43	9.85	std	20	min	21
sparse soft matching													
mean	80.77	138	52.10	9.82	best	86.59	155	52.19	10.44	mean	736	max	1015
std	2.37	7	0.64	0.42	worst	77.90	134	51.67	9.69	std	524	min	129
lmeds with geometric error distance and 7 pixel threshold													
mean	80.26	155	37.60	10.45	best	84.61	220	52.78	13.80	mean	41	max	64
std	1.95	39	6.55	2.17	worst	76.86	103	36.58	7.33	std	15	min	17
lmeds with max error distance and 7 pixel threshold													
mean	80.09	189	39.25	11.96	best	84.37	189	37.92	10.72	mean	32	max	63
std	3.54	29	2.27	2.24	worst	74.01	151	36.53	10.35	std	8	min	22
lmeds with symmetric error distance and 3 pixel threshold													
mean	79.70	120	34.97	8.32	best	88.29	166	43.23	10.20	mean	63	max	94
std	8.44	39	8.40	1.70	worst	62.63	57	24.85	5.80	std	35	min	15
lmeds with sampson error distance and 7 pixel threshold													
mean	69.64	148	32.35	10.18	best	71.84	148	36.86	10.54	mean	30	max	96
std	2.10	22	4.79	1.85	worst	65.68	111	26.50	6.51	std	6	min	29

Table 8.8: Evaluation results on the Spongebob sequence (see the text for details)

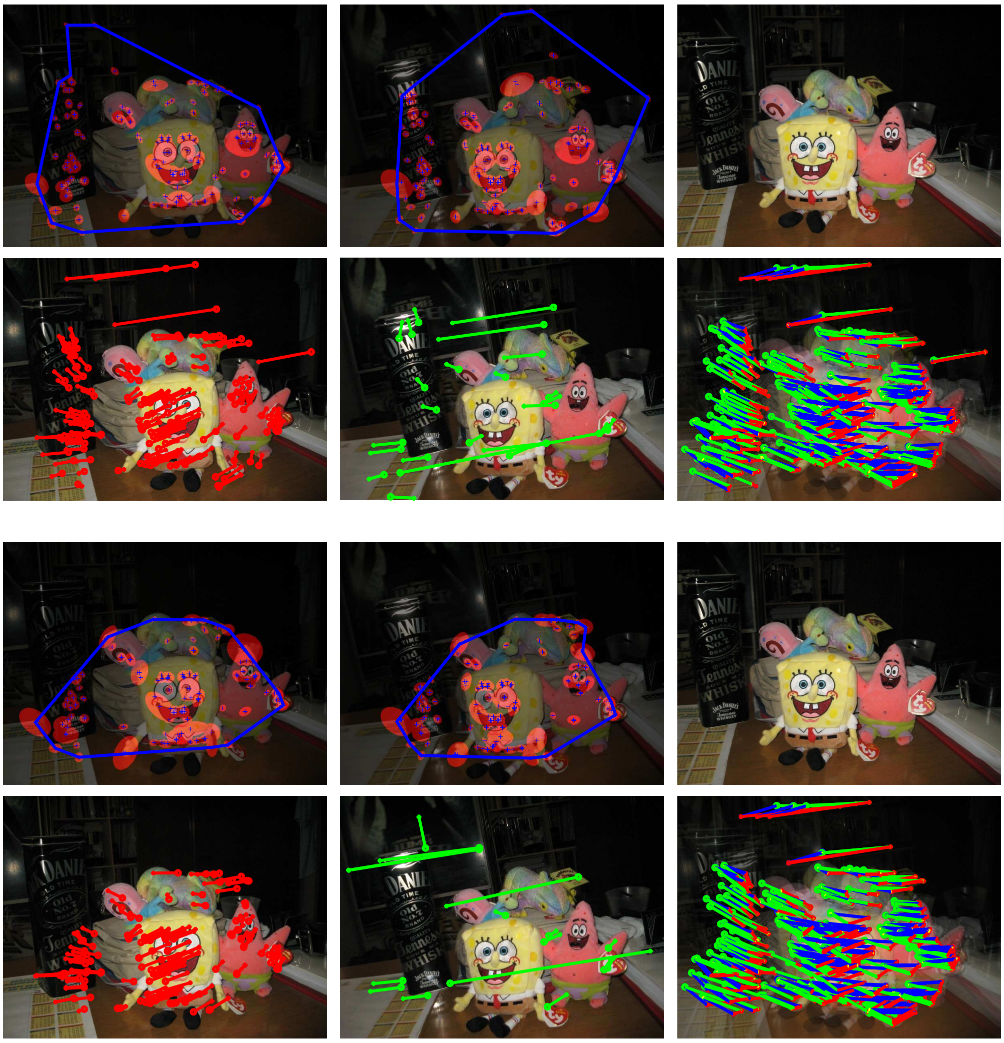


Figure 8.20: The ruins image sequence with the best results (see text for details)

Sequence: ruins
Best image pairs 0 2 with a 5 pixel chain error threshold

Groundtruth fund. matrix estimation pixel reprojection error

Image			1^{st} image		2^{nd} image			
pair	min	mean	max	std	min	mean	max	std
0–1	0.02	0.80	5.57	0.79	0.02	0.78	4.55	0.74
1–2	0.00	0.98	13.54	1.63	0.00	0.86	13.31	1.44
1–2	0.00	0.85	7.15	0.97	0.00	0.78	4.03	0.88

Algorithm statistics													**Running time**
	$\mathcal{I}(\%)$	$\mathcal{I}(\#)$	$\mathcal{H}(\%)$	$\mathcal{V}(\%)$		$\mathcal{I}(\%)$	$\mathcal{I}(\#)$	$\mathcal{H}(\%)$	$\mathcal{V}(\%)$		time(s)		time(s)
sparse soft matching													
mean	80.75	119	46.43	6.21	best	87.33	131	48.03	6.84	mean	1177	max	1858
std	3.48	10	1.34	0.48	worst	76.25	106	46.35	5.60	std	710	min	301
msac with max error distance and 5 pixel threshold													
mean	76.03	100	35.37	5.24	best	84.07	132	38.73	7.02	mean	81	max	101
std	5.50	17	4.42	1.01	worst	66.66	78	33.97	4.12	std	59	min	16
ransac with sampson error distance and 2 pixel threshold													
mean	75.01	78	23.98	4.38	best	85.29	87	31.41	5.17	mean	82	max	105
std	7.44	16	6.23	0.94	worst	61.95	57	20.63	3.14	std	58	min	17
msac with symmetric error distance and 4 pixel threshold													
mean	74.00	101	35.77	5.63	best	80.59	108	41.46	6.45	mean	86	max	154
std	3.78	9	5.42	0.80	worst	69.81	111	32.90	6.42	std	59	min	25
ransac with geometric error distance and 3 pixel threshold													
mean	73.51	60	24.91	3.65	best	86.55	103	40.75	5.91	mean	88	max	158
std	9.67	22	7.87	1.23	worst	48.27	28	11.89	1.94	std	57	min	29
msac with sampson error distance and 3 pixel threshold													
mean	72.63	96	26.09	5.27	best	77.35	123	34.37	6.27	mean	82	max	119
std	3.18	19	3.97	0.85	worst	67.70	65	26.73	3.83	std	58	min	19
msac with geometric error distance and 2 pixel threshold													
mean	72.47	53	22.36	2.95	best	87.59	113	35.14	6.18	mean	84	max	154
std	9.45	28	9.22	1.50	worst	52.08	25	7.30	1.33	std	60	min	25
ransac with max error distance and 4 pixel threshold													
mean	72.45	80	27.06	4.36	best	80.00	84	31.76	5.11	mean	81	max	102
std	6.35	16	6.64	0.73	worst	60.67	54	15.40	3.65	std	57	min	17
ransac with symmetric error distance and 4 pixel threshold													
mean	70.30	84	32.24	4.83	best	76.81	106	34.89	5.87	mean	82	max	100
std	6.33	14	6.61	0.74	worst	56.73	59	14.90	3.69	std	60	min	17
mlesac with sampson error distance and 2 pixel threshold													
mean	68.24	115	32.27	6.22	best	70.87	146	36.90	7.77	mean	77	max	103
std	2.28	18	4.18	0.93	worst	64.66	97	32.70	5.23	std	60	min	18
lmeds with symmetric error distance and 4 pixel threshold													
mean	67.04	89	28.08	4.71	best	75.18	100	31.36	4.95	mean	82	max	103
std	7.18	25	9.32	1.42	worst	50.58	43	10.95	2.08	std	59	min	17
mlesac with geometric error distance and 3 pixel threshold													
mean	66.03	96	30.31	5.06	best	73.29	129	43.01	6.72	mean	82	max	160
std	4.38	17	6.88	1.06	worst	60.56	86	22.19	4.42	std	53	min	28
mlesac with max error distance and 3 pixel threshold													
mean	65.79	124	34.50	6.65	best	70.17	160	42.67	8.51	mean	72	max	101
std	3.45	25	4.50	1.46	worst	60.13	86	31.25	4.34	std	47	min	21
mlesac with symmetric error distance and 2 pixel threshold													
mean	65.37	80	25.91	4.43	best	74.58	135	37.72	7.30	mean	88	max	151
std	6.27	32	9.22	1.77	worst	56.09	46	17.60	2.68	std	60	min	26
lmeds with max error distance and 6 pixel threshold													
mean	61.11	74	27.82	3.87	best	74.81	101	27.27	5.46	mean	82	max	100
std	8.65	20	4.45	1.18	worst	42.72	47	23.86	2.05	std	59	min	17
lmeds with geometric error distance and 4 pixel threshold													
mean	54.92	40	17.03	2.21	best	70.00	63	26.24	3.30	mean	86	max	161
std	11.41	13	7.37	0.76	worst	21.66	13	6.97	0.69	std	59	min	26
lmeds with sampson error distance and 5 pixel threshold													
mean	52.55	72	24.29	3.57	best	67.37	95	24.73	4.64	mean	82	max	107
std	12.71	39	6.15	2.14	worst	35.52	27	20.43	1.22	std	57	min	18

