University College London

Mullard Space Science Laboratory

Submitted in accordance with the requirements for the degree of Doctor of Philosophy in Space and Climate Physics at UCL.

Fusion of LIDAR with stereo camera data - an assessment

Doctoral thesis of:

Joshua Veitch-Michaelis

Primary Supervisor:

Prof. Jan-Peter Muller, UCL MSSL

Secondary Supervisor:

Dr. Jonathan Storey, IS-Instruments Ltd

Tertiary Supervisor:

Dr. David Walton, UCL MSSL

2016

Declaration

I, Joshua Veitch-Michaelis confirm that the work presented in this thesis is my own. Where information has been derived from other sources, I confirm that this has been indicated in the thesis.

Acknowledgements

I would like to start by thanking everyone both at UCL and IS-Instruments who both made this project possible over the last four years. I would like to extend an enormous amount of gratitude to my supervisors at UCL - Prof. Jan-Peter Muller and Dr David Walton for their continual guidance throughout the entire PhD process. At IS Instruments, I would like to thank Dr Jon Storey, Dr Ben Crutchley, Dr Mike Foster and Nick Bantin for mentoring, problem solving and for providing many opportunities which I would otherwise have not had as a PhD student. I would like to thank Ben for assisting with data collection and retaining my sanity throughout countless camera calibrations. I would like to acknowledge STFC and UCL for providing the necessary funding to complete this research. I am fortunate to have been a part of such a friendly and welcoming department (MSSL), the lab has been my second home for the past four years. Thank you to all the other PhD students at MSSL and especially in the imaging group.

I would like to say a huge thankyou to my family for their continued support and for encouraging me to finish! This PhD would not exist, were it not for both my grandmothers who irresponsibly fostered my love of useless gadgetry for most of my formative years, much to the horror of my parents. Special thanks to my good friend (soon to be Dr) Maryam Ahmed who has been a source of sage advice, scholarly wisdom and crass humour. Long may we remain friends.

Finally, the International Astronomical Youth Camp (IAYC) has been an integral part of my life. I would like to acknowledge both my good friends at the International Workshop for Astronomy (IWA) and indeed anyone who has attended the IAYC in the last 8 years. This is a long, and incomplete list: Aitor, Aga, Alex, Anci, Balazs, Dan, Eli, Hannah, Irati, James, Kieran, Klaus, Lina, Mac, Martin, Ondrej. It's no exaggeration to say that we've been through the best of times and the worst of times together.

Joshua Veitch-Michaelis

2016

Abstract

This thesis explores data fusion of LIDAR (laser range-finding) with stereo matching, with a particular emphasis on close-range industrial 3D imaging. Recently there has been interest in improving the robustness of stereo matching using data fusion with active range data. These range data have typically been acquired using time of flight cameras (ToFCs), however ToFCs offer poor spatial resolution and are noisy. Comparatively little work has been performed using LIDAR. It is argued that stereo and LIDAR are complementary and there are numerous advantages to integrating LIDAR into stereo systems. For instance, camera calibration is a necessary pre-requisite for stereo 3D reconstruction, but the process is often tedious and requires precise calibration targets. It is shown that a visible-beam LIDAR enables automatic, accurate (sub-pixel) extrinsic and intrinsic camera calibration without any explicit targets.

Two methods for using LIDAR to assist dense disparity maps from featureless scenes were investigated. The first involved using a LIDAR to provide high-confidence seed points for a region growing stereo matching algorithm. It is shown that these seed points allow dense matching in scenes which fail to match using stereo alone. Secondly, LIDAR was used to provide artificial texture in featureless image regions. Texture was generated by combining real or simulated images of every point the laser hits to form a pseudo-random pattern. Machine learning was used to determine the image regions that are most likely to be stereo-matched, reducing the number of LIDAR points required. Results are compared to competing techniques such as laser speckle, data projection and diffractive optical elements.

Contents

List of Figures

List of Tables

List of Acroynms and Abbreviations

n**D** - n-dimensional

ALSC - Adaptive Least Squares Correlation

AMCW - Amplitude-Modulated Continuous-Wave

ASIC - Application-Specific Integrated Circuit

CAD - Computer Aided Design

CCD - Charge Coupled Device

CMOS - Complementary Metal Oxide Semiconductor

COTS - Commercial Off-The-Shelf

CPI - Centre for Process Innovation

CPU - Central Processing Unit

CUDA - Compute Unified Display Architecture

CoC - Centre of Confusion

DARPA - Defense Advanced Research Projects Assocation

DLT - Direct Linear Transform

DNSP - Distance to Nearest SIFT Point

DP - Dynamic Programming

DoF - Depth of Field

FMCW - Frequency-Modulated Continuous-Wave

FOV - Field of View

FPGA - Field Programmable Gate Array

GBP - Great Britain Pound

GLCM - Grey Level Correlation Matrix

GPS - Global Positioning System

GPU - Graphics Processing Unit

GUI - Graphical User Interface

Gotcha - Gruen-Otto-Chau ALSC stereo matcher

HTP-C - High Temperature Process Control

IMU - Inertial Measurement Unit

IR - Infrared

JDL - Joint Directors of Laboratories

LED - Light Emitting Diode

LIDAR/LADAR - LIght Detection and Ranging

LM- Levenberg Marquadt (minimisation)

LUT - LookUp Table

MDF - Medium Density Fibreboard.

MP - Megapixel

MPI - Materials Processing Institute

NCC - Normalised Cross-Correlation

NIR - Near Infrared

NN - Neural Network

OEM - Original Equipment Manufacturer

OpenCL - Open Compute Langauge

PNG - Portable Network Graphics

PNP - Point-n-Perspective algorithm

PRF - Pulse Repetition Frequency

QE - Quantum Efficiency

RADAR - RAdio Detection and Ranging

RAM - Random Access Memory

RANSAC - RANdom SAmple Consensus

RF - Random Forest

RLRD - Real LIDAR Random Dot

RMS - Root Mean Square

SAD - Sum of Absolute Differences

SDK - Software Development Kit

SGM - Semi-Global Matching

SIFT - Scale Invariant Feature Transform

SLAM - Simultaneous Location and Mapping

SLRD - Simulated LIDAR Random Dot

SNR - Signal to Noise Ratio

SPAD - Single Photon Avalanche Photodiode

SSD - Solid State Disk

STFC (CASE) - Science Technologies Facilities Council ()

SVD - Singular Value Decomposition

SVM - Support Vector Machine

SWIR - Short Wave Infrared

ToF - Time of Flight

ToFC - Time of Flight Camera

USD - United States Dollar

VGA (resolution) - 640x480 px

WTA - Winner-Take-All

XML - eXtensible Markup Language

List of Symbols

SI units and prefixes are used throughout this thesis.

Bold capital letters are used to denote matrices, $\mathbf{X}$. In general rotation matrices are are denoted with $\mathbf{R}$.

Bold lower case denotes a vector, e.g. $\mathbf{x}$. Translation vectors are denoted $\mathbf{t}$.

Regular capital letters, e.g. X, denote metric world coordinates.

A subscript i is used to imply the presence of a set of points, e.g. a point cloud might be denoted X_i.

Lower case letters, e.g. x, denote local coordinate such as in a camera frame.

Pixel coordinates are denoted (u, v)

Unless explicitly stated in the text, the following definitions can be assumed:

b - inter-camera baseline

B - noise bandwidth of a photodiode

c - speed of light

(c_x, c_y) - camera centre (intersection of optical axis with sensor plane).

C - matching cost

d - disparity

f - camera focal length, also frequency of modulated LIDAR

$\mathbf{F}$ - camera intrinsic matrix

$\mathbf{h}$ - image height

$I(x, y)$ - an image

k_i - radial distortion coefficients

$\mathbf{K}$ - camera fundamental matrix

p_i - tangential distortion coefficients

$\mathbf{P}$ - a 3x4 projection matrix

R - range to target

$\mathfrak{R}_O$ - unity gain responsivity of a photodiode

s - skew

t - time

$\mathbf{w}$ - image width

ϵ - accuracy, e.g. range accuracy ϵ_R.

λ - wavelength

ϕ - phase

ρ - target reflectivity

φ - altitude angle

θ - azimuth angle

Introduction 1

1.1 Why do we need 3D imaging at close range?

There is an increasing need for 3D reconstruction of close range targets. Estimates of the market size of 3D imaging are as much as 17 bn USD by 2021[1].

Robotic vision systems typically require a knowledge of depth in order to be able to navigate or to manipulate objects in the world accurately (Besl, 1988; Hebert, 2000; Blais, 2004). Vehicles such as those used in the DARPA challenges (Thrun, 2006), the Google car[2] or the Oxford Robotics RobotCar[3] use a suite of sensors for both navigation and obstacle avoidance (Figure 1.1). These systems must be able to produce dense, accurate 3D information in realtime over a large depth of field. Navigation is often performed using a class of algorithm called Simultaneous Location and Mapping (SLAM), which relates 3D information over time to both navigate and localise in an environment (Newman et al., 2009; Geiger et al., 2013).

In the consumer sector, 3D imaging has been recently used for pose estimation and gesture detection (Kollorz et al., 2008). Several companies, including Intel[4] and Google[5], have developed handheld 3D imaging solutions capable of real-time performance. The Microsoft Xbox Kinect platform, a 3D imaging system designed for entertainment purposes, has sold millions of units and has become an attractive low-cost research platform (Khoshelham and Elberink, 2012; Han et al., 2013).

Imaging, both 2D and 3D, is already used for a variety of industrial tasks such as process inspection, quality assurance and defect detection (Newman and Jain, 1995; Malamas et al., 2003).

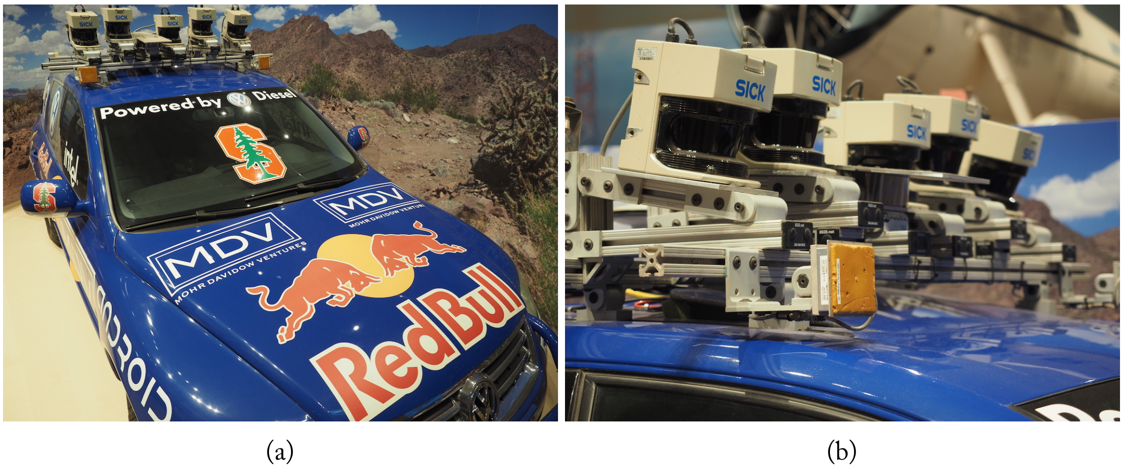
[1] `http://www.transparencymarketresearch.com/3d-imaging-market.html`, accessed 20/9/2016

[2] `https://www.google.com/selfdrivingcar/`, accessed 20/9/2016

[3] `http://robotcar.org.uk`, accessed 20/9/2016

[4] `http://www.intel.co.uk/content/www/uk/en/architecture-and-technology/realsense-overview.html`, accessed 20/9/2016

[5] `https://www.google.com/atap/project-tango/`, accessed 20/9/2016



(a)

(b)

Figure 1.1: (a) The autonomous car, ”Stanley” which won the 2005 DARPA Grand Challenge, covering 132 miles over desert terrain in around seven hours. The car is equipped with an array of 2D line-scan LIDAR units (b), stereo and monocular vision systems, amongst other sensors. The car is currently on exhibition at the Smithsonian National Air and Space Museum, Washington DC. Images author's own work.

The results from the imaging system may be analysed or compared to a reference to provide repeatable conformity assessments. The requirements for these kind of inspection systems is often very high accuracy (sub-mm) and real-time or near real-time measurements. Depending on the application, a number of technologies are used including stereo vision, LIDAR and laser triangulation. In many environments, such as the steel industry, inspection is frequently performed by hand in a qualitative fashion, requiring skilled operators (Landstrom and Thurley, 2012).

1.2 Terminology and scope

Due to the overlap between computer vision and photogrammetry, some terminology used in this thesis is outlined here to avoid confusion.

'Stereo' is widely used to describe the paradigm of stereo vision, encompassing a binocular vision system and an associated image matching algorithm. Similarly, 'LIDAR‘ is widely used to describe various different types of laser ranging system. In this thesis, LADAR refers explicitly to laser-based area time of flight sensors such as flash LIDAR. LIDAR is used to refer to systems that perform 1D measurements and therefore require scanning of some sort to image a scene.

For convenience, this thesis defines the following terrestrial ranges: very close range as (0-1m), close range as (1-20m), medium range as (20-100m), long range as (100-1km) and very long range as (>1km). Industrial process monitoring systems tend to fall into the categories of close to medium range. Long and very long range systems are typical in surveying and large asset management, such as mining stockpiles.

The output of a 3D imaging system is typically either a '2.5D' image where intensity in the image is proportional to range or a list of 3D points, a point cloud. In order to convert a range image to a metric point cloud, some kind of system calibration is required. In the case of stereo this involves knowledge of the lens characteristics and the inter-camera separation (baseline).

Point clouds are considered to be raw 3D information. Further processing might include conversion to a solid 3D model (meshing), object segmentation or some other kind of scene interpretation. In this thesis, the focus is on the production of the point cloud rather than any downstream analysis.

1.3 Goals for a 3D imaging system

A non-exhaustive list of the goals of an ideal imaging system are as follows:

- Measurements should have suitable**resolution**, referring to the physical separation between individual measurements.
- Measurements should be **accurate**, **repeatable** and **precise** (see Figure 1.2. Accuracy describes the difference between a measured quantity and its known value, i.e. compared to ground truth).
- Generating a 3D model of a scene should be fast, this is dependent on both **acquisition time** and **processing time**.
- The measurement system should be **robust**, coping with varying illumination conditions, indoor and outdoor operation, and varying surface reflectances.
- The system should be **low cost** and **compact**.

These performance metrics are necessarily ambiguous and what is deemed 'sufficient' for a particular task is context-specific. A vision system for an autonomous vehicle requires 3D models to be generated and interpreted in real-time. On the other hand, capturing a model of a static building can be performed more slowly and the analysis performed offline. Vision systems for

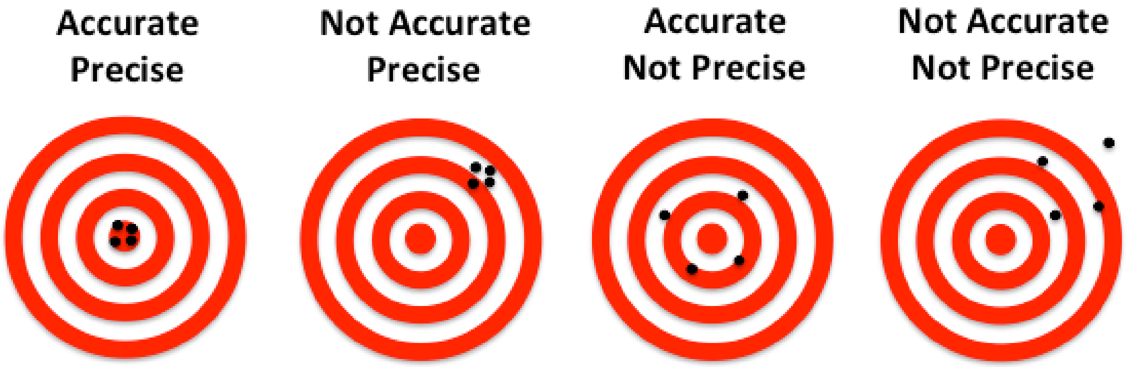


Figure 1.2: A popular "bullseye" representation of the difference between accuracy and precision.

industrial process lines are often able to exploit controlled illumination, whereas this is not possible for a system that must work outdoors.

The cost and size of measurement systems has tended to decrease over time. The past two decades have seen tremendous advances in computer systems. Portable electronic devices like smartphones and tablet computers have become ubiquitous and most now contain multi-core processors, discrete graphics processing units (GPUs), high resolution cameras and generous amounts of onboard storage (gigabytes).

Realtime stereo matching on megapixel imagery has become possible with algorithms running on high end consumer CPUs and GPUs (see Section 3.2.8). State-of-the-art LIDAR systems are capable of measuring at 1 Mpts/sec, but this is typically over a full hemisphere and precludes imaging dynamic scenes. Aside from speed, the accuracy of LIDAR and stereo systems is mostly limited by geometric constraints, discussed in Chapter 2.

Stereo matching provides high resolution 3D data, but performs poorly if the input images have poor local intensity variation (texture). Additionally stereo accuracy is strongly dependent on the ratio of camera baseline to distance with poor results at long range[6] if this baseline is fixed. LIDAR provides high resolution measurements dependent largely on the available signal-to-noise ratio (SNR), but systems are still relatively expensive and high resolution results take time to acquire. Time of flight cameras produce realtime results up to around 10m, but spatial resolution is currently poor and the returned range data is noisy.

Data fusion offers a solution to address the shortcomings of individual systems. Recently there have been a number of attempts to fuse stereo matching with additional sources of range data such

[6] Accuracy reduces proportionally to the square of the distance from the camera.

as LIDAR and time of flight cameras (ToFCs). These results are promising and show that the shortcomings of stereo (such as performance in low texture regions) can be addressed, producing point clouds which are superior to either technology alone. Although much work has been done investigating data fusion of stereo with ToFCs, comparatively little has been done using LIDAR.

The primary aim of this thesis is to investigate ways that LIDAR can be integrated into stereo imaging systems in order to enable dense 3D reconstruction in scenes which challenges stereo matching algorithms.

1.4 Contributions

This thesis presents four main contributions to knowledge in the field:

1. An automatic intrinsic and extrinsic camera calibration algorithm using a visible-beam scanning LIDAR.
2. Data fusion of LIDAR with a region-growing stereo matching algorithm. (Veitch-Michaelis et al., 2015)
3. Texture projection generated using a scanning LIDAR (Veitch-Michaelis et al., 2016)
4. Machine learning to define areas of poor texture which require additional seedpoints from LIDAR

(1) Camera calibration is necessary for metric 3D reconstruction using stereo imagery. Current calibration techniques are very accurate, but the process of acquiring calibration imagery is often tedious and requires explicit calibration targets. Some experience with the process is also helpful to ensure that the calibration images are of a high enough quality. By imaging the laser spot of a visible-beam scanning LIDAR, it is possible to generate highly accurate calibration points without an explicit target. LIDAR-derived intrinsic and extrinsic calibration is shown to be comparable in accuracy (sub-pixel) to standard camera calibration routines and may be performed automatically.

(2) Region growing stereo involves taking a set of tentative 'seed' correspondences between the two views before iteratively growing the disparity map around these points. This approach to stereo matching has proved to be highly accurate and is routinely used for terrestrial, planetary and rover imagery. However, this approach can fail in image regions with poor texture or in regions

where there are few seed points. The proposed method uses a LIDAR to provide additional high-confidence seed points.

(3) Texture projection improves stereo match performance in image regions with uniform intensity. By imaging the laser spot of a LIDAR as it scans, it is possible to generate texture in a scene by combining images of every point the laser hits. An alternative method is proposed where the LIDAR spot images are simulated, avoiding issues of scene illumination. Results are compared to competing techniques such as laser speckle, data projection and diffractive optical elements. (4) Machine learning was used to determine which regions of the image were unlikely to be matched using the LIDAR and should be augmented by artificial texture projection.

1.5 Thesis Outline

Chapter 2 discusses several forms of 3D imaging: stereo, structured light, LIDAR, laser triangulation and time of flight cameras. This chapter deals largely with the limitations of each system from a hardware or geometric perspective. The advantages and disadvantages of each system are described along with theoretical system performance.

Chapter 3 discusses stereo matching algorithms. The different classes of stereo matcher are reviewed. Stereo benchmarking is briefly discussed, as ground truth imagery is necessary to compare different algorithms. The remainder of the chapter reviews previous attempts at data fusion of stereo imagery with range data from time of flight cameras and LIDAR.

Chapter 4 introduces the hardware used for the experiments in this thesis, a stereo system and combined scanning LIDAR. A geometric model for the LIDAR is suggested, allowing compensation for misalignment of the LIDAR unit on its scanning platform. A method for robustly locating the LIDAR spot in a camera image is given and the theoretical performance of each system is discussed.

Chapter 5 discusses camera and LIDAR cross-calibration. First, previous LIDAR-camera calibration methods are reviewed. Then, the intrinsic calibration of the LIDAR using the model in Chapter 4 is given. The cross-calibration of a stereo camera system and a scanning LIDAR is introduced. Calibration procedures are given for the various levels of calibration required, from the case when the cameras are already calibrated to a full intrinsic and extrinsic calibration. Results are favourably compared to a standard 'chessboard' calibration method.

In Chapter 6, the Gruen-Otto-Chau Adaptive-Least-Squares-Correlation (Gotcha) stereo matcher Gruen (1985); Otto and Chau (1989); Shin and Muller (2012) is formally described. By considering heuristics such as image entropy and the distribution of seed points, machine learning is used to predict how well a particular image is likely to be matched. Using this information, a routine for generating additional seed points using the LIDAR is described. Results are presented from several challenging indoor scenes.

Chapter 7 focuses on 'active' stereo matching, where additional illumination is used to improve the texture of a scene. This chapter presents a method for generating texture based on a LIDAR scan, via both direct LIDAR spot imaging and simulation. A machine learning algorithm for predicting which parts of an image are likely to be unmatched is developed and used to efficiently direct the LIDAR during the scan.

Chapter 8 concludes the thesis, summarises the unique contributions of this research and gives some recommendations for further work.

Theory and Context 2

2.1 Overview

This chapter reviews the theory behind the 3D imaging techniques relevant to this research, primarily stereo and LIDAR. Time of flight cameras are also described, due to their relevance to data fusion. This chapter primarily deals with the hardware-dependent performance of each type of system. The performance of stereo systems is also heavily dependent on the software component, i.e. the image matching algorithm. This is discussed in greater detail in Chapter 3.

The geometry of each type of system is described as this places fundamental limits on the accuracy and resolution of the 3D measurements. Other important characteristics for comparison include depth of field and minimum/maximum sensing range. The strengths and weakness of the types of imaging system are compared and commercial examples are given, where possible.

2.2 Metrics for assessing 3D measurement systems

3D imaging systems have been compared quantitatively in a wide variety of ways including acquisition speed, processing time, depth of field, field of view, range accuracy and range resolution (Besl, 1988; Hebert, 2000; Amann et al., 2001; Blais, 2004; Berkovic and Shafir, 2012). Optical measurement systems are active or passive. Active systems measure range by transmitting light onto a target and performing some analysis on the reflected signal. Passive sensors rely on light emitted by the target (e.g. thermal infra-red) or external illumination that has scattered off the target. General inter-comparison between technologies requires spatial context; a system with active illumination may outperform a LIDAR indoors, but might be unable to measure distances greater than several metres.

Qualitative metrics are also important, if harder to define. These include illumination constraints, minimum object reflectivity (an issue for both active and passive systems) and ease of calibration. While system cost is an important consideration, it is dependent on a wide range of external factors and it is reasonable to expect systems to become cheaper over time.

Besl (1988) reviewed a variety of active imaging sensors and used the following figure of merit for comparison:

$$M = \frac{L_r}{\epsilon_R \sqrt{T}} \tag{2.1}$$

where L_r is the depth of field, ϵ_R the range accuracy and T the time per measurement such that an image with N total pixels takes time NT to read out. The merit figure is biased towards high-accuracy rather than high-speed systems. Besl noted that there had been order of magnitude improvements in acquisition time in the previous decade, though the main bottleneck was data processing time. Blais (2004) noted that this merit figure became quickly outdated by improvements in hardware.

2.3 Stereo Imaging

Distance measurement through triangulation has been used in mapping for millennia. Human binocular vision was investigated philosophically in the mid 1800s by Wheatstone (1838) who also detailed plans for the stereoscope, a device used for viewing pairs of images in apparent 3D.

The first instruments for performing correlation on digital imagery were proposed in the 1960s[1]. Since then, with the development of Charge Coupled Device (CCD) (Boyle and Smith, 2013) and Complementary Metal Oxide Semiconductor (CMOS) imaging sensors (Fossum, 1995), stereo image matching became a focus of photogrammetry and computer vision research (see Chapter 3). Stereo is routinely used to perform terrain reconstruction from aerial and orbital imagery (Toutin, 2004) and remains a powerful photogrammetric tool (Gruen, 2012). From the perspective of computer vision research, stereo is attractive for producing dense measurements suitable for simultaneous location and mapping (SLAM) and autonomous navigation (Geiger et al., 2013).

Modern image sensors are cost effective, high resolution and are available with sensitivities in a variety of wavebands. The process of deriving 3D information from two images is that of

[1] See G. Hobrough's patents US 2964642 (1960) and US 3145303 (1964)

triangulation. A point in the scene is imaged by two (or more) cameras with an overlapping field of view. If the camera is suitably calibrated and the pixel coordinates of the point is known in both images, trivial geometry gives the position of the point in the scene. However, identifying matching pixels between images is not trivial; this is known as the correspondence problem and is solved with an algorithm called a stereo matcher. As such, stereo performance is dependent on both the system geometry and the image matching algorithm. The next section discusses camera geometry, stereo matching algorithms are discussed in Chapter 3 and calibration is discussed in Chapter 5.

2.3.1 Camera geometry

In computer vision, cameras are typically represented using a pinhole camera model with additional corrections for radial and tangential lens distortion (Hartley and Zisserman, 2003). The pinhole model describes an ideal camera with focal length f and principal point (c_x, c_y). Figure 2.1 shows an example of a stereo system, modelled using the pinhole geometry.

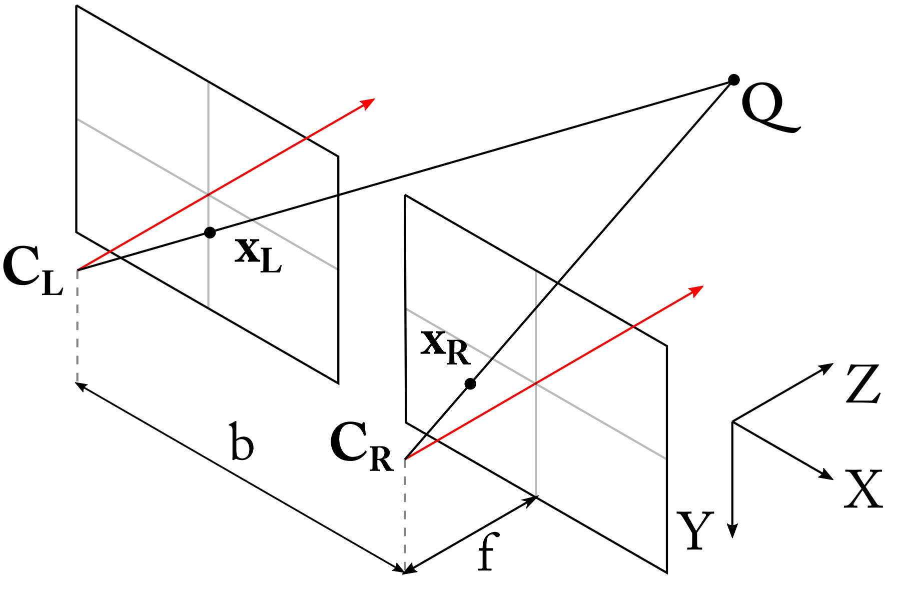


FIGURE 2.1: Pinhole geometry for a pair of identical cameras. x_L and x_R are the image of the 3D point Q. Physically the projection centres C_L and C_R are in front of the sensor, however as digital images are normally flipped horizontally and vertically after readout this arrangement is equivalent.

The focal length is the distance from the pinhole to the image (sensor) plane and the principal point is the intersection of the optical axis with the image plane. Additional parameters include the skew, s which introduces a shearing effect in the image and is non-zero for non-square pixels, normally $s = 0$.

A projective geometry maps points X in the world to points x in the image. By assuming that all rays from the world pass through a common centre of projection, nominally the centre of the lens, this mapping may be expressed in homogeneous coordinates as a 3x4 projection matrix **P**: $x = \mathbf{P}_{3\text{x}4}X$ where $x = (x, y, 1)^T$ and $X = (X, Y, Z, 1)^T$. The structure of **P** is then:

$$\mathbf{P} = \begin{bmatrix} f & s & c_x & 0 \\ 0 & f & c_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} = \mathbf{K}[\mathbf{I}|\mathbf{0}] \tag{2.2}$$

K is the intrinsic calibration matrix. Allowing for rotation and translation of the camera with respect to the world origin:

$$\mathbf{P} = \begin{bmatrix} f & 0 & c_x \\ 0 & f & c_y \\ 0 & 0 & 1 \end{bmatrix} [\mathbf{R}|\mathbf{t}] = \mathbf{K}\,[\mathbf{R}|\mathbf{t}] \tag{2.3}$$

with **R** a 3x3 rotation matrix and **t** a 3x1 translation vector. Introducing (u, v) as pixel coordinates on the sensor:

$$x' = \frac{fX}{Z} \tag{2.4}$$

$$y' = \frac{fY}{Z} \tag{2.5}$$

$$(u, v) = (x' - c_x, y' - c_y) \tag{2.6}$$

where the focal length is in units of pixels. Real lenses introduce distortion which may be modelled as a combination of radial and tangential components k and p respectively:

$$x'' = x' \frac{1 + k_1 r^2 + k_2 r^4 + k_3 r^6}{1 + k_4 r^2 + k_5 r^4 + k_6 r^6} + 2p_1 x' y' + p_2(r^2 + 2x'2) \tag{2.7}$$

$$y'' = y' \frac{1 + k_1 r^2 + k_2 r^4 + k_3 r^6}{1 + k_4 r^2 + k_5 r^4 + k_6 r^6} + 2p_2 x' y' + p_1(r^2 + 2y'2) \tag{2.8}$$

$$r^2 = x'^2 + y'^2 \tag{2.9}$$

$$(u, v) = (x'' - c_x, y'' - c_y) \tag{2.10}$$

Higher order terms are omitted above; the OpenCV (Bradski, 2000) camera calibration algorithm (`calibrateCamera`) by default fits k_1, k_2, k_3, p_1 and p_2 and both Tsai (1987) and Zhang (2000) ignore tangential distortion. Thus there are 3 degrees of freedom in the camera matrix ($s = 0$), 3 rotational degrees of freedom and 3 translational degrees of freedom. Introducing distortion adds another 5 degrees of freedom for a total of 14 parameters to specify a camera.

2.3.2 Depth of Field

When a lens is adjusted such that light rays originating from a point at distance D converge on the sensor plane, the lens is said to be focused at D as shown in Figure 2.2.The acceptable CoC, C, defines near and far focus limits D_N and D_F respectively. Objects in the range $[D_N, D_F]$ will be sharply imaged.

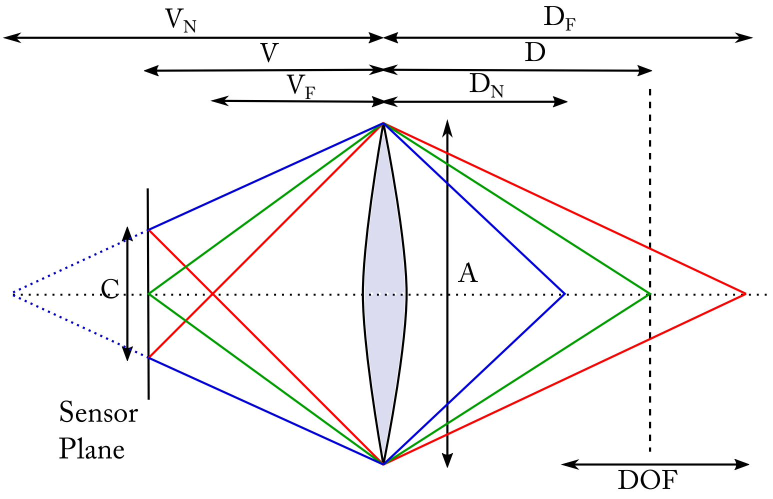


FIGURE 2.2: The depth of field of a lens is determined by the desired circle of confusion C which is related to the near and far distances D_N and D_F respectively.

Any object points not at this distance will appear blurred or defocused in the image (Potmesil and Chakravarty, 1982). The radius of the (blurred) image of a point source is called the circle of confusion (CoC). In practice some degree of blurring is unavoidable, but the choice of an acceptable CoC is subjective as it defines the difference between focused and unfocused parts of an image. A typical choice is the resolution of the sensor, i.e. the physical size of a single pixel (Biemond et al., 1990).

C is linearly dependent on the size of the lens aperture. The f-number of a lens, N is given as the ratio of the focal length f to the diameter of the lens, A:

$$N = \frac{f}{A} \tag{2.11}$$

It can be shown that (Krotkov, 1988):

$$D_F = \frac{Df^2}{f^2 + NC(D-f)} \tag{2.12}$$

$$D_N = \frac{Df^2}{f^2 - NC(D-f)} \tag{2.13}$$

In the limit $D_F \to \infty$:

$$D = H = \frac{f^2}{NC} + f \tag{2.14}$$

defining H as the hyperfocal distance. In this case $D_F = H/2$ is called the critical distance.

2.3.3 Epipolar geometry and rectification

Suppose two cameras are viewing a scene. A point in the left image must lie somewhere on a line in the right image - an epipolar line. The image of the left camera in the right image (and vice versa) is called an epipolar point. For a calibrated stereo system with the left camera at the origin, the locations of the epipoles are:

$$\mathbf{e} = \mathbf{K}\mathbf{R}^T\mathbf{t} \qquad \mathbf{e}' = \mathbf{K}'\mathbf{t} \tag{2.15}$$

where **K**, **R** and **t** have the same definitions as in section 2.3.1. If a point X in the world is imaged as x in the left camera and x' in the right, then the following is true:

$$x'^T \mathbf{F} x = 0 \qquad \mathbf{F} = \mathbf{K}'^{-T} \mathbf{R} \mathbf{K}^T [\mathbf{e}]_x \tag{2.16}$$

The 3×3 matrix $\mathbf{F}$ that satisfied these conditions is called the fundamental matrix. Under a particular 2D image transformation, called a rectifying homography (Loop and Zhang, 1999), the two images can be warped such that epipolar lines are parallel to the rows in the images. It is convenient to warp the images such that lens distortion is removed and the left and right cameras have the same focal length.