Table 8.9: Evaluation results on the ruins sequence (see the text for details)

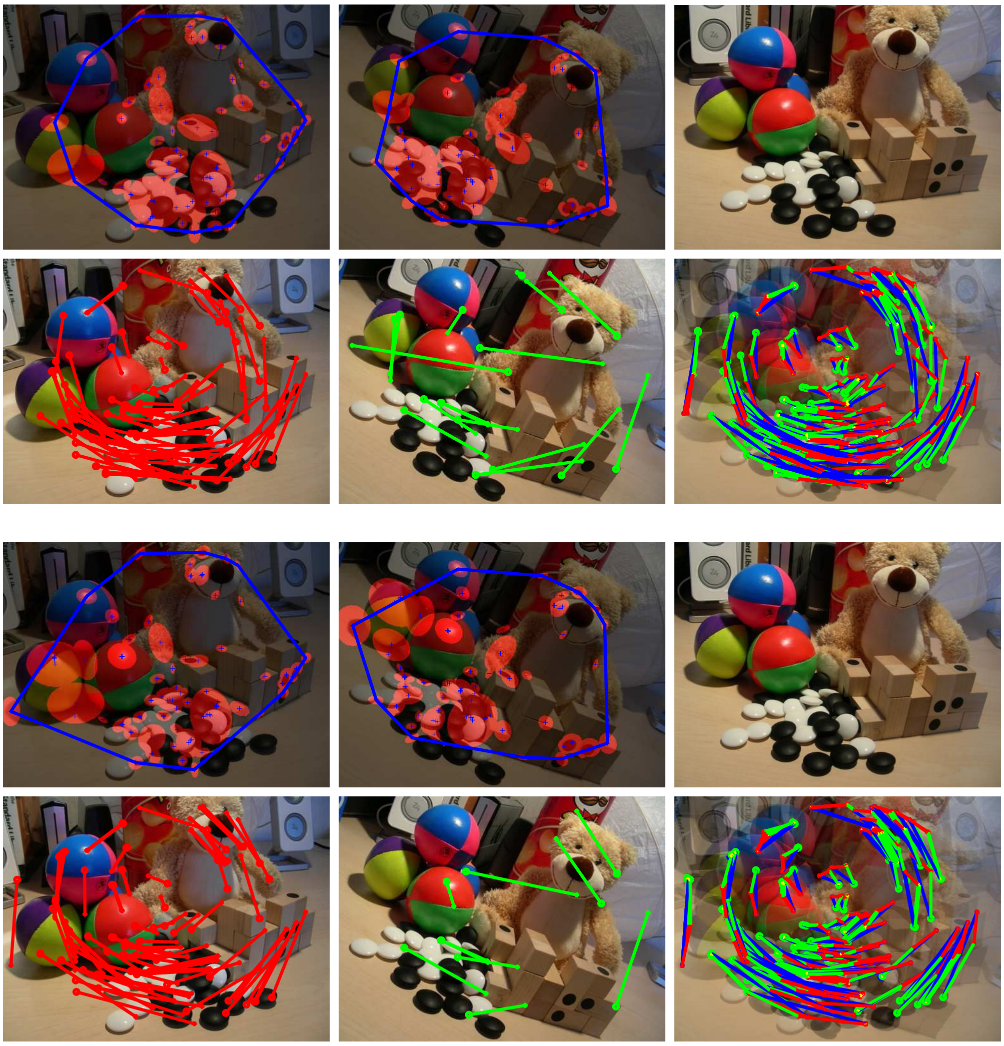


Figure 8.21: The Teddy image sequence with the best results (see text for details)

Sequence: Teddy
Best image pairs 0 1 with a 5 pixel chain error threshold

Groundtruth fund. matrix estimation pixel reprojection error

Image			1^{st} image		2^{nd} image			
pair	min	mean	max	std	min	mean	max	std
0–1	0.00	1.24	14.41	1.74	0.00	1.24	16.42	1.91
1–2	0.00	1.01	16.29	1.67	0.00	1.00	17.93	1.75
1–2	0.00	0.65	2.86	0.58	0.00	0.65	16.96	0.59