The epipolar constraint, applied to rectified images, is exploited by most stereo matching algorithms. For rectified images, a feature in the left image must lie somewhere on the same row in the right image (Figure 2.3). As the majority of stereo matching algorithms require rectified images, for the remaining discussion it is assumed that images are rectified prior to stereo matching.

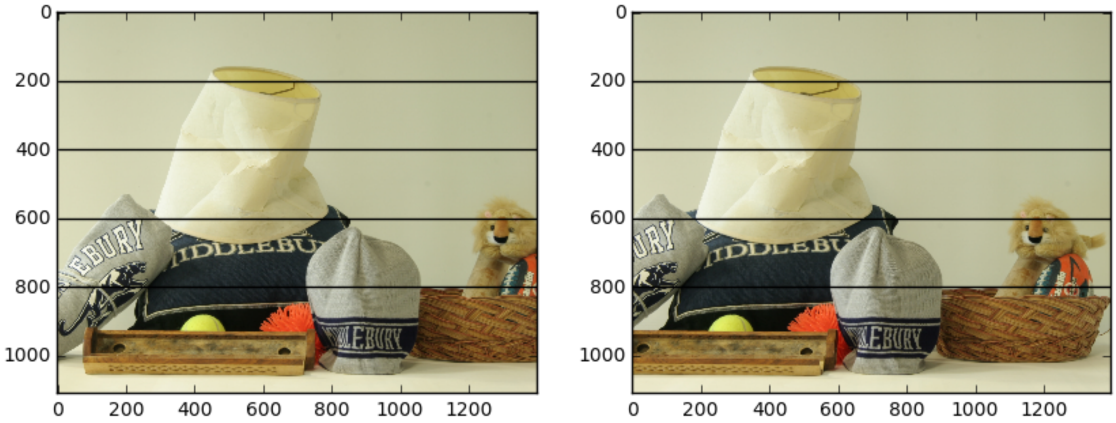


Figure 2.3: A stereo pair from the Middlebury Dataset with a number of epipolar lines shown. As the images have been epipolar rectified, the epipolar lines are horizontal.

2.3.4 Depth reconstruction

With a single calibrated camera, a point on the sensor defines a ray into the scene. At least one other view of the same point is necessary to remove depth ambiguity. For two views of a scene that are related by horizontal translation $\mathbf{T} = (b, 0, 0)^T$, as in the case of a rectified stereo pair, the image of a point $\mathbf{p} = (X, Y, Z)^T$ will be:

$$x_l = f\frac{X}{Z} \qquad x_r = f\frac{X+b}{Z} \tag{2.17}$$

Defining $d = x_r - x_l$ as the disparity:

$$Z = f\frac{b}{d} \tag{2.18}$$

The horizontal shift, b is the baseline between the two cameras and thus disparity is inversely related to depth. The output of a stereo matching algorithm is a disparity value that maps each pixel in the left image to the corresponding pixel in the right image. For a typical stereo rig, the disparity values will be negative.

2.3.5 Theoretical resolution

The range resolution, ΔZ, is determined by the uncertainty in the disparity measurement, ϵ_d:

$$\Delta Z = \frac{bf}{d} - \frac{bf}{d+\epsilon_d} = \frac{Z^2\epsilon_d}{bf + Z\epsilon_d} \approx \frac{Z^2}{bf}\epsilon_d \tag{2.19}$$

where the final result is obtained by Taylor expansion about $\epsilon_d = 0$. This uncertainty is also referred to as the correlation error since it is determined by the stereo matching algorithm. Stereo depth error degrades quadratically with distance from the camera baseline. Although increasing the focal length (at the expense of field of view) or decreasing the pixel size is a possible solution to lower the error, in practice increasing the camera baseline is simpler. (Gallup et al., 2008) describe a technique for variable baseline imaging to retain a constant depth accuracy throughout the scene, though it necessarily requires the baseline to be adjusted for each scene. The authors suggest that this technique would be applicable to video data where the camera is moving.

This depth uncertainty assumes that the disparity calculated for a particular pixel is correct (i.e. there is a good correspondence accuracy). The physical size of a pixel gives an upper bound to the error, providing the stereo matcher is at least pixel-accurate. State of the art stereo matchers return sub-pixel disparities, but measuring matching accuracy is not trivial. This issue is discussed further in Section 4.5.

2.3.6 Commercial stereo systems

There are relatively few commercially available stereo systems that explicitly output disparity information to the end-user. Table 2.1 shows some examples of currently available devices. The disparity speeds listed are assuming that the manufacturer's matching algorithm, if provided, is used.

Manufacturer	**System**	**Res. (Mpx)**	**Range (m)**	**Framerate (fps)**	Baseline (mm)
PointGrey	Bumblebee2	0.8	-	20	120
IDS	Ensenso N35	1.3	0-3	10	120
Stereolabs	ZED	5.5	0-15	15†	120
Intel	Realsense R200	0.17	3.5-10‡	60	70
Mobile Robots	MobileRanger	0.36	-	30	60
Videre Design	MEGA-DCS	1.3	-	-	80-240
e-Con Systems	Capella	0.36	-	-	100
LMI	Gocator 3100	1.3	0.25	5	-
Ricoh	SV-M-S1	1.3	0.8-1.2	30	200

Table 2.1: Commercially available stereo systems. Framerates refers to disparity computation speed. No maximum range is specified for the Bumblebee2, MobileRanger, Stinger or Capella. The MEGA-DCS and Capella are not provided with stereo matching software. † Up to 60 fps with 0.9 MP resolution. ‡ 10m extended range possible outdoors. The Gocator 3100 does not specify a baseline and specifies resolution .

The Pointgrey Bumblebee2 is provided with the Triclops Software Development Kit (SDK) which uses correlation with a sum of absolute differences (SAD) cost (see 3.2.4). The Ensenso N35 uses a variant of semi-global matching (see 3.2.6) and incorporates a Blue (465 nm) or IR (850 nm) random dot pattern projector to aid reconstruction. The Stereolabs ZED uses a proprietary matching algorithm that requires an Nvidia CUDA-enabled graphics card. The Intel Realsense R200 also has an IR projector and performs matching up to 64 disparity levels on an onboard Application Specific Integrated Circuit (ASIC). The Mobile Robots MobileRanger also uses SAD correlation algorithm (up to 64px disparity) running on an FPGA. The MEGA-DCS and Capella are not provided with matching software. The LMI Gocator 3100 uses a coded light projection system and requires several images of an object in order to reconstruct it; while the system uses a two-camera arrangement, the cameras are spectrally filtered and conventional stereo matching is not required. Ricoh's SV-M-S1 performs stereo matching on the device at 30fps and an additional LED texture projection system is available.

2.3.7 Summary

Stereo imaging is cost effective and has been routinely performed at scales ranging from microscopic to orbital distances. The quality of the final depth image is determined by the performance of the image matching algorithm. From a purely geometric point of view stereo is largely constrained by the baseline and accuracy degradation at long ranges (relative to the baseline length). As with any imaging system there is an inevitable trade-off between field of view, depth of field and range accuracy.

2.4 Laser Rangefinding

Though the term LIDAR (LIght Detection And Ranging) was introduced in the 1950s, it was not used in a laser context until the 1960s. Most early civilian experiments using LIDAR involved meteorological phenomena, such as clouds (Collis and Ligda, 1964). Terminology among disciplines is varied, the term LADAR (LAser Detection And Ranging) is also used, analogous to RADAR (RAdio Detection And RAnging); in this text LIDAR will be used. Other sources refer to time of flight (ToF) LIDAR for both pulsed and coherent methods, here ToF explicitly refers to pulsed LIDAR. Similarly this text is limited exclusively to LIDAR used for range determination.

Around the end of the 1960's, the first commercial rangefinders became available to surveyors, such as the AGA Model 8 (Scholdstrom, 1969). These electronic distance measurement (EDM) devices used phase shift to determine distances. Modern LIDAR scanners can operate at up to 1Mpts/sec and incorporate position awareness via GPS and inertial measurement units (IMUs). Wireless connectivity, high resolution panoramic cameras and onboard high capacity flash storage are increasingly common.

LIDAR is an active sensing method, using pulsed or coherent laser light to measure distance directly. The determination of a 3D point is straightforward once a distance has been measured, provided the direction of the beam is known. COTS systems tend to use either visible red (around 650nm) or infrared ($1 - 1.5\mu$m) light and eye safety is almost always the limiting factor in terms of laser power (Campbell et al., 2013). For stationary applications, rotating mirrors or prisms are used to direct the beam to cover a hemispherical region of interest. The positions of the mirrors, measured for example using an encoder, are used to determine the beam direction. Distance measurements are therefore taken relative to the origin of the scanner, however it is defined by the manufacturer.

The vast majority of imaging LIDAR systems require the beam to be scanned. For applications on moving vehicles, a 2D scanner is sufficient as it is possible to exploit the motion of the vehicle itself to provide data in the third dimension. Figure 2.4 shows two possible mechanisms for beam scanning.

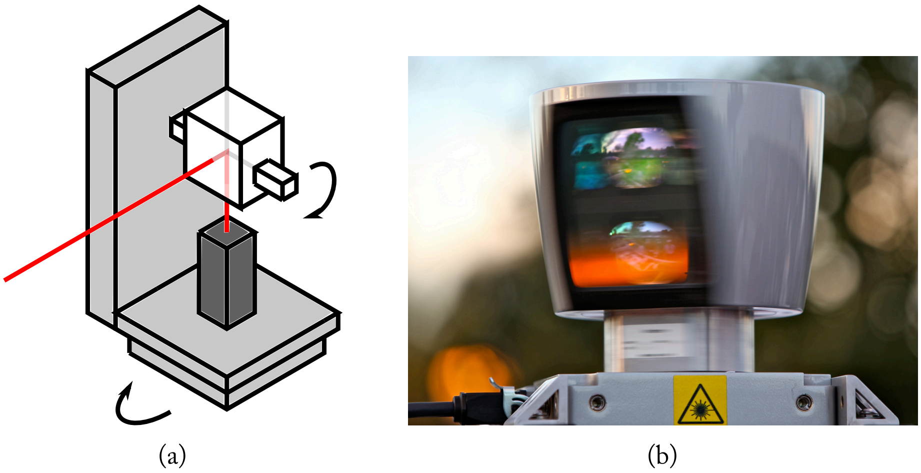


Figure 2.4: (a) Typical hardware arrangement for a scanning LIDAR. The laser is fired at a mirror or prism which enables 2D in the vertical axis. The 2D assembly is mounted on rotation platform which provides motion in the third axis. (b) A vehicle mounted Velodyne HL-64 scanning LIDAR. (Source: Steve Jurvetson, Flickr, CC License)

The power received by a LIDAR system from a target at distance R is given by (Shan and Toth, 2008):

$$P_R = \frac{1}{4} \cdot P_T \cdot \tau_{\text{total}} \cdot \rho \left(\frac{D}{R} \right)^2 \tag{2.20}$$

where P_T is the transmitted power, ρ is the reflectivity of the target, D is the aperture diameter, τ_{total} is the transmission factor due to atmospheric attenuation and instrument efficiency.

There are three classes of laser measurement system that will be discussed:

1. Time of flight LIDAR; direct measurement of photon travel time
2. Coherent LIDAR; indirect measurement of distance via frequency or amplitude information
3. Flash LIDAR or LADAR; area time of flight sensors which use lasers as the illumination source

2.4.1 Time of Flight

The distance to a target can be obtained by sending a pulse of laser light, measuring the time t until an echo is detected and using that time to calculate a range, r:

$$R = \frac{ct}{2n} \tag{2.21}$$

where n is the refractive index, and c the speed of light in vacuum:

$$\Delta R \geq \frac{t\Delta c}{2n} + \frac{c\Delta t}{2n} = \frac{c\tau}{2n} \tag{2.22}$$

where τ is the pulse width. The first term is zero for a beam path with a single refractive index. This places a lower bound on the accuracy since other instrumental errors such as pulse jitter are ignored.

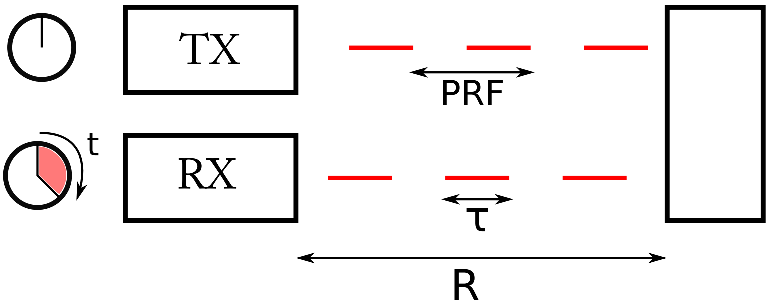


Figure 2.5: Hardware arrangement for a pulsed time of flight LIDAR.

The pulse repetition frequency (PRF) determines both the maximum range of the LIDAR and the necessary acquisition rate. In order to unambiguously differentiate between returned signals, there must be enough time between successive pulses for the previous pulse to return to the detector, though some systems can track multiple pulses. The desired maximum range therefore puts a hard limit on the sampling time per point, assuming the laser provides a high enough SNR to only require one return per point. Multiple pulse returns may be averaged to improve range, if it can be assumed that the returned distances are normally distributed.

Over large distances, for example in satellite remote sensing, the divergence of the laser pulse causes it to spread out to such an extent that it is reflected from objects at significantly different distances. This is observed as additional pulses at the detector and is used, for example, to simultaneously map tree canopy heights and the underlying terrain (Nelson et al., 1984; Hancock et al., 2011, 2012).

The speed of light poses a timing challenge for electronics - a 1 mm range difference alters the time of flight by 6.6 ps. Nevertheless, pulsed LIDAR systems allow measurements of very long distances with high relative accuracy. For instance the Earth-Lunar distance is now known to an accuracy of millimetres (Currie et al., 2011), though this is the result of decades of study. In principle the maximum range is limited only by SNR and the time needed to wait for the return signal.

A selection of COTS TOF imaging systems is given in Table 2.2. The systems listed are, in most cases, the current flagship product from each manufacturer. With the exception of the Velodyne HL-64 (Figure 2.4b) which achieves a high measurement rate by multiplexing 64 transmitter/reciever pairs, all systems are scanned single-beam. This system is also unique among these scanners in that it is designed to be vehicle-mounted for real-time imaging.

Manufacturer	**System**	**Range (m)**	**Meas. Speed (pts/sec)**	**Range Accuracy**
Leica Geosystems	ScanStation P40	0.4-270	1 M	6 mm at 100m
Riegl	VZ-6000	5-6000	220k	15 mm at 150 m
Trimble	TX8	0.6-340	1 M	2 mm
Teledyne Optech	ILRIS-LR	6-3000	10k	4 mm at 100 m
Velodyne	HL-64	1-50/120†	1.3 M	20 mm
Maptek	I-Site 8820	2.5-2000	80k	6 mm

Table 2.2: Selection of commercially available ToF imaging systems. All operate in the NIR. Accuracies and acquisition rates are quoted by the manufacturer and should be considered best-case. † 50 m for pavement, 120 m for cars/foliage.

2.4.2 Amplitude modulated LIDAR

Coherent, continuous wave (CW) LIDAR provides an alternative means of calculating range via phase differences between the outgoing and incoming laser beam (Srinivasan and Lumia, 1989; Adams, 1993). In an Amplitude Modulated (AMCW) system, the outgoing beam amplitude is modulated at a particular frequency.

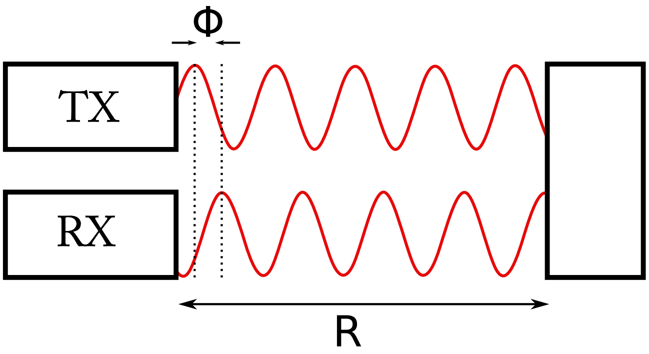


FIGURE 2.6: Schematic of hardware arrangement for an amplitude modulated continuous wave LIDAR.

In a homodyne arrangement, the transmitted pulse is used as a reference, mixed with the return signal; both signals have the same frequency. If a heterodyne system is used, the transmitted beam is compared to a reference with a slightly different frequency. The detected signal in this case is a beat signal, at the difference of the two frequencies. The phase difference between the received beam and the outgoing beam, ϕ, is related to the range as:

$$R = \frac{\phi}{4\pi}\lambda + n\lambda \tag{2.23}$$

where n represents some multiple of the wavelength. The range resolution is determined by the smallest change in phase, $\Delta\phi$ that can be measured (Wehr and Lohr, 1999). As the signal is periodic, there is an inherent ambiguity if the phase is greater than 2π:

$$R_{max} = \frac{1}{4\pi}\lambda\phi_{\mathrm{max}} = \frac{\lambda}{2} \tag{2.24}$$

This range is called the ambiguity interval and is lessened due to atmospheric attenuation. Multiple modulation frequencies or repeat measurements with the transmitted beam phase-shifted may be used to reduce ambiguity. In the multiple frequency case the longest frequency gives the maximum range and the shortest frequency determines the range resolution.

As the peak laser power is significantly lower than for pulsed systems, the maximum range is correspondingly less although ambiguity is more often the limiting factor. Surveying systems, such as Leica Geosystem's HDS7000, with typical ranges of a few hundred metres are available. AMCW LIDARs are able to operate beyond the laser coherence length if a single mode laser is used, but the SNR rapidly decreases at larger distances (Harris et al., 1998). Table 2.3 highlights a number of commercially available AMCW systems.

Zöller and Frolich (Z+F) manufacture a range of AMCW scanners; the current flagship product is the Imager 5010 series with an ambiguity interval (equation 2.23) of 187 m and an acquisition rate of 1Mpt/s. The quoted range accuracy is 0.2-10 mm depending on target distance (10-100 m) and reflectance (14-80%). This model is a considerable improvement over the previous Imager 5006 series with an ambiguity interval of 79 m. Faro manufacture two models of AMCW LIDAR under the Focus^{3D} brand. The X 130 has an ambiguity interval of 0.6-130 m while the X 330 is usable from 0.6-330 m. Both are quoted as having a range accuracy of 2 mm and have measurement

Manufacturer	System	Range (m)	Meas. Speed (pts/sec)	Range Accuracy
Leica Geosystems	HDS7000	0.4-270	1 M	6mm at 100 m
Leica Geosystems	DISTO D810†	0-80	N/A	1-2 mm
Zöller and Frolich	Imager 5010	187	1 M	0.2-10 mm
Faro	X 330	0.6-330	976 k	2 mm at 10-25 m
Surphaser	SR 100	1-7	1 M	0.3 mm at 3 m
Surphaser	105HS	1-130	1 M	0.7 mm at 15 m
Dimetix	FLS-C10†	0.05-500	20	1mm

Table 2.3: Selection of commercially available AMCW imaging systems. Accuracies and acquisition rates are quoted by the manufacturer and should be considered best-case. † 1D sensor, handheld. Longer ranges are only achievable with retroreflectors.

speeds of 976kpts/s. Like the Z+F systems, the scanners have location and orientation sensors and a panoramic camera. Surphaser manufacture high accuracy (sub-mm) systems targeted towards industrial metrology and offer both short range (1-7 m) and mid-range (1-130 m) solutions.

Handheld 'laser tape-measures' such as Leica's DISTO range have become popular among consumers and industry alike due to their high accuracy (mm) and comparatively low cost compared to scanning stations. These devices are targeted towards surveying or home improvement. The DISTO D810 measures up to 200 m with an accuracy of 1mm. Dimetix manufacture a range of industrial 1D LIDAR units based on Leica technology. The FLS-C10 measures up to 500 m with a reflective target with an accuracy of 1mm at up to 20pts/s.

2.4.3 Frequency modulated LIDAR

In a frequency modulated (FMCW) system, the laser frequency is continually modulated and distance is determined by measuring the beat frequency between the transmitted and echoed light (Amann et al., 2001; Pierrottet et al., 2008). The transmitted light is modulated using a triangular (linear) ramp enabling both distance and velocity of the target to be determined; this is particularly important for a vehicle mounted system where there is typically relative motion between the target and the detector. A sawtooth modulation may also be used, but in this case only distance determination is possible. The frequency excursion is given as Δf and the modulation frequency as f_{mod}.

Due to the Doppler effect, two beat frequencies are observed and the range and velocity of the target may be measured. The range is given as:

$$R = \frac{c(f_1 + f_2)}{8\Delta f \cdot f_{\text{mod}}} \tag{2.25}$$

and the velocity as:

$$v = \frac{\lambda(f_1 - f_2)}{4} \tag{2.26}$$

Depending on the modulation frequencies, the beat frequency can be selected for the desired range. This is convenient as the detector bandwidth can be substantially reduced within a range of interest as the beat frequency may be much lower.

Like AMCW, FMCW LIDAR has an ambiguity interval corresponding to the time it takes to linearly change (ramp) the frequency from f_1 to f_2. However, if longer range is required, the ramp time is simply extended. Resolution is determined by the laser frequency excursion, Δf and the linearity of the frequency ramp.

There are very few commercially available FMCW systems. Nikon Metrology's MV350 is designed for large scale industrial inspection in the aerospace and maritime industries. It features a range of up to 50 m with an accuracy of 300 μm at 30 m. Massaro et al. (2014) favourably compared the Bridger Photonics HRS-3D against an AMCW scanning system (Riegl VZ-400) for defence applications, but the authors note that the unit is significantly bigger than the AMCW scanner and has a slower acquisition rate.

2.4.4 Flash LIDAR

Pulsed laser imaging systems, flash LIDAR or LADAR, have been demonstrated using CMOS single photon avalanche photodiode (SPAD) arrays (Niclass and Charbon, 2005). These systems use single photon counting techniques to take an array of distance measurements simultaneously. The advantage is clear; imaging with the range and accuracy of conventional LIDAR, but without the need to scan.

Commercial flash LIDAR systems have been space-qualified and flown on the Space Shuttle for docking experiments; an Advanced Scientific Concepts (Santa Barbara, CA) DragonEye sensor was used to image retro reflectors on-board the International Space Station (Stettner, 2010). There has been interest in flash LIDAR systems for a variety of space borne applications including guidance and surface navigation (Pereira do Carmo, 2011), terrestrial autonomous navigation (Juberts and Barbera, 2004) and defense (Halmos et al., 2001).

Niclass et al. (2011) presented a 340x90 px array with up to 128 m range and 60 mm resolution. The sensor uses correlation to determine when the laser pulses return to within 208 ps. The cost of these systems is prohibitively high for most users.

Princeton Lightwave produce several cameras based on InP/InGaAs Geiger-mode avalanche photodiodes capable of timing photons with a resolution of up to 250 ps with a timing jitter of 500 ps. Models with 32x32 px and 128x32 px are available commercially. The same technology is used in systems produced by Spectrolab under the brand name SpectroCam. The range of the systems are between 75-1000 m and are designed to used with common SWIR laser wavelengths, e.g. 1030 nm or 1064 nm.

2.4.5 Theoretical accuracy

LIDAR range accuracy is inversely proportional to the square of the SNR (Baltsavias, 1999):

$$\epsilon_R \propto \frac{1}{\sqrt{\text{SNR}}} \tag{2.27}$$

The SNR is given by:

$$\text{SNR} \propto \frac{\Re_0 \cdot P_T}{B} \tag{2.28}$$

where $\Re_0$ is the unity gain responsivity of the receiving photodiode and B is the effective noise bandwidth (dependent on the sampling rate and pulse width). For pulsed systems, accuracy is determined by the response time of the photodetector and the performance of the timing circuits in the device. A small deviation in the counting system will lead to large offsets in the measured distance. This has limited commercial devices, typically using picosecond rise-time photodiodes, to accuracies on the order of 1mm for 1D measurements. The signal to noise ratio for a pulsed system is:

$$\epsilon_R \propto \frac{R}{\sqrt{\rho \cdot \tau_{\text{atmos}}}} \tag{2.29}$$

where τ_{atmos} is the transmission through the atmosphere. Laboratory accuracy has recently improved to sub-femtosecond timing (< 100µm scale) using optical cross-correlation (Lee et al., 2010).

For coherent LIDAR, the accuracy is determined by the ability to accurately measure phase difference in the case of AMCW, or beat frequency in the case of FMCW. The signal to noise ratio is:

$$\epsilon_R \propto \frac{R^2}{\rho \cdot \tau_{\text{atmos}}} \tag{2.30}$$

2.4.6 Compared to passive systems

The most obvious advantage of active measurement systems like LIDAR is that no ambient light is required. This enables systems to operate at night or in poor visibility. This is exploited by space-borne sensors which can operate continuously, compared to passive camera systems which require sunlight.

The number of photons returned, and therefore the SNR, is dependent on both atmospheric properties and surface properties. Diffuse surfaces or surfaces which strongly absorb the illuminating wavelength are challenging to measure, as are transparent or translucent materials[2]. However a (passive) stereo system may easily cope with a diffuse target if the image has sufficient local texture. Boehler et al. (2003) compared a number of LIDAR systems and observed systematic errors when measuring certain object reflectivities. These included surfaces with different levels of greyness and specularly reflective foils. For example, most systems under test over-estimated the distance to an orange traffic cone. Beraldin (2009) compared measurement uncertainties using three scanning LIDAR systems using targets with reflectances from 3-89%.

LIDAR scan resolution determines the ability of a system to cope with edges or depth discontinuities. On the other hand, image based systems like stereo perform optimally where there are depth boundaries as these are often coincident with large intensity gradients in the image.

2.4.7 Summary

LIDAR systems are increasingly commonplace as system costs decrease and measurement speeds improve. LIDAR is one of few methods that allows for robust, high accuracy distance measurement

[2] Reflectance curves for a variety of materials may be found in (Jelalian, 1990)

outdoors. Typically mm-scale LIDAR accuracy is possible at ranges of 10s to 100s of metres. At longer ranges (kilometres), pulsed LIDAR systems have little competition, while at shorter ranges up to several hundred metres CW systems offer superior accuracy. The vast majority of commercial systems operate either using pulsed or amplitude-modulated light; frequency modulated systems are promising, but there are few available. LIDAR has two main downsides: the first is cost, significantly higher than stereo imaging. The second is the inability for most systems to capture realtime data due to the need to scan.

2.5 Laser Triangulation

Among industrial users, laser triangulation is widely used to acquire 3D information (Hebert, 2000). A laser stripe or spot is projected onto a surface and imaged by a camera. The location of the imaged stripe changes in proportion to the variation in surface height (Figure 2.7). On production or conveyor lines this technique allows for 3D imaging by stacking successive scans. The location of the laser beam determines the distance to the surface, Z, as:

$$Z = \frac{b}{f\cot\theta - u} \tag{2.31}$$

where b is the baseline between the camera and the laser, f is the focal length of the camera, u is the position along the sensor and θ is the angle of projection.

For a fixed baseline system, the accuracy is dependent on the ability to measure θ and u (Baribeau and Rioux, 1991). Determination of u corresponds to pixel size and the ability to perform sub-pixel estimation of the beam location. Depth measurement is also limited by coherent noise (speckle) caused by the interference of scattered light with random phase.

$$\epsilon_Z = \frac{1}{\sqrt{2\pi}}\frac{\lambda}{\phi}\frac{Z}{\sin\theta} \tag{2.32}$$

where ϕ is the diameter of the lens, λ is the laser wavelength.

Laser triangulation accuracy is excellent; micron level accuracy is achievable. MTI Instruments produce a variety of surface profiling instruments. The MICROTRAK Pro 2D is a stripe triangulation system with resolution between 3 µm and 200 µm depending on the desired field of view. NextEngine produce a desktop scanner with a turntable suitable for small objects with an accuracy

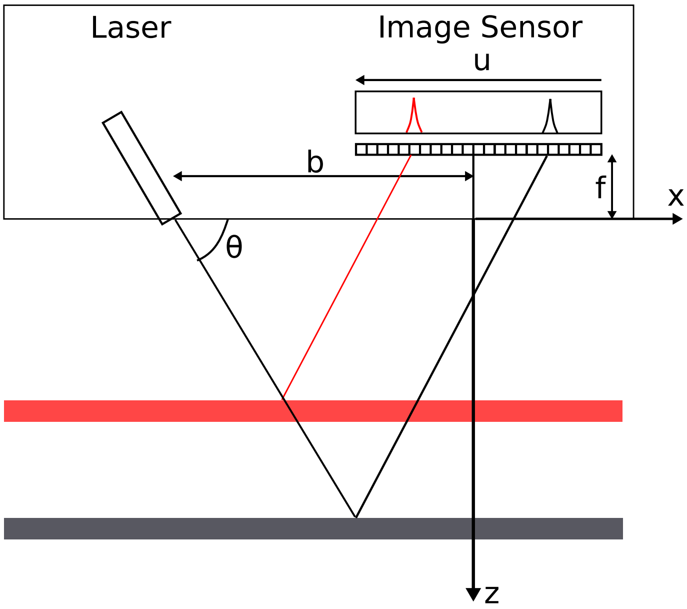


Figure 2.7: Typical arrangement for a laser triangulation system. The laser here is fixed, but it is straightforward to extend the system to allow scanning. The laser beam is seen as a stripe on an imaging sensor and peak detection is used to locate the row corresponding to the beam.

of 0.13 mm.

Handheld triangulation systems exist, though most use structured light. The Polhemus (USA) FastScan Cobra C1 incorporates an electromagnetic tracking system that determines the orientation of the scanner wirelessly. Scan resolution is 0.5 mm at 200 mm with an accuracy of 0.13 mm.

More advanced systems such as Faro's Edge ScanArm HD place the scanner at the end of a mechanical arm. Encoders in the joints determine where the scanner is pointing so the resulting fused scan is very accurate. The ScanArm HD has a working volume of 1.8 m and is accurate to 0.034 mm (or larger volumes with decreased accuracy). Repeatability as low as 0.024 mm is demonstrated.

Triangulation is particularly attractive for industrial users as the systems typically have no moving parts and give extremely accurate, dense data. Both fixed and handheld systems are widely available. The working distance of most systems is < 4 m, but this is generally acceptable for scanning on process lines. Accuracies on the order of microns are achievable, but depths of field are rather small (a few centimetres)[3] .

Laurin et al. (1999) developed a ranging system for space operations that combined a fast

[3] http://www.keyence.co.uk/products/measure/laser-1d/lk-g5000/specs/index.jsp, accessed 20/9/16

laser triangulation system with time of flight capability at longer range (above 10 m). The laser was scanned in a Lissajous pattern and took several minutes to image an object at 4000x4000 px resolution.

2.6 Time of Flight Cameras

Time of Flight Cameras (ToFCs) are a relatively recent class of 3D imaging device. Modulated infrared light is used to illuminate an entire scene and a CMOS sensor is used to determine the per-pixel phase shift of the returned light, thereby giving a distance. Early devices were limited by comparatively poor resolution with sub-MP sensors, however they allowed realtime 3D imaging before realtime stereo became computationally feasible. Their popularity stems largely from use in gesture/pose recognition systems where short-range coarse depth information is sufficient (Kollorz et al., 2008; Ganapathi et al., 2010).

The principle is largely similar to AMCW LIDAR, except banks of infrared LEDs are used instead of a laser. The illumination is either a pulsed (square wave) or continuous wave light source. Although in principle direct ToF systems are possible, the vast majority of ToFCs use indirect methods to calculate the ToF.

Modulation is normally performed at several tens of MHz. Schwarte et al. (1997) and Lange et al. (1999), presented similar sensors: the Photonic Mixing Device/PMD and the lock-in CCD. Both are imaging sensors where each 'multitap' pixel is capable of storing 4 (or more) distinct amounts of charge. By sampling the light at four times during each modulation period, it is possible to reconstruct the phase and therefore distance at each pixel.

For such a system, Lange and Seitz (2001) calculate the range accuracy due to shot noise as:

$$\epsilon_Z = \frac{Z}{\sqrt{8}}\frac{\sqrt{B}}{2A} \tag{2.33}$$

Where A is the number of photoelectrons per pixel generated by the modulated light source and B is the number of photoelectrons per pixel from ambient light and other noise sources. $Z = c/2f_{\mathrm{mod}}$ is the non-ambiguous maximum range with modulation frequency f_{mod}. This is the absolute accuracy limit for a 4-sample system.

2.6.1 Calibration Issues

ToFCs are prone to a number of systematic errors (Lindner et al., 2010; Foix et al., 2011). Depth distortion occurs due to imperfect sinusoidal modulation of the illumination; this is correctable by a look up table (LUT) as it is a distance dependent effect. Temperature bias is observed requiring cameras to be temperature stabilised during operation. Variation in integration time, used to determine the phase of the return light, causes a depth bias. It is unclear what causes this effect; one solution is to either use a fixed integration time or to perform several calibrations with different times. Imperfect pixels can produce a rotation of the image plane, some manufacturers supply LUTs to correct this.

Depth accuracy is strongly dependent on the returned light intensity and often this is lower in the corners of the image. Underexposed areas tend to produce overestimated depths and ToFCs are in general very sensitive to SNR. ToFCs are also affected by multipath returns, light scattering from close objects and motion blurring. Lindner et al. (2010) goes as far as stating that no ToFC should be used without calibration and suggests using a modified chessboard-style calibration pattern that includes squares with varying levels of greyness.

Table 2.4 shows a selection of commercially available ToFCs. High resolution (MP) sensors have recently begun to be mass produced, for instance the latest Microsoft Kinect sensor, a video gaming accessory. Due to illumination constraints and phase ambiguity, the usable range is limited to under 4.5m. Fankhauser et al. (2015) tested the Kinect outdoors and found that performance, with degraded accuracy, was acceptable in overcast conditions, but not in direct sunlight.