Algorithm statistics													**Running time**
	$\mathcal{I}(\%)$	$\mathcal{I}(\#)$	$\mathcal{H}(\%)$	$\mathcal{V}(\%)$		$\mathcal{I}(\%)$	$\mathcal{I}(\#)$	$\mathcal{H}(\%)$	$\mathcal{V}(\%)$		time(s)		time(s)
RANSAC WITH MAX ERROR DISTANCE AND 3 PIXEL THRESHOLD													
mean	84.32	82	44.15	15.23	best	90.29	93	51.60	17.88	mean	35	max	44
std	2.97	13	3.84	1.63	worst	78.66	59	40.41	15.67	std	17	min	8
RANSAC WITH GEOMETRIC ERROR DISTANCE AND 4 PIXEL THRESHOLD													
mean	84.29	84	39.58	15.21	best	86.72	98	45.66	17.47	mean	35	max	42
std	1.59	7	7.12	3.11	worst	80.95	85	46.29	19.40	std	21	min	8
MSAC WITH SYMMETRIC ERROR DISTANCE AND 4 PIXEL THRESHOLD													
mean	84.09	96	44.04	18.49	best	91.59	109	48.88	18.37	mean	31	max	43
std	4.06	9	4.27	1.77	worst	78.89	86	37.22	18.71	std	12	min	12
MSAC WITH GEOMETRIC ERROR DISTANCE AND 4 PIXEL THRESHOLD													
mean	83.93	74	38.59	15.18	best	86.20	75	40.83	15.01	mean	35	max	43
std	1.80	8	3.58	1.41	worst	80.23	69	38.98	17.02	std	16	min	8
MSAC WITH SAMPSON ERROR DISTANCE AND 3 PIXEL THRESHOLD													
mean	83.23	106	45.58	19.84	best	87.02	114	45.72	20.73	mean	32	max	44
std	3.96	12	7.28	2.44	worst	75.00	81	30.53	15.06	std	13	min	13
RANSAC WITH SAMPSON ERROR DISTANCE AND 2 PIXEL THRESHOLD													
mean	82.54	81	39.09	15.47	best	87.38	97	50.90	20.13	mean	34	max	43
std	2.78	9	10.51	2.23	worst	77.90	67	22.69	15.56	std	16	min	9
MSAC WITH MAX ERROR DISTANCE AND 5 PIXEL THRESHOLD													
mean	82.30	104	45.25	19.22	best	85.71	120	51.99	21.23	mean	30	max	43
std	2.32	11	4.88	1.89	worst	77.23	95	46.72	20.21	std	11	min	13
MLESAC WITH GEOMETRIC ERROR DISTANCE AND 2 PIXEL THRESHOLD													
mean	81.96	79	37.04	13.54	best	86.36	95	46.25	15.69	mean	35	max	43
std	3.44	7	7.35	1.82	worst	75.00	69	37.85	12.37	std	19	min	8
MLESAC WITH MAX ERROR DISTANCE AND 2 PIXEL THRESHOLD													
mean	81.87	90	46.13	17.21	best	86.44	102	50.93	18.92	mean	33	max	44
std	3.65	11	4.47	1.68	worst	74.44	67	34.23	13.55	std	11	min	12
RANSAC WITH SYMMETRIC ERROR DISTANCE AND 3 PIXEL THRESHOLD													
mean	81.79	82	38.58	15.33	best	88.67	94	50.88	20.50	mean	33	max	43
std	3.42	8	7.85	2.95	worst	75.82	69	35.67	13.60	std	16	min	9
MLESAC WITH SAMPSON ERROR DISTANCE AND 1 PIXEL THRESHOLD													
mean	81.38	45	33.59	10.62	best	86.79	46	25.94	10.92	mean	50	max	72
std	3.05	2	6.51	1.19	worst	72.72	40	34.01	9.94	std	38	min	13
MLESAC WITH SYMMETRIC ERROR DISTANCE AND 1 PIXEL THRESHOLD													
mean	80.16	37	27.02	7.31	best	89.58	43	33.34	7.92	mean	51	max	72
std	6.30	3	3.39	1.33	worst	67.34	33	26.32	7.37	std	36	min	14
SPARSE SOFT MATCHING													
mean	75.45	73	49.28	16.22	best	82.82	82	48.42	16.81	mean	159	max	239
std	3.26	4	2.83	0.41	worst	68.04	66	51.57	15.96	std	91	min	47
LMEDS WITH SYMMETRIC ERROR DISTANCE AND 7 PIXEL THRESHOLD													
mean	72.34	70	32.09	13.28	best	79.05	117	47.16	20.02	mean	30	max	44
std	3.63	23	9.54	3.70	worst	66.19	47	20.41	9.06	std	8	min	15
LMEDS WITH MAX ERROR DISTANCE AND 9 PIXEL THRESHOLD													
mean	71.41	84	39.75	15.99	best	73.94	105	42.63	18.83	mean	29	max	43
std	3.41	17	5.33	2.02	worst	64.70	55	29.42	12.93	std	6	min	16
LMEDS WITH GEOMETRIC ERROR DISTANCE AND 8 PIXEL THRESHOLD													
mean	71.35	63	32.56	12.60	best	77.30	109	46.68	19.61	mean	32	max	42
std	3.66	26	8.40	5.24	worst	62.71	74	35.91	16.57	std	11	min	12
LMEDS WITH SAMPSON ERROR DISTANCE AND 6 PIXEL THRESHOLD													
mean	69.13	70	38.95	14.31	best	77.27	85	48.43	16.17	mean	29	max	43
std	4.51	9	8.28	2.09	worst	61.62	53	41.86	10.82	std	6	min	16

Table 8.10: Evaluation results on the Teddy sequence (see the text for details)

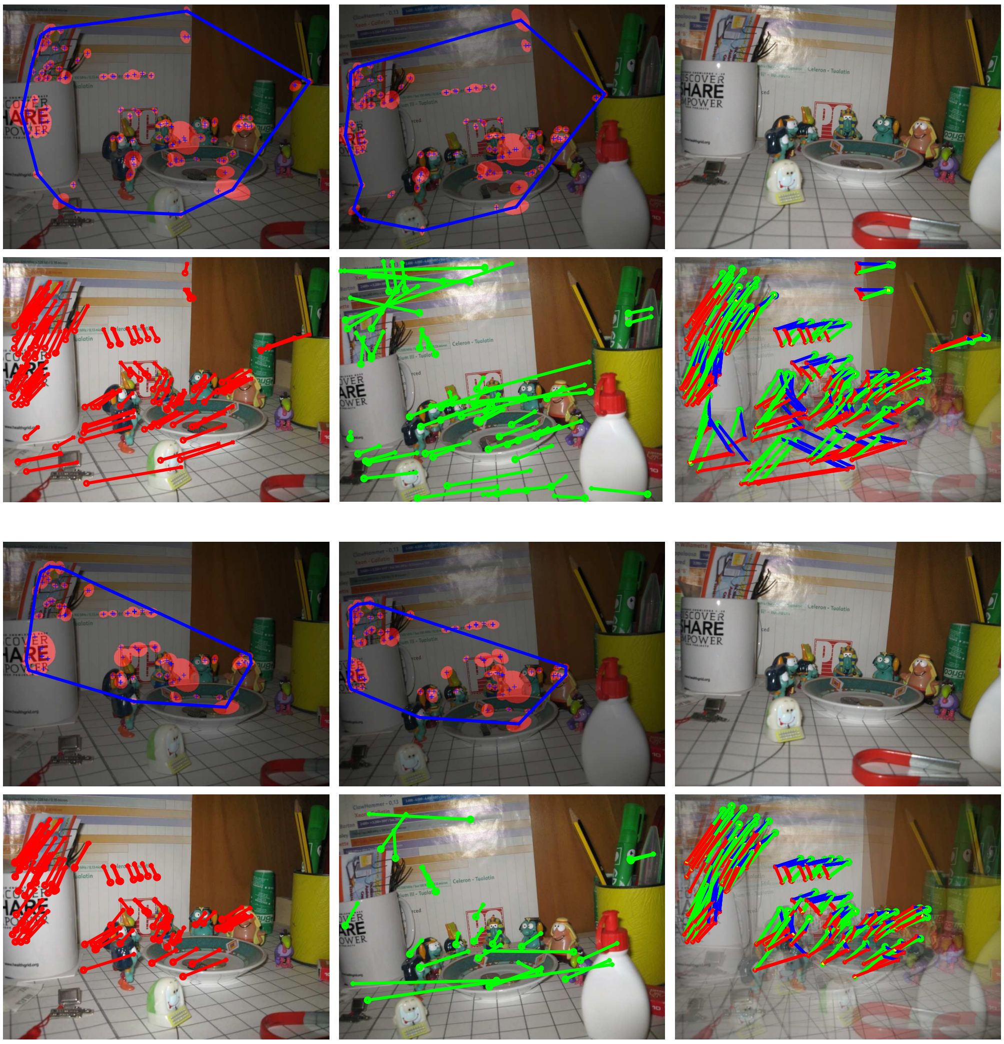


Figure 8.22: The tribal image sequence with the best results (see text for details)

Sequence: tribal
Best image pairs 0 2 with a 5 pixel chain error threshold

Groundtruth fund. matrix estimation pixel reprojection error

Image			1^{st} image		2^{nd} image			
pair	min	mean	max	std	min	mean	max	std
0–1	0.00	1.57	25.57	2.63	0.00	1.63	27.17	2.78
1–2	0.00	0.99	4.71	0.86	0.00	1.06	5.65	0.93
1–2	0.01	1.04	6.45	0.98	0.01	1.07	27.06	1.03