Manufacturer	System	Resolution	Range	Framerate	Accuracy
Swissranger	SR4500	175x144px	9m	30fps	20mm
Pmdtec	19k-S3	160x120px	2m	90fps †	5mm
Odos Imaging	Real.iz VS-1000	1280x1024px	9m	30fps	10mm
Fotonic	X E-Series	160x120px	10m	52fps	10-40mm
Softkinetic	DS325	320x340px	1m	60fps	14mm
Microsoft	Kinect v2	512x424px	0.5-4.5m	30fps	< 3.1mm ‡

Table 2.4: Selection of commercially available ToFCs. Accuracies and acquisition rates are quoted by the manufacturer and should be considered best-case. † The 19k-S3 is an OEM sensor, this framerate is achieved using the reference design. ‡ Accuracy estimate by Fankhauser et al. (2015).

2.7 Summary

There is no single imaging solution that can provide realtime, dense, accurate 3D data over a large depth of field, indoors and outdoors. Stereo and LIDAR are usable over a wide range of distances depending on hardware configuration. Laser triangulation and ToF cameras are only suitable for short distances (< 10 m typical).

Stereo and LIDAR still dominate the 3D imaging sector and remain the only robust systems suitable for outdoor use. Time of flight cameras show promise, particularly for environments where controlled illumination is not a problem; there is limited evidence to support outdoor use, but even with spectral filtering most units cannot cope with sunlight. In terms of absolute accuracy over a large depth of field coherent LIDAR is unrivalled (millimetres over hundreds of metres), but scanning times are still a limiting factor. Until time of flight camera sensors improve in resolution, stereo imaging still provides the densest data per frame although the speed at which 3D data can be calculated depends on the matching algorithm used.

Stereo Matching and Data Fusion 3

3.1 Overview

Chapter 2 outlined the performance of stereo systems in terms of their geometry, such as the choice of camera separation (baseline) or focal length. While camera geometry places constraints on system resolution and accuracy, the final quality of the depth map produced by a stereo system is determined by how well the images are matched. Stereo matching is a mature research field and there are many algorithms to choose from.

The first part of this chapter discusses the different classes of stereo matching algorithms and the cost functions that are used to determine image similarity. Stereo algorithm benchmarking is discussed, as it is necessary to be able to compare the performance of different methods. Finally, an overview of progress towards real time matching is given.

The second part of this chapter examines data fusion as a process to provide improved 3D data from multiple imaging sensors. A general definition of data fusion is given with a particular focus on methods which enhance stereo matching using additional range data from a LIDAR or ToFC. A literature review of fusion methods is given with an outlook towards novel ways of combining stereo and LIDAR data.

3.2 Stereo Matching

The purpose of a stereo matcher is to determine which pixels in one image correspond to the pixels of another image, the correspondence problem. The output from a matching algorithm is a list of labels for each pixel in one image indicating the corresponding location in the other image. This output is called a disparity map. Typically a disparity map labels every pixel, though in some cases a coarser map is sufficient. When combined with camera calibration information, this disparity map is sufficient to reconstruct the scene in 3D through application of equations 2.17 and 2.18 for each pixel with known disparity.

There are a very large number of stereo matching algorithms, with more proposed each year (Scharstein and Szeliski, 2002; Mroz and Breckon, 2012). In their evaluation of several common algorithms Scharstein and Szeliski suggested the following generic structure which remains relevant today: Firstly, a cost function is defined to measure similarity between two pixels. Next, costs are aggregated over a support region or window for a locally robust similarity measure. The best disparity for that pixel is chosen using some selection process. Finally, the disparity map may be checked for consistency, holes may be filled and so on. Some steps may be omitted, for instance a global algorithm may not perform cost aggregation.

First, an overview of stereo matching costs is given since these are not unique to a particular matching algorithm. Next, stereo benchmarking on ground truth data is discussed as this is critical for fair comparison of algorithms. Finally, several classes of matching algorithm are detailed:

- Local matching algorithms; correspondences are determined based on similarity in a small neighbourhood surrounding each pixel
- Global matching; correspondences are determined based on a minimisation process that tries to produce an optimal disparity map for a given pair of images
- Semi-global matching; a recent, efficient approximation of global matching
- Region growing; an initial list of correspondences is used to 'seed' the disparity map.

3.2.1 Matching costs

A cost function is used by a matching algorithm to score potential correspondences in terms of their similarity; the output is a single number. Matching cost is determined by comparing one pixel $\mathbf{x_1}$ in an image I_1 with another $\mathbf{x_2}$ in another image I_2. The simplest cost function is the absolute intensity difference:

$$C = |I_1(\mathbf{x_1}) - I_2(\mathbf{x_2})| \tag{3.1}$$

The squared intensity difference is also common. These functions assume that each image is radiometrically similar - two corresponding pixels have the same brightness. Birchfield and Tomasi (1998) proposed a popular cost which is insensitive to image sampling by linearly interpolating the intensity at each pixel.

In order to increase the robustness of these costs, the input image may be filtered or transformed (Hirschmuller and Scharstein, 2007). Non parametric transforms are defined by their dependence on the ordering of data values, not the values of the data themselves. Zabih and Woodfill (1994) suggested two transforms for image matching: rank and census. The rank transform replaces each pixel with the number of pixels in a local neighbourhood with a lower intensity. The transformed image is matched using a standard correlation method as described above. The census transform is similar, but labels each pixel with a bit string which describes whether pixels in the neighbourhood have a lower intensity than the central one. Census transformed images are efficiently matched using the Hamming distance, the number of bits that differ between two strings.

In the aggregation step, pixel costs within a neighbourhood or window are combined to produce a more robust similarity measure. At its simplest, this is the sum of pixel costs within the window such as the sum of absolute differences (SAD) or sum of squared differences (SSD). Normalised cross correlation (NCC) is a standard local cost that is insensitive to gain and bias:

$$C = \frac{\sum_{\mathbf{u,v}} I_1(u,v) \cdot I_2(u+d,v)}{\sqrt{\sum_{\mathbf{u,v}} I_1^2(u,v) \cdot \sum_{\mathbf{u,v}} I_2^2(u+d,v)}} \tag{3.2}$$

for points $(u, v) \in w_1, w_2$, two windows in the images I_1 and I_2.

Stereo matchers rely on cost functions returning, over the disparity range, a strong maximum or minimum at the feature of interest. This occurs when the intensity variation of the region is strong and unique, i.e. there is good texture in the image. Regions with homogenous or repetitive texture can cause issues as the matching costs will be similar for multiple disparity values. This is a particular issue for local algorithms.

3.2.2 Matching assumptions

In order to attempt to reduce the number of mismatched pixels, algorithms may make a number of assumptions (Marr and Poggio, 1979; Grimson, 1985; Brown et al., 2003):

- Uniqueness constraint: each pixel corresponds to at most one other pixel (or it is occluded)
- Continuity constraint: the variation in disparity values surrounding a pixel should be smooth.
- Epipolarity: if the images are rectified, matches should lie along the same row in the corresponding image. This is assumed by almost all matchers as it significantly reduces the search space for each pixel.
- Ordering constraint: points along an epipolar line appear in the same order in each image.
- Radiometric similarity: each image is exposed under the same conditions and all surfaces are Lambertian scatterers.

The epipolar constraint is primarily applicable to close-range imaging systems. In aerial or orbital imagery, it is sometimes not possible to obtain a fundamental matrix that maps points in one image to lines in another. Algorithms developed specifically for this kind of imagery, such as Gotcha (section 3.2.7) do not necessarily require rectified images as an input.

Stereo matchers should ideally be able to identify occluding regions in an image which implies the ability to differentiate between a mismatched pixel and a pixel with no correspondence in the other image. A simple method for occlusion detection is enforcing left-right consistency: the images are matched from left to right and then from right to left (Zitnick and Kanade, 2000). The resulting disparity maps should be the same, but negated. Any pixel with a disparity that does not agree in both directions is marked as occluded or unmatched.

3.2.3 Stereo Benchmarking

As stereo performance is dependent on the geometry of the cameras, algorithms are best benchmarked by using images with known ground truth. Dense truth imagery is produced using either LIDAR information such as the KITTI dataset (Geiger et al., 2013) or using structured light, such as Middlebury (Scharstein and Szeliski, 2002, 2003; Scharstein and Pal, 2007; Hirschmuller and Scharstein, 2007; Scharstein et al., 2014). Other approaches have used calibration objects with known geometries such as planes or spheres (Ahmadabadian et al., 2013).

KITI imagery is taken from a vehicle mounted stereo bar (Figure 3.1) and is targeted towards algorithms for autonomous navigation. Ground truth is available for some stereo pairs and is generated from an onboard Velodyne LIDAR. Explicit ground truth accuracy is not given in the original paper, but is presumably related to the camera-LIDAR cross-calibration and the intrinsic performance of the LIDAR (±2 cm). The LIDAR data is co-registered to the image data with some manual intervention.

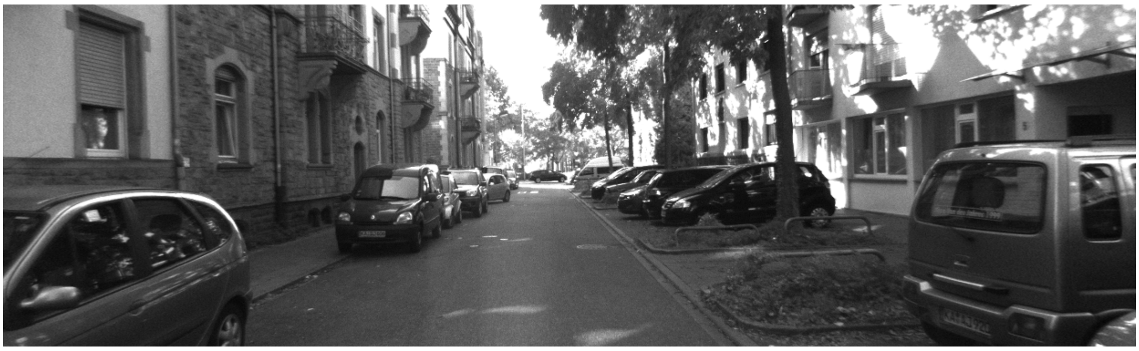
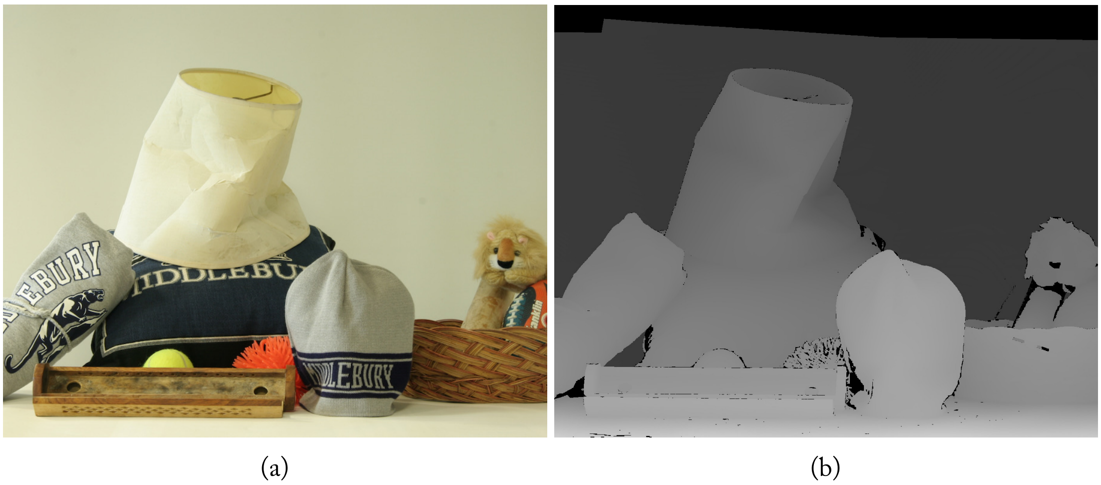


Figure 3.1: A representative image taken from the KITTI dataset. The images are generally accepted to be more challenging than Middlebury due to more varied illumination and the presence of specular reflections on vehicles.

Middlebury provides images of a variety of indoor scenes (see, for example Figure 3.2; each scene typically contains multiple illumination variants and the 2014 dataset contains images with imperfect rectification. Both datasets are freely available to download and users have the option of submitting code to be benchmarked on images where no public ground truth is available.



(a)

(b)

Figure 3.2: (a) Left stereo image and (b) high resolution ground truth disparity map from Middlebury 2012.

Both benchmarks score algorithms based on the percentage of incorrectly matched pixels, average disparity error and algorithm runtime.

3.2.4 Local stereo matching

Local matching algorithms consider only a small window around each pixel when computing match cost. The match for each pixel is independent of the match for every other pixel. A reference window in one image is compared to a number of comparison windows in the image (Figure 3.3). The best disparity is typically chosen with a greedy algorithm such as winner-take-all (WTA) where the window with the minimised or maximised cost is chosen as the match:

$$d_{\text{match}} = \arg\min C(u, v, d) \tag{3.3}$$

The neighbourhood is typically rectangular with a fixed size, but it may also be adapted to respond to intensity boundaries in the image (Kanade and Okutomi, 1994). Large windows produce smooth disparity maps at the expense of recovering short-scale detail.

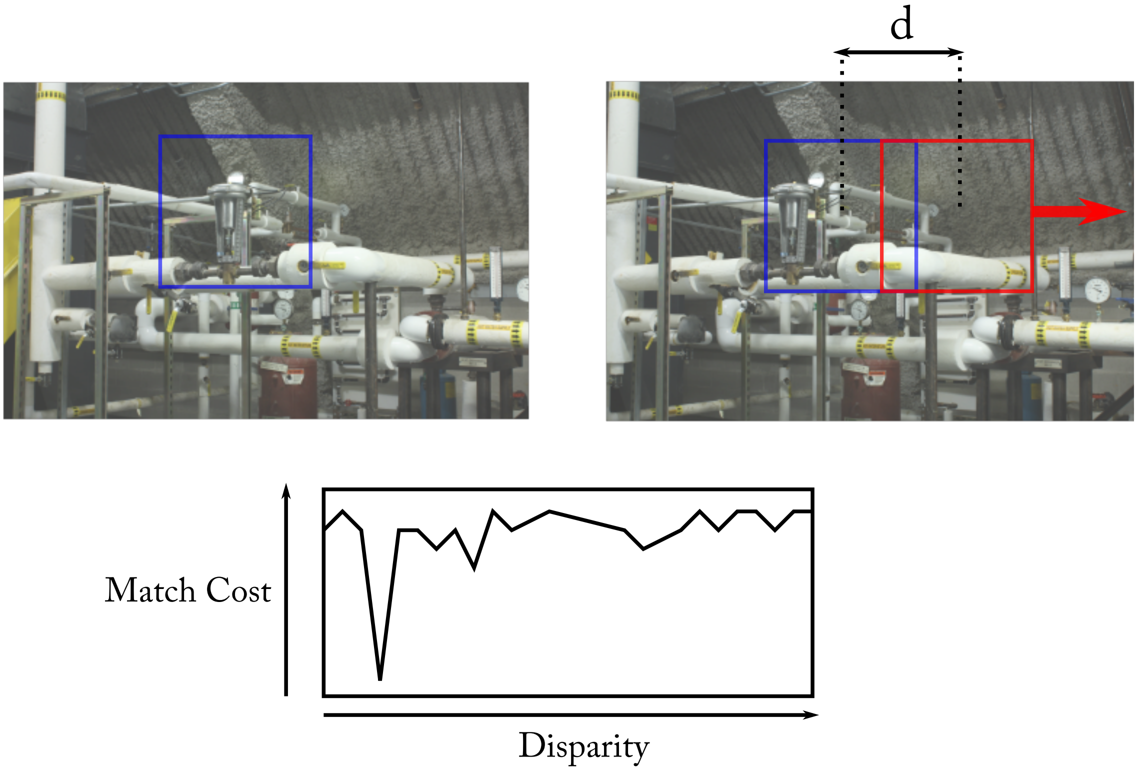


Figure 3.3: The Pipes stereo pair from Middlebury 2014. An example neighbourhood (oversized for clarify) is shown in blue. The sliding window used for cost aggregation is shown in red in the right image.

Local algorithms are simple to implement and are straightforward to paralellise as each comparison is independent. The time taken to construct the disparity map is proportional to the size of the image itself, the disparity range of interest, d_r, and the window radius r giving a naive complexity of $O(hwd_rr^2)$. As only local information is used to match pixels, local algorithms tend to perform poorly in regions with little texture due to ambiguous matches.

3.2.5 Global stereo matching

Global algorithms aim to generate a disparity map that minimises some function (often called an energy). If the minimisation is limited to 1D, that is along the rows of the image, then it may be efficiently solved using dynamic programming (DP) (Ohta and Kanade, 1985; Veksler, 2005). Dynamic programming algorithms break a problem down into sub-problems and store the results so that they do not need to be re-computed. Unless intra-row consistency is taken into account, the disparity map shows characteristic streaking artefacts. Parallelisation is straightforward as the computation for each row (or group of rows) is independent.

Extending the optimisation to compute an optimal disparity map for the entire image is computationally complex[1] Scharstein and Szeliski (2002), but the problem is solvable if the energy function is chosen carefully Kolmogorov and Zabih (2004). The disparity map is calculated using iterative methods. Global stereo algorithms have included Markov random fields, belief propagation (Sun et al., 2003) and graph cuts (Kolmogorov and Zabih, 2001). Global matching algorithms typically outperform both local and dynamic programming methods at the expense of computation time and complexity.