Algorithm statistics **Running time**

	$\mathcal{I}$(%)	$\mathcal{I}$(#)	$\mathcal{H}$(%)	$\mathcal{V}$(%)		$\mathcal{I}$(%)	$\mathcal{I}$(#)	$\mathcal{H}$(%)	$\mathcal{V}$(%)		time(s)		time(s)
MSAC WITH MAX ERROR DISTANCE AND 3 PIXEL THRESHOLD													
mean	66.12	78	26.64	5.28	best	77.86	95	24.10	6.47	mean	60	max	69
std	8.81	13	5.90	1.14	worst	47.24	60	19.34	3.95	std	52	min	5
MSAC WITH SYMMETRIC ERROR DISTANCE AND 3 PIXEL THRESHOLD													
mean	61.74	83	31.98	5.40	best	72.03	85	26.93	5.81	mean	59	max	68
std	4.99	21	8.06	1.14	worst	50.00	43	26.15	2.89	std	47	min	6
MLESAC WITH SAMPSON ERROR DISTANCE AND 2 PIXEL THRESHOLD													
mean	61.51	107	31.50	6.84	best	65.19	118	27.27	7.34	mean	61	max	70
std	3.73	13	5.84	0.78	worst	53.12	85	32.37	5.31	std	49	min	6
MSAC WITH SAMPSON ERROR DISTANCE AND 4 PIXEL THRESHOLD													
mean	61.39	96	28.98	6.42	best	66.66	82	18.40	5.49	mean	60	max	68
std	3.46	19	9.36	1.11	worst	56.14	96	35.03	6.51	std	49	min	5
RANSAC WITH SYMMETRIC ERROR DISTANCE AND 5 PIXEL THRESHOLD													
mean	60.77	97	29.42	6.58	best	65.36	117	33.44	7.43	mean	60	max	68
std	3.10	15	1.98	0.61	worst	53.78	71	25.56	5.75	std	48	min	5
MSAC WITH GEOMETRIC ERROR DISTANCE AND 7 PIXEL THRESHOLD													
mean	60.59	97	33.46	6.30	best	68.42	91	25.41	5.85	mean	61	max	69
std	5.22	24	8.68	1.34	worst	51.26	81	31.39	5.62	std	46	min	6
RANSAC WITH SAMPSON ERROR DISTANCE AND 3 PIXEL THRESHOLD													
mean	60.37	87	32.29	5.95	best	66.29	118	35.59	7.83	mean	61	max	69
std	4.99	20	7.05	1.34	worst	51.21	63	25.18	4.30	std	55	min	5
SPARSE SOFT MATCHING													
mean	60.37	94	47.88	6.55	best	64.59	104	53.75	7.70	mean	857	max	1107
std	2.76	4	4.39	0.51	worst	55.55	90	48.30	6.28	std	525	min	125
RANSAC WITH MAX ERROR DISTANCE AND 6 PIXEL THRESHOLD													
mean	60.06	94	31.55	6.26	best	66.94	81	25.01	5.53	mean	60	max	67
std	5.20	21	9.23	1.30	worst	46.45	59	24.70	4.21	std	54	min	5
RANSAC WITH GEOMETRIC ERROR DISTANCE AND 9 PIXEL THRESHOLD													
mean	59.38	105	34.35	6.88	best	65.23	137	38.13	8.78	mean	61	max	70
std	4.53	21	4.55	1.33	worst	51.89	82	33.62	5.49	std	55	min	5
MLESAC WITH GEOMETRIC ERROR DISTANCE AND 2 PIXEL THRESHOLD													
mean	58.43	69	25.00	4.45	best	69.29	79	26.69	5.23	mean	61	max	70
std	6.46	13	5.97	0.90	worst	45.91	45	15.25	3.41	std	49	min	7
MLESAC WITH MAX ERROR DISTANCE AND 2 PIXEL THRESHOLD													
mean	55.89	79	26.32	5.23	best	64.81	105	31.58	5.96	mean	61	max	71
std	6.04	18	10.61	0.95	worst	40.74	66	18.19	4.32	std	50	min	6
MLESAC WITH SYMMETRIC ERROR DISTANCE AND 2 PIXEL THRESHOLD													
mean	49.99	64	25.59	4.42	best	65.34	115	48.79	7.23	mean	61	max	71
std	9.51	26	10.01	1.64	worst	25.84	23	18.75	1.69	std	47	min	6
LMEDS WITH SYMMETRIC ERROR DISTANCE AND 7 PIXEL THRESHOLD													
mean	47.45	89	34.18	5.72	best	58.79	117	42.30	7.02	mean	60	max	70
std	5.66	17	7.81	0.87	worst	35.23	68	26.86	3.87	std	49	min	5
LMEDS WITH MAX ERROR DISTANCE AND 7 PIXEL THRESHOLD													
mean	47.41	80	32.11	5.48	best	54.82	125	49.41	7.20	mean	61	max	69
std	5.87	24	6.46	1.30	worst	35.19	44	24.40	2.87	std	55	min	4
LMEDS WITH SAMPSON ERROR DISTANCE AND 7 PIXEL THRESHOLD													
mean	43.94	91	33.56	5.95	best	54.22	109	41.81	7.00	mean	60	max	69
std	6.12	23	9.32	1.53	worst	32.65	64	27.63	4.82	std	49	min	6
LMEDS WITH GEOMETRIC ERROR DISTANCE AND 9 PIXEL THRESHOLD													
mean	41.61	60	21.19	4.25	best	58.65	105	35.86	7.07	mean	61	max	69
std	8.99	22	8.28	1.46	worst	30.70	39	16.38	2.73	std	55	min	5

Table 8.11: Evaluations result on the tribal sequence (see the text for details)

Chapter 9

Conclusions and future works

In this thesis the following contributes in field of the image feature matching have been presented:

- the *HarrisZ detector*, an improved affine feature detector based on the Harris corner detector. It provides stable and robust results in terms of the repeatability index and the matching score, without the requirement of a fine tuning of the algorithm parameters. The results obtained for planar and three dimensional objects are comparable with the state of the art affine detectors, such as the Hessian-affine detector and the MSER detector. Though the method is not fast, it is still appropriate for off-line tasks which require high accuracy;
- the *sGLOH descriptor*, an extension of the GLOH descriptor. By using the proposed descriptor, the similarity between two features can be checked not only in a predefined orientation, the dominant gradient orientation, but also according to a set of discrete rotations. This can be achieved by shifting the descriptor vector with a reasonable computational cost and by using an improved feature distance. The proposed descriptor has been compared with the SIFT and the GLOH descriptors on the Oxford image dataset, and shows good results which point out its robustness and stability;
- a RANSAC-based matching algorithm, named *sparse soft matching.* It achieves an image-guided selection of the error threshold and uses a soft matching strategy in contrast to the one-to-one matching required by RANSAC, which increases the absolute number of matches. Moreover it performs a less random choice of candidate matches, according to a global-to-local constrain generation. The final matches are homogeneously distributed along the image, resulting in a more stable estimation of the homography or the fundamental matrix. As weak point, it is computationally more expensive than RANSAC;
- a validation framework to test feature detectors, descriptors and matching algorithms has been also proposed. It is effective, uses only geometric information and can obtain the ground truth data very easy.

All the proposed algorithms are promising. Future works will include a faster approximation to improve the HarrisZ detectors, as well as further evaluations. The sGLOH descriptor is robust, and an extension to include distances based on the EMD could be interesting. The proposed soft sparse matching generates an high number of correct matches and an high coverage of the image, however some aspects have to be further investigated. In particular the use of a fast model check to reduce the running time. Lastly, the proposed validation framework can be used to test and validate not only other matching strategies but also feature detectors and descriptors. The framework can be also extended to include other measures and further information about the features, such as their patch shapes.