3.2.6 Semi-global stereo matching

Semi-global matching (SGM) (Hirschmuller, 2008) is an extension of dynamic programming, exploiting the fact that 1D optimisation can be solved efficiently. SGM performs cost computation along a number of paths radiating out from the pixel of interest. The performance of SGM is excellent, with results that are comparable to global matchers but with better computational cost (Hirschmuller and Scharstein, 2007). As of 2015, the best performing algorithm in the Middlebury stereo benchmark uses a combination of SGM and convolutional neural networks (Zbontar and LeCun, 2016). Hirschmüller's original algorithm, as of 2015, ranks eighth.

[1] 2D disparity map optimisation falls into the class of Nondeterministic Polynomial-time (NP)-hard problems which are at least as computationally complex as those in NP Knuth (1974).

3.2.7 Region growing stereo matching

Region growing algorithms take an initial set of seed correspondences (also called tiepoints or ground control points) and attempt to 'grow' the disparity map from these points. The Gruen-Otto-Chau Adaptive Least Squares Correlation (Gotcha) (Otto and Chau, 1989; Shin and Muller, 2012). Lhuillier (1998) proposed a similar algorithm that introduces 'seed areas' which describe regions of colour uniformity before propagating the initial correspondences into textured regions. The tiepoints used by Gotcha are either selected manually or are automatically generated using a feature detector such as the Scale Invariant Feature Transform (SIFT) (Lowe, 2004). The complexity of region growing algorithms is not constrained by a disparity range which adds some flexibility over local/global methods, however dense matching requires good local texture for the regions to expand into.

This method has proved to be accurate and robust, for example when applied to spacecraft (Day and Muller, 1989; Thornhill et al., 1993); close-range industrial (Muller and Anthony, 1987; Muller et al., 1988); close-range medical (Deacon et al., 1991) and Martian rover imagery (Shin and Muller, 2012).

The algorithm uses Adaptive Least Squares Correlation (ALSC) (Gruen, 1985) to refine and determine correspondences to sub- pixel accuracy, providing a disparity estimate and a confidence score. If a tiepoint is successfully matched, its neighbouring pixels are added to a priority queue, sorted by match confidence. ALSC is performed on the neighbours of the highest confidence tiepoint and any matches are added to the tiepoint queue. The process iterates until the queue is empty. Thus the disparity is grown from the initial seed points, preferentially matching from the regions with highest confidence.

Further technical details of the algorithm are given in Section 6.2.

3.2.8 Realtime stereo

By 2003 the best algorithms already performed very well on reference imagery. Real-time solutions had been demonstrated on CPUs, but using low-resolution (320x240 px) images and small disparity ranges of up to 32 px. Global methods were too demanding to run in real-time on the hardware of that period and so there was a preference for local algorithms.

Real-time algorithms generally require the use of parallel programming. For most algorithms the matching speed is proportional to the number of processing cores available. Almost all new discrete GPUs have the ability to perform general purpose computing and contain a large number of cores. Algorithms for GPU matching are normally written in either the Open Computing Language (OpenCL) [2] or Nvidia's Compute Unified Display Architecture (CUDA) [3].

Stereo matching algorithms implemented on GPUs include correlation, semi-global matching (Ernst and Hirschmuller, 2008), dynamic programming (Wang et al., 2006) and belief propagation (Liang et al., 2011).

Lazaros et al. (2008) presented a good overview of hardware implementations of stereo matchers. Their focus was largely on Field Programmable Gate Array (FPGA) based systems as they provide higher performance than GPUs without the high expense of moving to ASICs. Due to the simplicity of the algorithm, the majority of FPGA matchers use correlation matching with SAD costs. Typical image resolutions are VGA (640x480 px) with speeds of around 30 fps.

Jin and Maruyama (2012) demonstrated a two pass algorithm using a census-like cost By 2003 the best algorithms already performed very well on reference imagery. Real-time solutions had been demonstrated on CPUs, but using low-resolution (320x240 px) images and small disparity ranges of up to 32 px. Global methods were too demanding to run in real-time on the hardware of that period and so there was a preference for local algorithms.

Larger disparity ranges require larger FPGAs that inevitably increase system cost. Kalarot and Morris (2010) presented a dynamic programming matcher on an Altera Stratix III FPGA that operated at up to 128 disparity levels, but could not scale their design to 256. The same algorithm running on a GPU did not suffer this limitation, but the performance was slower by approximately 50 %.

TYZX (recently acquired by Intel) produced a stereo system based on the DeepSea 2 ASIC (Woodfill et al., 2004). The image size is up to 512x2048 px, with correlation matching at 30 fps up to a maximum of 200fps at 512x480 px. The disparity search window is 52 px and the specified range is 2.7-35 m.

[2] `https://www.khronos.org/opencl/`, accessed 20/9/2016

[3] `http://www.nvidia.com/object/cuda_home_new.html`, accessed 20/9/2016

3.2.9 Summary

Matching requires well-illuminated images, with sufficient reflectance to dominate detector noise. Unambiguous matching also requires well-textured images. In regions of repetitive or low texture, potential matching pixels may be assigned similar costs and differentiating between them is difficult. This is a limitation of stereo as a passive technique and preliminary results from the challenging 2014 Middlebury images suggest there is still significant progress to be made. Alongside LIDAR, stereo is one of the only 3D imaging methods that works even somewhat reliably outdoors, though dynamic range can become an issue.

Hardware based stereo is maturing as FPGAs and GPUs become ever more powerful although FPGAs are memory-limited for large images. While there are many new algorithms published annually, many authors do not provide easily available source code or executables for their matchers. Users must therefore re-implement algorithms or rely on the few open libraries available, e.g. in OpenCV. Commercially available systems (section 2.1) typically provide proprietary matching software. Stereo is still very much a research topic and will arguably continue to have limited commercial penetration until this issue is addressed.

3.3 Data Fusion

Data fusion is a broad term that has a variety of different meanings depending on context and topic of interest. Most definitions agree that data fusion is a process by which multiple sources of data are combined to produce a product that is in some way improved from that of each individual sensor. The Joint Directors of Laboratories (JDL) Data Fusion Subgroup (Steinberg and Bowman, 2004) suggested the following definition:

> *Data fusion is the process of combining data to refine state estimates and predictions.*

Some of the most highly cited work in data fusion presents a military perspective, (Hall and Llinas, 1997), where fusion occurs on a variety of levels representing how 'close' to the raw sensor data the fusion is performed. The approach and terminology used is rather specific to defence requirements. Wald (1999), in collaboration with the European Association of Remote Sensing Laboratories, recognised this issue and attempted to define data fusion in a general context:

> *Data fusion is a formal framework in which are expressed means and tools for the alliance of data originating from different sources. It aims at obtaining information of greater quality; the exact definition of 'greater quality' will depend upon the application.*

Alongside the definition of the fusion process, definition of the following terms is given:

- *Measurement* - The outputs of a sensor, containing a number of samples. For imaging, this is the image itself, containing pixels. It may also be called a *signal.*

- *Object* - An object is something which is defined by its properties. That is, something that has been classified - e.g. building, road or field. An individual property is a *feature* or *attribute.* A pixel may be an object if it has been classified as one.

- *State Vector* - The combined properties of an object are called a state or feature vector, ideally forming a unique definition.

- *Rules* - These define relationships between objects and their state vectors. The form of a rule may be an equation or mathematical operator. When rules are applied to a set of objects and state vectors, *decisions* are made.

- *Topology* - The arrangement of sensors, how information is exchanged and the cost of acquiring it.

- *Processing* - This addresses how the data should be fused and whether the data is suitable for fusion.

Commonly data fusion is organised into three categories:

1. Pixel - The lowest level, fusion of raw image data. The data may or may not be geometrically corrected.

2. Feature - Fusion of data after some sort of classification has occurred, such as segmentation. Features from different sensors are then fused together.

3. Decision - Output from the sensors are processed individually and the results are then combined using a set of rules.

For this work, data fusion is considered at the pixel level.

3.3.1 Stereo/Range Fusion

In the context of fusing stereo with other data sources, typically LIDAR/ToFCs, there are a number of stages where data fusion may occur:

- Using range data as an *a priori* constraint for the stereo matcher to produce an improved disparity map.
- Fusing the stereo disparity map and the range data *a posteriori*.

The literature published as fusion of stereo and range data can broadly be partitioned into these categories. In the first, each point in the LIDAR/ToFC point cloud can be converted to a disparity (plus uncertainty) in the rectified stereo frame. This disparity range is then used to define the search window of the stereo matcher. The second method is perhaps the simplest and involves combining the output stereo disparity map and the LIDAR point cloud projected into disparity space. This approach is therefore limited to a statistical addition of the two data sets and does not provide any improvement in computation time.

3.3.2 Previous work in Stereo/Range fusion

A review by Beraldin (2004) discussed the fusion of laser scanning and photogrammetry, though applications were mostly limited to visual enhancement of the point cloud. Diebel and Thrun (2006) used a Markov random field to combine a laser scan with a single colour image to produce a high resolution range map by exploiting coincident colour and depth discontinuities. Recent literature has been concentrated around ToFCs as a promising technology for realtime scene reconstruction. As such, most techniques use ToFCs as they can quickly generate a rangemap of a scene in a format that is easily transformed to the stereo disparity space. Alternatively, both outputs are mapped to 3D world coordinates. That said, provided the output from the range sensor and the stereo system may be transformed to the same coordinate system, these methods are agnostic to the type of sensor used.

Similarly, a variety of stereo matching algorithms have been used. The particular choice of algorithm is dependent on the application - for instance realtime data fusion. If speed is not considered, the stereo match results below all have similar deficiencies. If matching is done *a priori* then the insertion method of the range data is algorithm dependent.

3.3.3 A priori (pixel level) fusion

In these methods, data is exchanged between the rangefinder and the stereo camera before disparities are calculated. These ranges are then used as part of the stereo matching cost.

Romero et al. (2004) used a 2D laser scanner and a trinocular camera system to create seed disparity values which are propagated through the image. Each laser point is located in image coordinates. A set of rules determine whether disparity at a pixel is propagated to nearby points based on local image texture and a search for a matching pixel. The initial disparity seed points are taken from the laser scanner, then each of their 8 neighbours. This process is complemented by a stereo matcher which preferentially matches edge regions. The laser seed points are used in homogeneous regions, and the correlation method is used at edges providing improved coverage. Results are only given from the methods presented in the paper, no comparison is given with a standard stereo method or ground truth. However, disparity estimation for textureless regions in the images are improved greatly compared to using the basic matcher alone. This method is the only listed here that uses a 2D laser scan.

Hahne and Alexa (2008) built on work by Kuhnert and Stommel (2006) (detailed below). Their setup is functionally identical, with two consumer cameras and a ToFC (160x120px, 7.5m range). Planar calibration is used for the stereo camera calibration. The authors note that when calibrating the ToFC, the plane with zero phase-shift does not necessarily coincide with the imaging plane and that high noise and inaccurate depth measurement causes calibration results to vary. A graph cut approach was used, with a 400x300x100 (x, y, depth) grid used as the search volume. ToF camera pixels are then matched to this volume using the calibration results and the result is a 3D surface that represents the depth-plus-uncertainty from the camera. This surface is then used as the search volume for the graph cut algorithm, substantially reducing the search space. A consistency term derived from the ToF rangemap is used in the graph cut algorithm and the result takes a few minutes to process. The authors note that while the results are improved, there is difficulty in precisely calibrating the ToF system. Only raw ToF data are shown, but fused results show reduced noise and improved results near depth discontinuities.

Gudmundsson et al. (2008) presented fusion results of a ToFC with stereo imagery at a range of 0.9-4m. Each point in the ToF rangemap is mapped to both left and right stereo cameras and hence a known disparity. This ToF derived disparity map is used to constrain a dynamic programming algorithm on a per-pixel basis. Raw results from ToF and stereo matching are presented for comparison and in this case the fused matcher successfully reconstructs points on a wall where stereo alone fails completely.

Zhu et al. (2011) gave an overview of ToF and stereo fusion using belief propagation. A SwissRanger 176x144px camera was used, up to 7.5m. The authors noted that the ToF sensor exhibits a significant depth bias depending on the reflectance of the target and this is compensated using a per-pixel lookup table (LUT). The bias is independent of target range and is dependent on integration time, which is kept to approximately the stereo shutter speed. After refinement, ToF results are integrated into the matching cost function. Results are presented from raw and refined ToF, stereo, and ground truth based on a structured light approach. When comparing results to truth, the fusion method presented achieves greater than 50% improvement compared to raw stereo. The best results were found when global stereo methods were used in the fusion algorithm, compared to local methods.

Song et al. (2011) used ToF fusion to image plants at a range of 0.4-1.2m. A variety of matchers were compared and it was found that stereo alone did not provide satisfactory results, particularly in terms of discontinuity preservation. Graph cuts were chosen for fusion. The rangemap is used to provide a localised disparity search range; the map is upscaled to the stereo image resolution. If no ToF range is present at a given pixel, a normal full-scale disparity search is performed. A quality metric is provided, based on the sharpness of manually selected edges in the image and the smoothness of calculated leaf surfaces. In general there is a clear improvement when ToF is used to steer the stereo match algorithm.

Fischer et al. (2011) presents a modification of the SGM algorithm, where the pixelwise cost function is adapted to include ToF data that has been reprojected onto the left match image. This approach leverages the speed of local stereo methods and the high accuracy of global methods. A SwissRanger ToF camera is used and the raw data are filtered to remove spurious points. Valid ToF data are included in the aggregated cost function based on an inverted Gaussian weighted by the difference between stereo and ToF disparities. The results demonstrate that over-propagation of ToF disparities to neighbouring pixels causes block artefacts in the fused result as stereo disparities are overridden and a 5x5 neighbourhood is optimal. No ground truth comparison was provided, but fused results overcome the typical failings of stereo in homogeneous regions.

Badino et al. (2011) presented a fusion method using a Velodyne LIDAR as opposed to a ToF system, enabling outdoor applications. The LIDAR data is obtained as a spherical range image which is interpolated sequentially in the horizontal and vertical directions while attempting to preserve discontinuities using an empirical relation. A maximum and minimum range filter is constructed using dilation and erosion morphological operators respectively. These maximum and minimum images are converted to disparity maps and provide a bound on the disparity computed from stereo. DP is used for stereo matching, with the inclusion of a cost that penalises deviations from the minimum/maximum disparity maps. Additionally there is a cost term that encapsulates the confidence of the LIDAR data. The smoothness term is also adjusted to include the expected disparity gradient at each point. Results in outdoor scenes show smoother results, with fewer discontinuities with the prior disparity information. The results are further post processed to remove outliers and the horizontal streaking effects characteristic of DP. No absolute timing is given, but the use of LIDAR data enables between 2-5 times faster computation of the final disparity.

Zhang et al. (2013) used the depth map created by a Microsoft Kinect sensor to aid stereo matching. The data are fused using a belief propagation framework. The stereo system was a COTS 3D compact camera (a JVC GS-TD1B FHD 3D camcorder) which is first calibrated using (Zhang, 2000) and then cross-calibrated with the Kinect using (Zhang and Zhang, 2011). Results were presented from indoor scenes as well as simulations using the Middlebury images for accuracy evaluation. On the simulated data, the number of 'bad' pixels with an disparity error of > 1 px is reduced from 1.27% (stereo alone) and 10.1% (Kinect alone) to 0.15% (fused) on the Venus stereo pair.

All of the above approaches follow a common pattern. First, range data and stereo images are acquired. Then, the two data sets are co-registered, often with some form of interpolation such that each pixel has an associated disparity estimate and confidence. The stereo matcher is run and the disparity estimate is used to constrain the search in some fashion. This may simply limit the disparity search range or it may involve an adjusted cost function.

3.3.4 A Posteriori fusion

These methods fuse data after calculation of range and disparity in an attempt to produce a more accurate, consolidated map. At its simplest this involves adding the disparity map and the range map together with the expectation that the range and disparity maps have complementary coverage.

Kuhnert and Stommel (2006) present one of the first attempts to fuse ToF and stereo, up to an ambiguity range of 7.5m. The authors note it is only suitable for indoor use, and has a 160x64px array. Two CCD cameras with VGA resolution were used for stereo acquisition. Two algorithms were compared: optimised winner-takes-all and simulated annealing, described in Scharstein and Szeliski (2002), producing maps of disparity and 95% confidence across the frame. The ToF data is stored in a minimal and maximal rangemap, corresponding to a range interval for each pixel. This map is upscaled and compared, pixelwise, with the range from the stereo matcher in cartesian 3D coordinates. The fused output contains only pixels that have an overlapping range in the stereo and ToF rangemaps. Results are only presented for one scene and while the ToF camera does provide coverage where the stereo algorithm doesn't, there is significant pixellation in the final images due to the low resolution of the ToF system.

Beder et al. (2007) present an approach based on patchlets (Murray and Little, 2005), rectangular surface elements defined at every pixel in a disparity image. These patchlets encode a best-fit plane at that point, along with an uncertainty measure. The disparity image and depth map are both subsampled to enable data fusion. Combining patchlet data from stereo and ToF data gives the best results, but the speed of the algorithm is not discussed.

Gurram, Lach, Saber, Rhody, and Kerekes (Gurram et al.) applied LIDAR and stereo fusion to building extraction from aerial distances. Stereo data is used to segment buildings and generate planar fits to surfaces. Separately, LIDAR point clouds are also segmented to extract building surfaces. The data are fused to remove errors from solar shadowing (stereo) and poor edge extraction (LIDAR).