Bibliography

[1] J. Babaud, A.P. Witkin, M. Baudin, and R.O. Duda. Uniqueness of the gaussian kernel for scale-space filtering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(1):26–33, 1986. 9

[2] W. Banzhaf, P. Nordin, R.E. Keller, and F.D. Francone. *Genetic Programming - An Introduction On the Automatic Evolution of Computer Programs and Its Applications*. Morgan Kaufmann Publishers, 1997. 33

[3] A. Baumberg. Reliable feature matching across widely separated views. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages 774–781 vol.1. IEEE Computer Society, 2000. 8, 12, 23, 30, 47

[4] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool. Speeded-up robust features (surf). *Computer Vision and Image Understanding*, 110(3):346–359, 2008. 31, 32, 48, 71

[5] P.R. Beaudet. Rotationally invariant image operators. In *International Joint Conference on Pattern Recognition*, pages 579–583, 1978. 13, 24, 30

[6] F. Bellavia, M. Cipolla, D. Tegolo, and C. Valenti. An evolution of the non-parametric Harris affine corner detector: A distributed approach. In *International Conference on Parallel and Distributed Computing, Applications and Technologies*, pages 18–25, 2009. 71

[7] F. Bellavia, G. Gagliano, D. Tegolo, and C. Valenti. Global archaelogical mosaicing for underwater scenes. *WSEAS Transactions on Signal Processing*, 2(7):997–1004, 2006. 88

[8] F. Bellavia, D. Tegolo, and E. Trucco. Improving SIFT-based descriptors stability to rotations. In *International Conference on Pattern Recognition*, 2010. 65

[9] F. Bellavia, D. Tegolo, and C. Valenti. A non-parametric scale-based corner detector. In *International Conference on Pattern Recognition*, pages 1–4, 2008. 65

[10] F. Bellavia, D. Tegolo, and C. Valenti. Improving Harris corner selection strategy. *IET Computer Vision*, to appear. 65

[11] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(4):509–522, 2002. 5, 42

[12] A.C. Berg, T.L. Berg, and J. Malik. Shape matching and object recognition using low distortion correspondences. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages 26–33. IEEE Computer Society, 2005. 45, 46

[13] A.C. Berg and J. Malik. Geometric blur for template matching. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages 607–614, 2001. 45

[14] M. Berthold and D.J. Hand, editors. *Intelligent Data Analysis: An Introduction*. Springer-Verlag, Secaucus, NJ, USA, 1999. 103

[15] J. Bigün. A structure feature for some image processing applications based on spiral functions. *Computer Vision, Graphics and Image Processing*, 51(2):166–194, 1990. 21

[16] T. Botterill, S. Mills, and R. Green. New conditional sampling strategies for speeded-up RANSAC. In *British Machine Vision Conference*, 2009. 96, 98

[17] M. Brown and D.G. Lowe. Invariant features from interest point groups. In *British Machine Vision Association*, 2002. 14

[18] M. Brown, R. Szeliski, and S. Winder. Multi-image matching using multi-scale oriented patches. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages 510–517. IEEE Computer Society, 2005. 21, 23

[19] M.C. Burl, M. Weber, and P. Perona. A probabilistic approach to object recognition using local photometry and global geometry. *Lecture Notes in Computer Science*, 1407, 1998. 18

[20] P.J. Burt and E.H. Adelson. The laplacian pyramid as a compact image code. *Readings in computer vision: issues, problems, principles, and paradigms*, pages 671–679, 1987. 11, 31, 67

[21] J. Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(6):679–698, 1986. 34

[22] D.P. Capel. An effective bail-out test for RANSAC consensus scoring. In *Proceedings of British Machine Vision Conference*, 2005. 101

[23] G. Carneiro. A comparative study on the use of an ensemble of feature extractors for the automatic design of local image descriptors. In *International Conference on Pattern Recognition*. IEEE Computer Society, 2010. 51

[24] H. Chen, H. Chang, and T. Liu. Local discriminant embedding and its variants. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 846–853. IEEE Computer Society, 2005. 43, 51

[25] H. Chen and P. Meer. Robust regression with projection based m-estimators. In *IEEE International Conference on Computer Vision*, page 878, Washington, DC, USA, 2003. IEEE Computer Society. 102

[26] O. Chum. *Two-View Geometry Estimation by Random Sample and Consensus.* PhD thesis, Center for Machine Perception, Czech Technical University in Prague, 2005. 98, 102

[27] O. Chum and J. Matas. Geometric hashing with local affine frames. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 879–884. IEEE Computer Society, 2006. 16, 29

[28] O. Chum, J. Matas, and J. Kittler. Locally optimized ransac. In *Deutsche Arbeitsgemeinschaft für Mustererkennung Symposium for Pattern Recognition*, pages 236–243, 2003. 98, 104

[29] O. Chum, T. Werner, and J. Matas. Two-view geometry estimation unaffected by a dominant plane. In *Computer Vision and Pattern Recognition*, pages 772–779, 2005. 98, 100, 101

[30] R. Cipolla and P. Giblin. *Visual motion of curves and surfaces.* Cambridge University Press, 2000. 24, 25

[31] T.H. Cormen, C. Stein, R.L. Rivest, and C.E. Leiserson. *Introduction to Algorithms.* McGraw-Hill Higher Education, 2001. 5, 24, 42, 53, 70

[32] C. Cortes and V. Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995. 51

[33] T.M. Cover and J.A. *Elements of information theory.* Wiley, 1991. 49

[34] D. Crandall, P.F. Felzenszwalb, and D.P. Huttenlocher. Spatial priors for part-based recognition using statistical models. In *Computer Vision and Pattern Recognition*, pages 10–17, 2005. 18

[35] J.L. Crowley, O.R., and J.H. Piater. Fast computation of characteristic scale using a half-octave pyramid. In *International Conference on Scale-Space theories in Computer Vision*, 2002. 11, 31, 67

[36] G. Csurka, C.R. Dance, L. Fan, J. Willamowski, and C. Bray. Visual categorization with bags of keypoints. In *ECCV Workshop on Statistical Learning in Computer Vision*, pages 1–22, 2004. 5, 16

[37] Y. Cui, N. Hasler, T. Thormählen, and H. Seidel. Scale invariant feature transform with irregular orientation histogram binning. In *International Conference on Image Analysis and Recognition*, pages 258–267. Springer-Verlag, 2009. 43, 44, 64

[38] M. de Berg, M. van Kreveld, M. Overmars, and O. Schwarzkopf. *Computational Geometry: Algorithms and Applications*. Springer-Verlag, 2000. 112

[39] R.J.M. den Hollander and A. Hanjalic. A combined RANSAC-hough transform algorithm for fundamental matrix estimation. In *British Machine Vision Conference*, 2007. 102

[40] R. Deriche and G. Giraudon. Accurate corner detection: An analytical study. In *International Conference on Computer Vision*, pages 66–70, 1990. 25, 26

[41] Timo Dickscheid and Wolfgang Förstner. Evaluating the suitability of feature detectors for automatic image orientation systems. In *Computer Vision Systems*, volume 5815 of *Lecture Notes in Computer Science*, pages 305–314. Springer, 2009. 6, 62, 63

[42] A. Djouadi, Ö. Snorrason, and F.D. Garber. The quality of training sample estimates of the bhattacharyya coefficient. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(1):92–97, 1990. 50

[43] J. Duchon. Splines minimizing rotation-invariant semi-norms in sobolev spaces. In *Constructive Theory of Functions of Several Variables*, pages 85–100. Springer Link, 1977. 42

[44] O. Faugeras. *Three-Dimensional Computer Vision – A Geometric Viewpoint*. MIT Press, 1996. 5, 85, 86, 91

[45] P.F. Felzenszwalb and D.P. Huttenlocher. Pictorial structures for object recognition. *International Journal of Computer Vision*, 61(1):55–79, 2005. 18

[46] R. Fergus, P. Perona, and A. Zisserman. Weakly supervised scale-invariant learning of models for visual recognition. *International Journal of Computer Vision*, 71(3):273–303, 2007. 3

[47] R. Fergus, P. Perona, and A. Zisserman. Weakly supervised scale-invariant learning of models for visual recognition. *International Journal of Computer Vision*, 71(3):273–303, 2007. 18

[48] V. Ferrari, T. Tuytelaars, and L.J. Van Gool. Wide-baseline multiple-view correspondences. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 718–728, 2003. 85

[49] V. Ferrari, T. Tuytelaars, and L.J. Van Gool. Simultaneous object recognition and segmentation by image exploration. In *European Conference on Computer Vision*, pages 40–54, 2004. 16

[50] V. Ferrari, T. Tuytelaars, and L.J. Van Gool. Wide-baseline stereo matching with line segments. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 329–336, 2005. 19

[51] M.A. Fischler and R.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. 5, 86, 95

[52] L. Florack. *The Syntactical Structure of Scalar Images*. PhD thesis, Universiteit Utrecht, 1993. 9

[53] P. Forssén. Maximally stable colour regions for recognition and matching. In *IEEE Conference on Computer Vision and Pattern Recognition*. IEEE Computer Society, 2007. 29

[54] P. Forssén and D.G. Lowe. Shape descriptors for maximally stable extremal regions. In *International Conference on Computer Vision*. IEEE Computer Society, 2007. 6, 29, 60, 61

[55] W. Förstner, T. Dickscheid, and F. Schindler. Detecting interpretable and accurate scale-invariant keypoints. In *International Conference on Computer Vision*, pages 2256–2263, 2009. 21

[56] M. A. Föstner and E. Gülch. A fast operator for detection and precise location of distinct points, corners and centers of circular features. In *ISPRS Intercommission Workshop*, 1987. 12, 21, 26

[57] J.M. Frahm and M. Pollefeys. RANSAC for (quasi-)degenerate data (QDEGSAC). In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 453–460, Washington, DC, USA, 2006. IEEE Computer Society. 98, 101

[58] F. Fraundorfer. Local detector evaluation, 2010. `http://www.icg.tu-graz.ac.at/Members/fraunfri/local-detector-evaluation`. 57, 60

[59] F. Fraundorfer and H. Bischof. A novel performance evaluation method of local detectors on non-planar scenes. In *IEEE Conference on Computer Vision and Pattern Recognition*, page 33. IEEE Computer Society, 2005. 5, 6, 57, 62, 63, 71

[60] F. Fraundorfer, M. Winter, and H. Bischof. MSCC: Maximally stable corner clusters. In *Scandinavian Conference on Image Analysis*, pages 45–54, 2005. 24, 25

[61] D. Freedman, R. Pisani, and R. Purves. *Statistics*. W. W. Norton & Company, 2007. 99

[62] W.T. Freeman and E.H. Adelson. The design and use of steerable filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13:891–906, 1991. 4, 46, 47

[63] A. Fusiello, E. Trucco, and A. Verri. A compact algorithm for rectification of stereo pairs. *Machine Vision and Applications*, 12(1):16–22, 2000. 85, 91

[64] R. Gherardi, M. Farenzena, and A. Fusiello. Improving the efficiency of hierarchical structure-and-motion. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2010. 85

[65] M. Goesele, N. Snavely, B. Curless, H. Hoppe, and S.M. Seitz. Multi-view stereo for community photo collections. In *International Conference on Computer Vision*, pages 1–8, 2007. 85

[66] R.C. Gonzalez and R.E. Woods. *Digital Image Processing*. Prentice-Hall, 2006. 18, 39, 99, 102

[67] K. Grauman and T. Darrell. The pyramid match kernel: Efficient learning with sets of features. *The Journal of Machine Learning Research*, 8:725–760, 2007. 5, 17

[68] A. Haja, B. Jahne, and S. Abraham. Localization accuracy of region detectors. In *Computer Vision and Pattern Recognition*, pages 1–8, 2008. 57, 63

[69] C. Harris and M. Stephens. A combined corner and edge detector. In *Alvey Vision Conference*, pages 147–151, 1988. 3, 4, 12, 19, 20

[70] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2000. 5, 6, 7, 60, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 100, 101, 111

[71] T. Hastie, R. Tibshirani, and J.H. Friedman. *The Elements of Statistical Learning*. Springer, 2003. 97

[72] G. Hua, M. Brown, and S. Winder. Discriminant embedding for local image descriptors. *International Conference on Computer Vision,*, 0:1–8, 2007. 43, 51, 64

[73] B. Huet and E.R. Hancock. Cartographic indexing into a database of remotely sensed images. In *IEEE Workshop on Applications of Computer Vision*, page 8. IEEE Computer Society, 1996. 50

[74] A. K. Jain and R.C. Dubes. *Algorithms for clustering data*. Prentice-Hall, 1988. 11, 16, 40, 43, 52, 62, 65

[75] A. Johnson. *Spin-Images: A Representation for 3-D Surface Matching*. PhD thesis, Robotics Institute, Carnegie Mellon University, 1997. 41

[76] T. Kadir and M. Brady. Saliency, scale and image description. *International Journal of Computer Vision*, 45(2):83–105, 2001. 30, 34, 37

[77] T. Kadir, A. Zisserman, and M. Brady. An affine invariant salient region detector. In *European Conference Of Computer Vision*, pages 228–241, 2004. 36

[78] Y. Ke and R. Sukthankar. PCA-SIFT: a more distinctive representation for local image descriptors. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 506–513, 2004. 43

[79] M.G. Kendall. A new measure of rank correlation. *Biometrika*, 30(1/2):81–93, 1938. 54

[80] C.S. Kenney, M. Zuliani, and B.S. Manjunath. An axiomatic approach to corner detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages 191–197. IEEE Computer Society, 2005. 63

[81] L. Kitchen and A. Rosenfeld. Edge evaluation using local edge coherence. *IEEE Transactions on Systems, Man, and Cybernetics*, 1981. 26

[82] J.J. Koenderink. The structure of images. *Biological Cybernetics*, 50(5):363–370–370, 1984. 9

[83] J.J. Koenderink and A.J. van Doom. Representation of local geometry in the visual system. *Biological Cybernetics*, 55(6):367–375, 1987. 8, 46, 47

[84] A. Konouchine, V. Gaganov, and V. Veznevets. AMLESAC: A new maximum likelihood robust estimator. In *International Conference on Computer Graphics and Vision*, 2005. 97

[85] S. Lazebnik, C. Schmid, and J. Ponce. A sparse texture representation using local affine regions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(8):1265–1278, 2005. 5, 41, 43, 45, 78

[86] S. Lazebnik, C. Schmid, and J. Ponce. A sparse texture representation using local affine regions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(8):1265–1278, 2005. 5

[87] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2169–2178. IEEE Computer Society, 2006. 5, 16, 17

[88] Y. LeCun and Y. Bengio. Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks*, pages 255–258, 1998. 18

[89] M. Lhuillier and L. Quan. Match propogation for image-based modeling and rendering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(8):1140–1146, 2002. 85

[90] K. Lin and C. Yang. The ann-tree: An index for efficient approximate nearest neighbor search. In *International Conference on Database Systems for Advanced Applications*, pages 174–181. IEEE Computer Society, 2001. 5

[91] T. Lindeberg. Scale-space for discrete signals. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(3):234–254, 1990. 9

[92] T. Lindeberg. Detecting salient blob-like image structures and their scales with a scale-space primal sketch: A method for focus-of-attention. *International Journal of Computer Vision*, 11:283–318, 1993. 29

[93] T. Lindeberg. Junction detection with automatic selection of detection scales and localization scales. In *International Conference on Image Processing*, volume 1, pages 924–928. IEEE Computer Society Press, 1994. 26

[94] T. Lindeberg. *Scale-Space Theory in Computer Vision*. Kluwer Academic Publishers, 1994. 9, 10, 11

[95] T. Lindeberg. Feature detection with automatic scale selection. *International Journal of Computer Vision*, 30(2):79–116–116, 1998. 11, 23, 30

[96] T. Lindeberg and J. Gårding. Shape-adapted smoothing in estimation of 3-d depth cues from affine distortions of local 2-d brightness structure. In *European conference on Computer vision*, volume 2, pages 389–400. Springer-Verlag, 1994. 8

[97] H. Ling and K. Okada. Diffusion distance for histogram comparison. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 246–253. IEEE Computer Society, 2006. 5, 53, 64

[98] H. Ling and S. Soato. Proximity distribution kernels for geometric context in category recognition. In *IEEE International Conference on Computer Vision*, 2007. 17

[99] D.G. Lowe. Object recognition from local scale-invariant features. In *International Conference on Computer Vision*, volume 2, page 1150. IEEE Computer Society, 1999. 3, 16

[100] D.G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004. 4, 5, 8, 32, 33, 40, 42, 43, 61, 71, 96

[101] B. D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *International Joint Conference on Artificial Intelligence*, pages 674–679, 1981. 19

[102] Q. Luong and O.D. Faugeras. The fundamental matrix: Theory, algorithms, and stability analysis. *International Journal of Computer Vision*, 17(1):43–75, 1996. 89

[103] D. Marr and T. Poggio. Cooperative computation of stereo disparity. *Science*, 194:283–287, 1976. 3

[104] J. Matas and O. Chum. Randomized RANSAC with t*, d* test. *Image and Vision Computing*, 22(10):837–842, 2004. 98, 101