3.4 Summary

It is clear that ToF cameras are favoured for fusion with stereo over scanning LIDAR. However, a frequent limitation of ToF data is that it is noisy and difficult to calibrate. Noise is dominated by the shot/Poisson component, but Zhu et al. (2011) showed that depth is also biased by surface reflectance. Due to the range and illumination requirements of current ToF cameras, applications are largely limited to indoor use only.

Almost all autonomous vehicles utilise some kind of stereo vision arrangement, most also include LIDAR and/or RADAR units. Fusion is typically performed at the object level for collision detection purposes. There is very little literature available concerning fusion of scanning LIDAR and stereo for imaging purposes. The results from Beder et al. (2007) and Romero et al. (2004) are encouraging in this respect and demonstrate that outdoor fusion is possible, though there are issues at translucent surfaces.

Results from across the literature are in agreement; using an additional (active) range sensor to compensate for stereo's shortcomings (and vice versa) is viable. Fusion has been studied extensively in the short range (0-10m), up to the limit of ToFCs. In longer range data, airborne LIDAR has been combined with stereo to improve building segmentation (Lee et al., 2008) and to aid with transport network surveying (McCarthy et al., 2007).

Previous efforts in data fusion have focussed on pixel-level, *a priori*, and *a posteriori* fusion using standard stereo match algorithms augmented to include additional range data. Real time operation has only been realised with multiple-beam scanning LIDAR systems and ToFCs.

There has been research into providing artificial texture for stereo imagery, e.g. via the projection of some kind of random pattern. This could also be performed using a LIDAR - an example of this would be taking the stereo imagery during the LIDAR scan, imaging the laser scan pattern. Individual pattern points may be used to provide texture in homogeneous areas while also giving accurate distance measurements in those regions. Alternatively these points may be introduced as part of a region growing algorithm, such as GOTCHA, as a disparity seed point. These ideas are explored in Chapters 6 and 7.

3D Imaging Systems 4

4.1 Overview

This chapter describes the stereo system and LIDAR system used for this research. The components chosen reflect the decision to construct a system that is highly accurate, at the expense of acquisition speed. The specification of each system is given, along with calibrated values of intrinsic and extrinsic parameters. A model is suggested for the LIDAR system, taking into account positioning errors with respect to the mount.

The hardware is controlled using custom software that enables the user to fine-tune scan parameters such as resolution and angular extent. The software enables storage of intermediate images, captured during the scan at every step. Imaging at every point enables the simulation of randomised textures (Chapter 7).

Locating the laser spot is a key element in system cross-calibration (Chapter 5) and data fusion (Chapter 6). Two simple, but effective approaches for direct spot detection are proposed.

Finally, the theoretical performance of the two systems are compared and 'real-world' accuracy measurements are given.

4.2 Stereo System

The stereo camera system used a pair of Imaging Source[1] DMK23UM021 monochrome USB3.0 cameras with a resolution of 1280 ×960 px. The cameras were operated using the manufacturer provided Windows drivers. These were supplied without an infrared blocking filter, allowing testing of light sources such as the pattern projector used by the Kinect (Figure 7.5). The cameras

[1] `http://www.theimagingsource.com/`, accessed 20/9/2016

were each fixed to a rotation platform (Thorlabs[2] XT95P11, Thorlabs RP01) mounted on a section of extruded aluminium rail (Thorlabs XT95). This allowed an adjustable baseline of up to 1m. A baseline of around 0.5m was chosen, providing an acceptable trade-off between stereo image overlap and range resolution. Detailed specifications are shown in Table 4.1.

Parameter	Value
Sensor	ON Semi MT9M021
Sensor Size (mm)	(3.52×2.64)
Pixel Size (μ m)	3.75×3.75
Frame rate (fps)	45 (full resolution)
Shutter	Global
Sensitivity (lux)	0.015
Dynamic Range	8/12 bit
Shutter Speed (s)	1/20000 to 1/4

Table 4.1: Imaging Source DMK23UM021 camera specification.

Two C-mount Computar[3] M0814MP lenses were used with a focal length of 8mm, giving a theoretical single-camera field-of-view of 33.4° by 25.4° per camera. The lens apertures were set to f/8, giving a suitable depth of field, and focused at 1 m. This gave a theoretical depth of field, with a circle of confusion of 1 px, to be 0.67 m - 1.98 m. For a circle of confusion of 2 px, the depth of field expands to 0.5 m - ∞.

Stereo calibration was performed using the `calibrateCamera` and `stereoCalibrate` functions from the OpenCV library[4]. Single camera calibration is performed initially for each camera to provide a robust initial estimation for the stereo calibration step. Further details about this procedure are provided in Chapter 5. 14 stereo pairs were used for calibration and the results are summarised in Table 4.2.

4.2.1 Calibration stability

The system was re-calibrated several times over the course of the research period. There was no evidence to suggest that, without physical interaction, the calibration degraded over periods of several months. This is expected, as the cameras were rigidly mounted on quality optomechanical components in a laboratory with no significant temperature variation or vibration. Unfortunately OpenCV does not (yet) provide uncertainties on individual estimated parameters, only a single

[2] `https://www.thorlabs.com/`, accessed 20/9/2016

[3] `http://computar.com/`, accessed 20/9/2016

[4] `http://www.docs.opencv.org/modules/calib3d/doc/calib3d.html`, accessed 20/9/2016

Parameter	Left Camera	Right Camera
Focal length (mm)	8.38	8.37
Camera centre (px)	(657.70, 479.62)	(657.87, 489.06)
Distortion coefficients, $(k1, k2, k3, p1, p2)$	$(-8.17 \cdot 10^{-2}, -0.14, 1.79 \cdot 10^{-4},$ $2.65 \cdot 10^{-4}, 3.29)$	$(-9.57 \cdot 10^{-2}, 0.30, -8.02 \cdot 10^{-5},$ $1.47 \cdot 10^{-3}, -0.18)$
Horizontal FOV (deg)	31.98	31.99
Vertical FOV (deg)	24,22	24.34
Position (m)	(0, 0, 0)	$(0.46, 4.9 \cdot 10^{-3}, 0.064)$
Rotation (deg)	(0, 0, 0)	$(-0.68, -12.11, 1.28)$
Reprojection Error (px)	0.10	0.10

Table 4.2: Camera calibration results using OpenCV's calibration routine.

reprojection error. Although parameter variations were not explicitly measured, verification was performed by checking vertical disparity values in stereo matching results. Any changes to the optical system would result in the calculated rectification would no longer be correct and vertical disparities not close to zero[5]. This conveniently decouples calibration assessment from any particular calibration process or target and can be quickly performed online after a stereo pair has been matched. Alternatively a known (static) target could be used, for instance Habib et al. (2005) used a wall covered with markers.

Whenever the system was moved, or adjustments made to lens focus or aperture, a re-calibration was performed. Calibration results reported in this thesis, for example in this chapter and in Chapter 5, represent the most recent calibration prior to the measurements being acquired. Were the system mounted on a vehicle or in a more dynamic environment, it is expected that calibration would either be performed more frequently or adjusted using, for example, bundle adjustment with self calibration (Fraser, 1997; Chow and Lichti, Chow and Lichti).

4.3 LIDAR System

In order to evaluate different strategies for data fusion, a single point LIDAR was mounted on a gimbal mount. Unlike conventional scanning LIDAR systems which capture over a full hemisphere, this method enabled the scan pattern to be precisely controlled and limited to just the field of view of the cameras. Both LIDAR and mount were independently controlled via serial connection to a computer. The Dimetix[6] FLS-C 10 is an accurate LIDAR unit designed for industrial positioning applications. It has a slow acquisition rate in its most accurate mode, with a maximum

[5] A typical threshold would be 1px

[6] `http://www.dimetix.com/`, accessed 20/9/2016

speed of 20Hz, however it offers higher accuracy and repeatability at a much lower cost than scanning LIDAR systems. Analogue output at up to 200Hz is possible, but with degraded accuracy. Detailed specifications are given in Table 4.3 and a dimensional drawing is shown in Figure 4.1. The unit was connected via RS-232 connected to a local PC.

Parameter	Value
Resolution (mm)	0.1
Accuracy (mm)	± 1
Repeatability (mm)	±0.3
Measurement range (m)	0.05 to 65
Laser wavelength (nm)	620-690
Laser beam Divergence (deg)	0.01 by 0.03
Radiant Power (mW)	0.95
Pulse Duration (s)	0.45×10^{-9}
Measurement time (s)	0.05 to 4

Table 4.3: Dimetix FLS-C 10 LIDAR specification. Measurement range specified by Dimetix on natural surfaces.

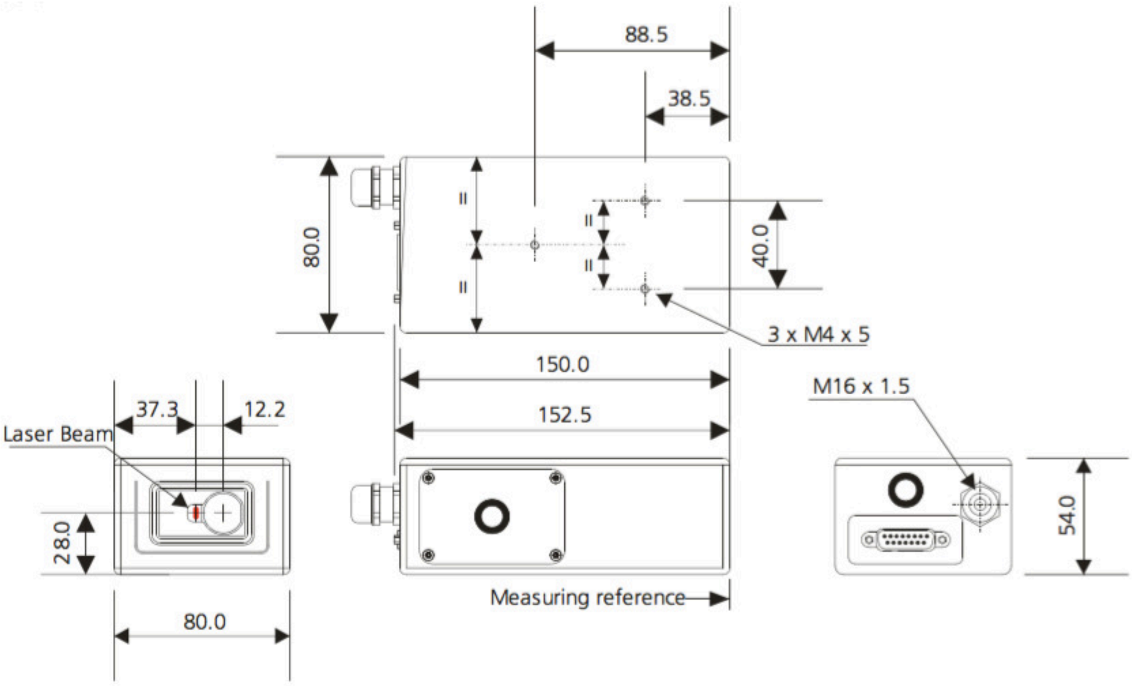


Figure 4.1: FLS-C 10 LIDAR unit dimensional drawing, courtesy of Dimetix

The LIDAR was mounted on a Newmark Systems Inc.[7] GM-12E 2-axis gimbal mount. This mount has excellent positional accuracy and optical homing switches, which allow repeatable measurements with respect to the camera coordinate system. Specifications are given in Table 4.4. A

[7] http://www.newmarksystems.com/gimbal-mounts/, accessed 20/9/2016

dimensional drawing is shown in Figure 4.2. The mount was controlled using an NSC-G2 (Newmark Systems Inc.) motion controller, connected via RS-232 to a local PC.

Parameter	Value
Azimuth Range (deg)	±90
Altitude Range (deg)	±90
Resolution (deg)	2×10^{-4}
Accuracy (deg)	0.004
Repeatability (deg)	6×10^{-4}
Maximum Speed (deg/s)	20

Table 4.4: Newmark GM-12E gimbal mount specification.

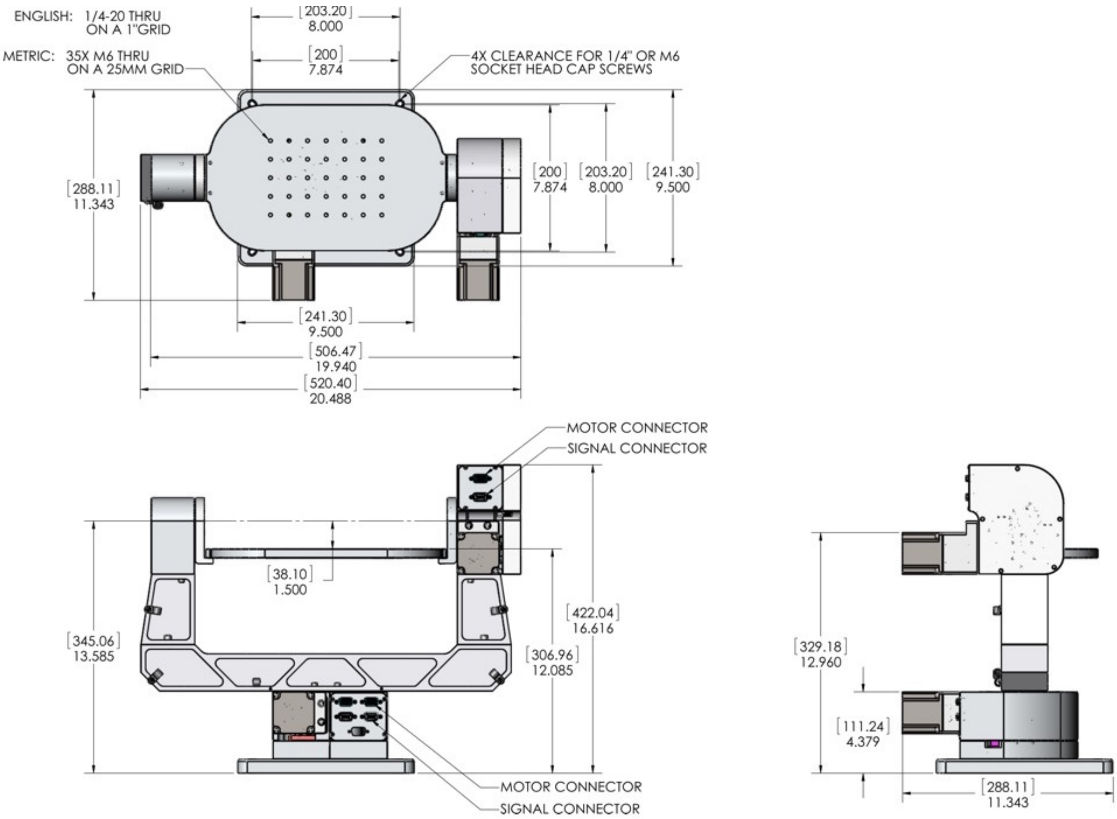


Figure 4.2: GM-12E gimbal mount dimensional drawing, used with permission of Newmark Systems Inc.

Both the LIDAR and stereo bar were fixed using bolts to a thick sheet of MDF. The LIDAR was positioned in the centre of the stereo bar, but with a vertical separation due to the height of the mount itself. The complete system is shown in Figure 4.3.

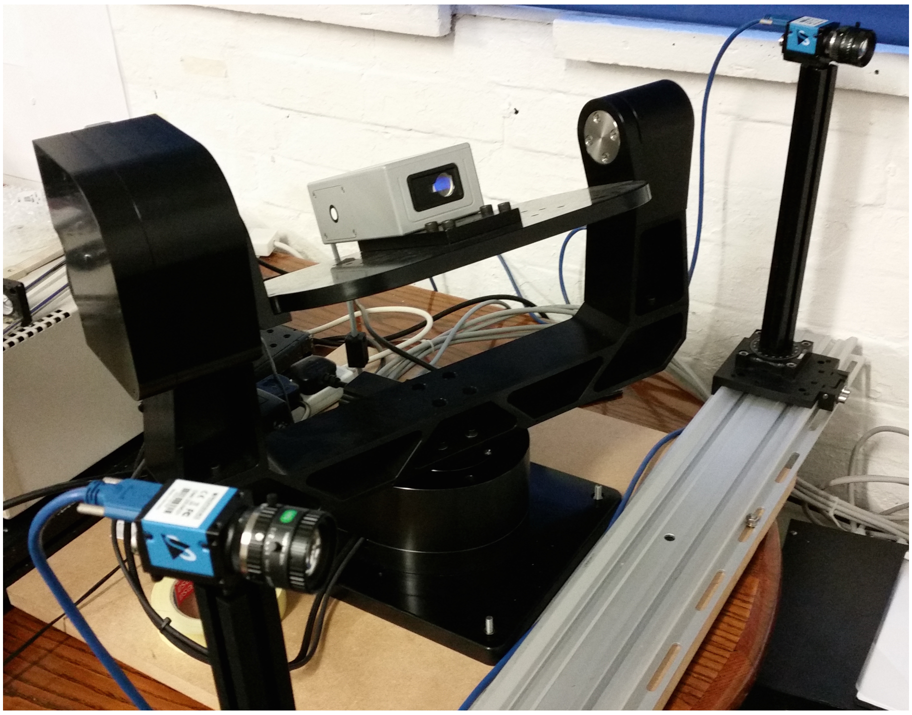


Figure 4.3: Stereo/LIDAR system.