[105] J. Matas and O. Chum. Matching with prosac - progressive sample consensus. In *IEEE Conference on Computer Vision and Pattern Recognition*. IEEE Computer Society, 2005. 98, 99, 117

[106] J. Matas and O. Chum. Randomized RANSAC with sequential probability ratio test. In *International Conference on Computer Vision*, pages 1727–1732, 2005. 98, 101

[107] J. Matas, O. Chum, U. Martin, and T. Pajdla. Robust wide baseline stereo from maximally stable extremal regions. In *British Machine Vision Conference*, volume 1, pages 384–393, 2002. 6, 28, 30

[108] J. Matas, S. Obdrzalek, and O. Chum. Local affine frames for wide-baseline stereo. In *International Conference on Pattern Recognition*, volume 4, page 40363. IEEE Computer Society, 2002. 8, 29, 30

[109] M.A. Mattar, A.R. Hanson, and E.G. Learned-Miller. Sign classification using local and meta-features. In *IEEE Conference on Computer Vision and Pattern Recognition - Workshops*, page 26, Washington, DC, USA, 2005. IEEE Computer Society. 96

[110] A. Meler, M. Decrouez, and J.L. Crowley. BetaSAC: A new conditional sampling for RANSAC. In *International Conference on Computer Vision*, 2010. 98, 99

[111] K. Mikolajczyk. *Detection of local features invariant to affine transformations*. PhD thesis, INPG, 2002. 11, 14, 24, 40, 47, 49

[112] K. Mikolajczyk and C. Schmid. Scale & affine invariant interest point detectors. *International Journal of Computer Vision*, 60(1):63–86, 2004. 5, 6, 8, 12, 13, 19, 20, 23, 30, 69, 71, 96

[113] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(10):1615–1630, 2005. 5, 8, 15, 40, 43, 63, 64, 79, 81

[114] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. Van Gool. A comparison of affine region detectors. *C*, 65(1-2):43–72, 2005. 4, 5, 6, 8, 28, 36, 56, 57, 63, 70, 71

[115] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. Van Gool. Affine covariant features, 2010. `http://www.robots.ox.ac.uk/~vgg/research/affine`. 57, 58, 59

[116] F. Mindru, T. Moons, and L. Van Gool. Recognizing color patterns irrespective of viewpoint and illumination. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 368–373, 1999. 45

[117] M. Mitchell. *An Introduction to Genetic Algorithms*. MIT Press, Cambridge, MA, USA, 1998. 102

[118] H. Moravec. Towards automatic visual obstacle avoidance. In *International Joint Conference on Artificial Intelligence*, 1977. 12, 19

[119] P. Moreels and P. Perona. Evaluation of features detectors and descriptors based on 3d objects. *International Journal of Computer Vision*, 73(3):263–284, 2007. 6, 61, 63, 103, 112, 113

[120] J.M. Morel and G. Yu. ASIFT: A new framework for fully affine invariant image comparison. *SIAM Journal on Imaging Sciences*, 2(2):438–469, 2009. 33

[121] P. Moreno, A. Bernardino, and J. Santos-Victor. Improving the SIFT descriptor with smooth derivative filters. *Pattern Recognition Letters*, 30(1):18–26, 2009. 43, 64

[122] J. Munkres. Algorithms for the assignment and transportation problems. *Journal of the Society for Industrial and Applied Mathematics*, 5(1):32–38, 1957. 52, 53

[123] D.R. Myatt, P.H.S. Torr, S.J. Nasuto, J.M. Bishop, and R. Craddock. Napsac: High noise, high dimensional robust estimation - it's in the bag. In *British Machine Vision Conference*, 2002. 98

[124] K. Ni, H. Jin, and F. Dellaert. GroupSAC: Efficient consensus in the presence of groupings. In *International Conference on Computer Vision*. IEEE Computer Society, 2009. 98, 99, 101, 102

[125] C.W. Niblack, R.J. Barber, W.R. Equitz, M.D. Flickner, D. Glasman, D. Petkovic, and P.C. Yanker. The qbic project: Querying image by content using color, texture, and shape. In *SPIE Conference on Storage and Retrieval for Image and Video Databases*, volume 1908, pages 173–187, 1993. 52

[126] D. Nistér. Preemptive RANSAC for live structure and motion estimation. In *IEEE International Conference on Computer Vision*, page 199, Washington, DC, USA, 2003. IEEE Computer Society. 102

[127] D. Nistér. An efficient solution to the five-point relative pose problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(6):756–777, 2004. 94

[128] D. Nistér, O. Naroditsky, and J.R. Bergen. Visual odometry for ground vehicle applications. *Journal of Field Robotics*, pages 3–20, 2006. 96, 97

[129] D. Nistér and H. Stewenius. Scalable recognition with a vocabulary tree. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2161–2168. IEEE Computer Society, 2006. 15, 16

[130] L. Parida, D. Geiger, and R. Hummel. Kona: A multi-junction detector using minimum description length principle. In *Computer Vision and Pattern Recognition*, volume 1223, pages 51–65, 1997. 26, 27

[131] O. Pele and M. Werman. A linear time histogram metric for improved SIFT matching. In *European Conference on Computer Vision*, pages 495–508. Springer-Verlag, 2008. 53, 64

[132] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. Object retrieval with large vocabularies and fast spatial matching. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2007. 15, 16

[133] M. Pollefeys, R. Koch, and L. Van Gool. A simple and efficient rectification method for general motion. In *International Conference on Computer Vision*, pages 496–501, 1999. 85, 91

[134] J. Ponce, T. L. Berg, M. Everingham, D. A. Forsyth, M. Hebert, S. Lazebnik, M. Marszalek, C. Schmid, B. C. Russell, A. Torralba, C. K. I. Williams, J. Zhang, and A. Zisserman. Dataset issues in object recognition. In *Toward Category-Level Object Recognition*, pages 29–48. Springer, 2006. 3

[135] R. Raguram, J.M. Frahm, and M. Pollefeys. A comparative analysis of RANSAC techniques leading to adaptive real-time random sample consensus. In *European Conference on Computer Vision*, pages 500–513, Berlin, Heidelberg, 2008. Springer-Verlag. 98, 101, 102

[136] R. Raguram, J.M. Frahm, and M. Pollefeys. Exploiting uncertainty in random sample consensus. In *International Conference on Computer Vision*, pages 47–52, 2009. 98

[137] M. Riesenhuber and T. Poggio. Hierarchical models of object recognition in cortex. *Nature Neuroscience*, 2(11):1019–1025, 1999. 18

[138] V. Rodehorst, M. Heinrichs, and O. Hellwich. Evaluation of relative pose estimation methods for multi-camera setups. In *Congress of the International Society for Photogrammetry and Remote Sensing*, pages 135–140, 2008. 94, 97

[139] V. Rodehorst and O. Hellwich. Genetic algorithm sample consensus (GASAC) - a parallel strategy for robust parameter estimation. In *International Workshop 25 Years of RANSAC*, IEEE Computer Society, pages 1–8, New York, USA, June 2006. 102

[140] V.A. Rodehorst and A.B. Koschan. Comparison and evaluation of feature point detectors. In *Turkish-German Joint Geodetic Days*, 2006. 14, 15, 22, 63

[141] E. Rosten, R. Porter, and T. Drummond. Faster and better: A machine learning approach to corner detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32:105–119, 2010. 27, 28

[142] P.J. Rousseau and A.M. Leroy. *Robust regression and outlier detection*. John Wiley & Sons, New York, NY, USA, 1987. 97, 102

[143] Y. Rubner, C. Tomasi, and L.J. Guibas. The earth mover"s distance as a metric for image retrieval. Technical report, Stanford University, 1998. 49, 50, 52

[144] T. Sattler, B. Leibe, and L. Kobbelt. SCRAMSAC: Improving RANSAC's efficiency with a spatial consistency filter. In *International Conference on Computer Vision*, pages 2090–2097. IEEE Computer Society, 2009. 98, 99, 101, 102

[145] S. Savarese, J. Winn, and A. Criminisi. Discriminative object class models of appearance and shape by correlatons. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2033–2040. IEEE Computer Society, 2006. 18

[146] F. Schaffalitzky and A. Zisserman. Multi-view matching for unordered image sets, or "how do i organize my holiday snaps?". In *European Conference on Computer Vision*, pages 414–431. Springer-Verlag, 2002. 47

[147] D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International Journal of Computer Vision*, 47(1-3):7–42, 2002. 85

[148] C. Schmid and R. Mohr. Local grayvalue invariants for image retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19:530–535, 1997. 11

[149] C. Schmid, R. Mohr, and C. Bauckhage. Comparing and evaluating interest points. In *International Conference on Computer Vision*. IEEE Computer Society Press, January 1998. 5, 55

[150] L.G. Shapiro and G.C. Stockman. *Computer Vision*. Prentice Hall, 2001. 16, 28, 48

[151] E. Shechtman and M. Irani. Matching local self-similarities across images and videos. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE Computer Society, 2007. 47, 48

[152] J. Shi and C. Tomasi. Good features to track. Technical report, Cornell University, Ithaca, NY, USA, 1993. 12, 21

[153] J. Sivic and A. Zisserman. Video google: a text retrieval approach to object matching in videos. In *IEEE International Conference on Computer Vision*, volume 2, pages 1470–1477, 2008. 15, 16

[154] S.M. Smith and J.M. Brady. Susan-a new approach to low level image processing. *International Journal of Computer Vision*, pages 45–78, 1997. 27

[155] N. Snavely, R. Garg, Steven M. Seitz, and R. Szeliski. Finding paths through the world's photos. *ACM Transactions on Graphics*, 27(3):11–21, 2008. 85

[156] N. Snavely, S.M. Seitz, and R. Szeliski. Modeling the world from internet photo collections. *International Journal of Computer Vision*, 80(2):189–210, 2008. 85, 86

[157] C. Spearman. The proof and measurement of association between two things. *The American journal of psychology*, 100(3-4):441–471, 1987. 54

[158] C.V. Stewart. Minpran: A new robust estimator for computer vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(10):925–938, 1995. 102

[159] R. Subbarao and P. Meer. Beyond RANSAC: User independent robust regression. In *Conference on Computer Vision and Pattern Recognition Workshop*, page 101, Washington, DC, USA, 2006. IEEE Computer Society. 102

[160] M.J. Swain and D.H. Ballard. Color indexing. *International Journal of Computer Vision*, pages 11–32, 1991. 50

[161] R. Szeliski. *Computer Vision: Algorithms and Applications*. Springer, 2010. 6, 8, 22, 85, 89, 91, 92

[162] B.M. ter Haar Romeny, L. Florack, A.H. Salden, and M.A. Viergever. Higher order differential structure of images. In *International Conference on Information Processing in Medical Imaging*, pages 77–93. Springer-Verlag, 1993. 8, 47

[163] M. Toews and W. Wells. SIFT-rank: Ordinal description for invariant feature correspondence. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 172–177. IEEE Computer Society, 2009. 5, 54, 64

[164] E. Tola, V. Lepetit, and P. Fua. Daisy: an efficient dense descriptor applied to wide baseline stereo. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(5):815–830, 2010. 5, 44, 45

[165] C. Tomasi and T. Kanade. Detection and tracking of point features. Technical report, Carnegie Mellon University, 1991. 21

[166] C. Tomasi and T. Kanade. Shape and motion from image streams under orthography: a factorization method. *International Journal of Computer Vision*, 9(2):137–154–154, 1992. 85

[167] F. Tombari, S. Mattoccia, L. di Stefano, and E. Addimanda. Classification and evaluation of cost aggregation methods for stereo correspondence. In *Computer Vision and Pattern Recognition*, 2008. 85

[168] P.H.S. Torr. An assessment of information criteria for motion model selection. In *Computer Vision and Pattern Recognition*, pages 47–52, 1997. 100

[169] P.H.S. Torr and A. Zisserman. Robust computation and parametrization of multiple view relations. In *International Conference on Computer Vision*, page 727. IEEE Computer Society, 1998. 5, 97

[170] P.H.S. Torr and A. Zisserman. MLESAC: a new robust estimator with application to estimating image geometry. *Computer Vision and Image Understanding*, 78(1):138–156, 2000. 5, 97

[171] P.H.S. Torr, A. Zisserman, and S.J. Maybank. Robust detection of degenerate configurations for the fundamental matrix. In *International Conference on Computer VisionV*, 1995. 100

[172] M. Trajkovic and M. Hedley. Fast corner detection. *Image and Vision Computing*, 16(2):75–87, 1998. 27, 28

[173] B. Triggs, P. McLauchlan, R.I. Hartley, and A.W. Fitzgibbon. Bundle adjustment - a modern synthesis. In *International Workshop on Vision Algorithms*, pages 298–372, London, UK, 2000. Springer-Verlag. 85

[174] E. Trucco and A. Verri. *Introductory Techniques for 3-D Computer Vision*. Prentice Hall, 1998. 13, 26, 85, 86, 87, 89, 91, 92

[175] L. Trujillo and G. Olague. Using evolution to learn how to perform interest point detection. In *International Conference on Pattern Recognition*, pages 211–214. IEEE Computer Society, 2006. 33

[176] T. Tuytelaars and L. Van Gool. Matching widely separated views based on affine invariant regions. *International Journal of Computer Vision*, 59(1):61–85, 2004. 30, 34, 35, 36

[177] T. Tuytelaars and K. Mikolajczyk. Local invariant feature detectors: a survey. *Foundations and Trends in Computer Graphics and Vision*, 3(3):177–280, 2008. 3, 8, 19, 25, 28, 29, 31, 34, 35, 36, 63

[178] F. Ullah and S. Kaneko. Using orientation codes for rotation-invariant template matching. *Pattern Recognition*, 37(2):201–209, 2004. 79

[179] A. Vedaldi. *Invariant Representations and Learning for Computer Vision*. PhD thesis, University of California at Los Angeles, 2008. 51, 55

[180] A. Vedaldi and S. Soatto. A complexity-distrotion approach to joint pattern alignment. In B. Sch olkopf, J.C. Platt, and T. Hofmann, editors, *Advances in Neural Information Processing Systems*, Cambridge, MA, 2007. MIT Press. 52

[181] A. Vedaldi and S. Soatto. Relaxed matching kernels for object recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2008. 17

[182] P. Viola and M. Jones. Robust real-time object detection. In *International Workshop on Statistical and Computational Theories of Vision: Modelling, Learning, Sampling and Computing*, 2001. 31

[183] A. Wald. *Sequential Analysis (Dover Phoenix Editions)*. Dover Publications, 2004. 101

[184] C. F. R. Weiman and G. Chaikin. Logarithmic spiral grids for image processing and display. *Computer Graphics and Image Processing*, 11:197–226, 1979. 42

[185] S. Winder, G. Hua, and M. Brown. Picking the best daisy. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 0, pages 178–185. IEEE Computer Society, 2009. 52, 64

[186] H.J. Wolfson and I. Rigoutsos. Geometric hashing: An overview. *Computing in Science and Engineering*, 4:10–21, 1997. 16

[187] W. Yan, Q. Wang, Q. Liu, and H. Lu. Topology-preserved diffusion distance for histogram comparison. In *British Machine Vision Conference*, 2007. 53, 54, 64

[188] R. Zabih and J. Woodfill. Non-parametric local transforms for computing visual correspondence. In *European Conference on Computer Vision*, volume 2, pages 151–158. Springer-Verlag, 1994. 40

[189] W. Zhang and J. Kosecka. Generalized RANSAC framework for relaxed correspondence problems. In *International Symposium on 3D Data Processing, Visualization, and Transmission*, pages 854–860, Washington, DC, USA, 2006. IEEE Computer Society. 96

[190] Z. Zhang, R. Deriche, O. Faugeras, and Q. Luong. A robust technique for matching two uncalibrated images through the recovery of the unknown epipolar geometry. *Artificial Intelligence*, 78(1-2):87–119, 1995. 5, 102

[191] Z. Zhang and Y. Shan. A progressive scheme for stereo matching. In *European Workshop on 3D Structure from Multiple Images of Large-Scale Environments*, pages 68–85, London, UK, 2001. Springer-Verlag. 85