4.3.1 LIDAR Model

The LIDAR coordinate system is defined to be the same handedness as the stereo system. The z-axis is positive into the scene, the y-axis is positive downwards (Figure 4.4). Each measurement consists of a measured range r_m, altitude φ and azimuth θ. In this coordinate system, altitude is the angle between the y-axis and xz-plane and azimuth is the angle between the x- and z-axes. The FLS-C 10 reports distances measured from the front surface of the unit.

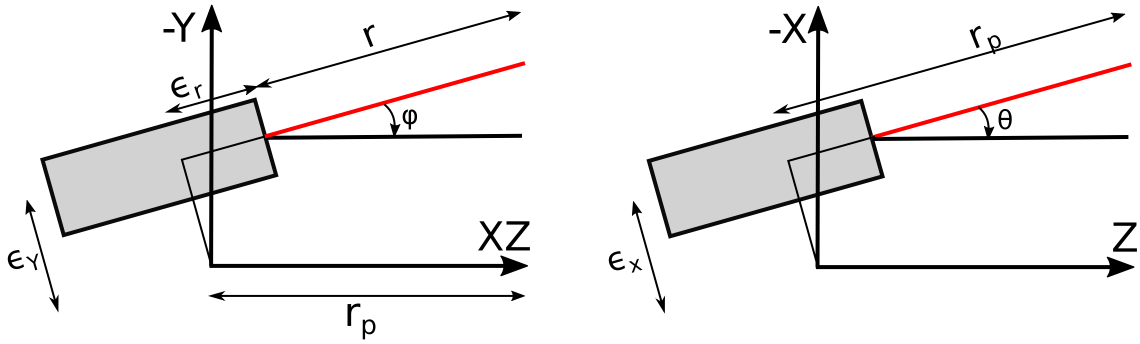


Figure 4.4: LIDAR system geometry shown from side (left) and above (right). Note that angles reported by the GM-12E mount are positive upwards and rightwards.

Additional corrections are required as the LIDAR is not perfectly centred on the mount. Also the LIDAR receiver aperture is offset laterally from the centre of the unit, as shown in Figure 4.1.

There are 3 possible translational offsets $(\epsilon_x, \epsilon_y, \epsilon_z)$ from the centre of rotation, 3 angular offsets $(\epsilon_{rx}, \epsilon_{ry}, \epsilon_{rz})$ from the coordinate axes and 3 systematic errors in reported distance/direction $(\epsilon_r, \epsilon_\theta, \epsilon_\varphi)$. Several of these parameters are degenerate: an error in reported azimuth or altitude is equivalent to a rotation offset of the LIDAR from the coordinate axes; a systematic error in reported distance is equivalent to a translational offset in the z-direction; rotation about the z-axis does not affect the measurement once horizontal/vertical translation is corrected. The LIDAR system geometry may be modelled using 8 parameters $(r_m, \theta_m, \varphi_m, \epsilon_x, \epsilon_y, \epsilon_r, \epsilon_\theta, \epsilon_\varphi)$ and the coordinate conversion from polar to Cartesian is given by:

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\varphi & -\sin\varphi \\ 0 & \sin\varphi & \cos\varphi \end{bmatrix} \begin{bmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{bmatrix} \begin{bmatrix} \epsilon_x \\ \epsilon_y \\ r_m + \epsilon_r \end{bmatrix} \tag{4.1}$$

where $\varphi = -\varphi_m + \epsilon_\varphi$ and $\theta = -\theta_m + \epsilon_\theta$.

For this work the LIDAR coordinate system is defined to be the world coordinate system. Stereo coordinates are therefore related to LIDAR coordinates by a rotation, **R** and translation **t** as shown in Figure 4.5.

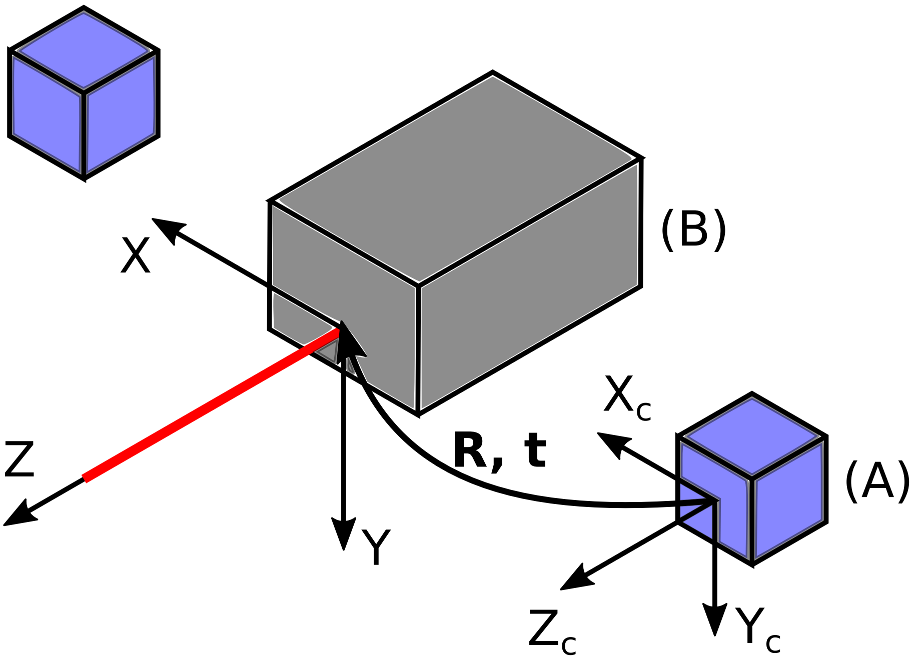


FIGURE 4.5: Relationship between the camera (A) coordinate system, (X_c, Y_c, Z_c) and the LIDAR (B) coordinate system (X, Y, Z).

An example calibration result is given in Table 4.5. Including the offsets in the rotation angles were found to cause excessive instability in the fitting process and so were set to zero.

Parameter	**Value**
Translation (x, y, z) (m)	(-0.230,-0.254,-0.212)
Rotation about (x, y, z) (deg)	(0.01, 0.26, 0.08)
ϵ_x (mm)	8.4
ϵ_y (mm)	1.1
ϵ_r (mm)	4.1
ϵ_ϕ (deg)	0
ϵ_θ (deg)	0

Table 4.5: LIDAR system calibration parameters. Rotation and translation is given with respect to the left stereo camera as the LIDAR system is at the origin by definition.

4.3.2 LIDAR Spot Location

A key component of the calibration and data fusion methods presented in this research is the determination of the location of the LIDAR spot in the stereo images. Experimentally, shutter speeds of below 1/500 s are sufficient to suppress indoor background light even in the presence of strong sunlight and shadowing, leaving only the laser spot visible.

The simplest method to determine the spot location is a maximum filter combined with a threshold, but this tends to suffer from aliasing as only integer pixel locations are given. Using the pixel with the maximum intensity as an initial location estimate, the laser spot location may be refined using a 2D Gaussian fit. An example fit is shown in Figure 4.6. The LIDAR emits a rectangular beam pattern which is approximately gaussian in both axes. The Gaussian fit included a rotation parameter to compensate for distortion when the beam is incident on surfaces that are not orthogonal to the beam direction. This operation is computationally expensive, but takes only around 40 ms on an Intel 2.4 GHz Core 2 Duo processor which is sufficient for real-time usage with the hardware used.

Detecting occlusions in stereo pairs is possible by calculating the y-disparity between spots detected in each rectified image and discarding those which differ by more than a pixel.

4.4 Scanning Procedure and Software

Software was developed in C++ using the Qt [8] Graphical User Interface (GUI) framework to scan the LIDAR whilst simultaneously acquiring stereo imagery. Users are presented with 'live' images

[8] https://www.qt.io/, accessed 20/9/2016

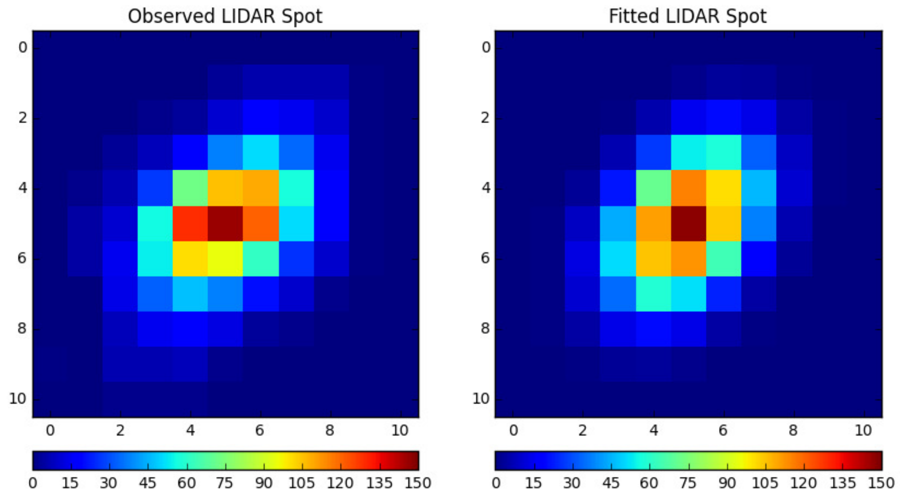
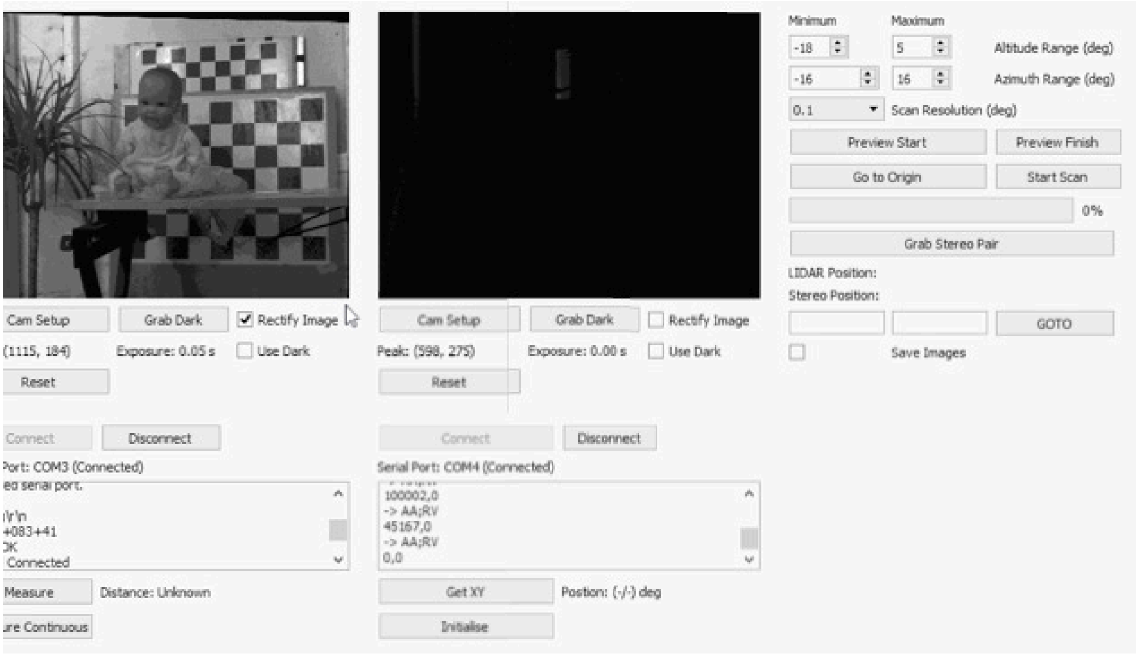


Figure 4.6: 2D Gaussian fit to a typical image of a LIDAR spot. Note that the LIDAR beam profile is not radially symmetric and the image is also distorted by the angle of the surface with respect to the beam. Exposure time of 1/1000 s at f/8.

from both cameras and can query both the mount and LIDAR for current position information. Control of each camera, the mount and the LIDAR is delegated to a separate thread for efficiency. A screenshot of the GUI is shown in Figure 4.7.



☑ Rectify Image

(1115, 184) Exposure: 0.05 s ☐ Use Dark

Port: COM3 (Connected)

```
ed serial port.

\r\n
+083+41
OK
 Connected
```

Distance: Unknown

☐ Rectify Image

Peak: (598, 275) Exposure: 0.00 s ☐ Use Dark

Serial Port: COM4 (Connected)

```
100002,0
-> AA;RV
45167,0
-> AA;RV
0,0
```

Postion: (-/-) deg

Minimum	Maximum	
-18	5	Altitude Range (deg)
-16	16	Azimuth Range (deg)

0.1 Scan Resolution (deg)

0%

LIDAR Position:

Stereo Position:

☐ Save Images

Figure 4.7: Stereo/LIDAR scan control software.

Before a scan is acquired, the user may optionally initialise the gimbal mount. The position of the mount is stored in volatile memory in the mount controller; the orientation of the mount is assumed to be ($\varphi = 0, \theta = 0$) when power is applied. The initialisation procedure locates the optical home switches in both axes and defines their locations to be the zero points of each axis.

Exposure settings for both cameras are user-defined since the brightness of the intensity tends to confuse the automatic exposure algorithm. Prior to a scan, well-exposed ambient light imagery is acquired for stereo matching. The sensor gain was set to zero in all images to reduce noise.

Acquired images are optionally rectified in real-time using the OpenCV library's image warping functions and a user-supplied calibration file. Optionally a 'dark' frame may be acquired and subtracted if there are many sources of specular reflection in the scene that might be mis-detected as the laser spot. This method works well provided that the ambient illumination does not change during a scan and is therefore limited to indoor operation. Users have the option of saving images at each point in a scan for the purpose of more detailed analysis, such as more accurate laser peak detection. Saved images are also required for simulating texture projection. The workflow for the scanning process is shown in Figure 4.8

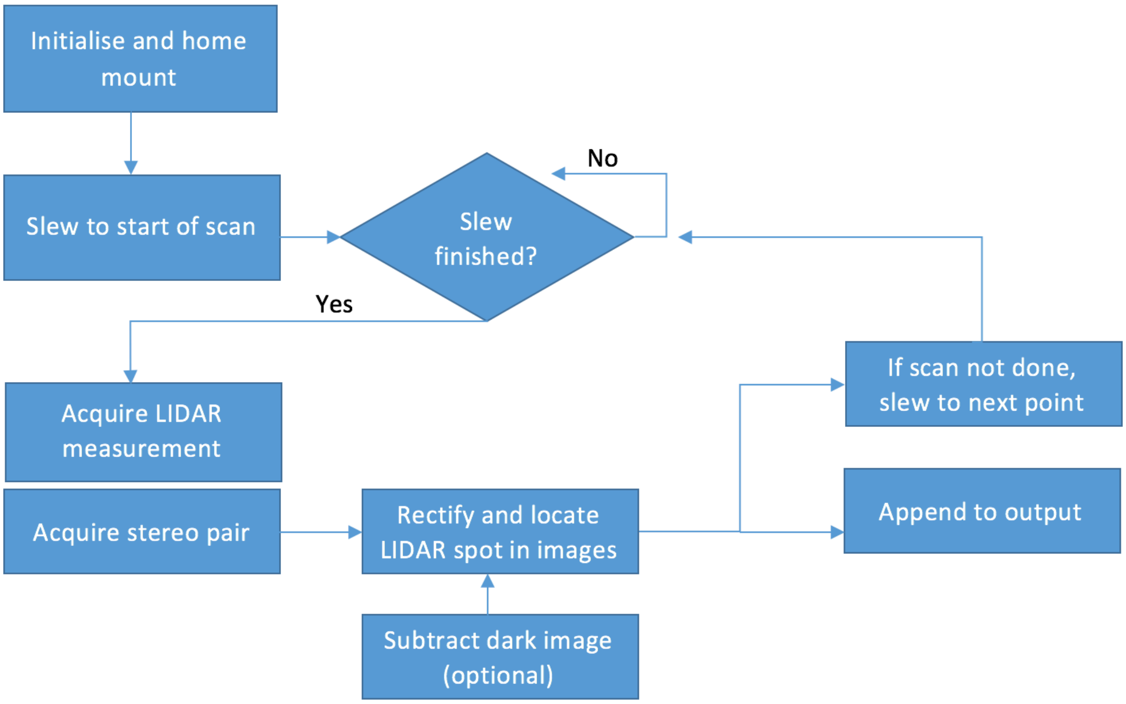

Initialise and home mount
Slew to start of scan
Slew finished?
No
Yes
Acquire LIDAR measurement
Acquire stereo pair
Rectify and locate LIDAR spot in images
Subtract dark image (optional)
If scan not done, slew to next point
Append to output

Figure 4.8: Flow diagram showing the steps performed in the scanning procedure.

Laser spot location using a thresholded maximum filter is continually performed upon image acquisition. If the images are rectified, then the filter is applied to the rectified images. Accurate spot detection using a Gaussian peak detector is currently performed on the saved images. The maximum pixel location is stored even if the laser is occluded so post-processing is necessary to label points where no laser spot was detected (Section 4.3.2).

If the LIDAR reports an error, the range for that point is stored as zero. Since the minimum measurable distance is 0.05m, these points are easily filtered post-scan. The cause of the error is not stored, as the LIDAR does not provide detailed information, but possible reasons are to insufficient SNR, detector saturation or ambiguous range near a depth boundary.

As a high resolution scan (>100k points) may take several hours to acquire at 20 Hz, the angular limits of each scan are user-specified. Similarly the altitude/azimuth step between points is user-specified as necessary.

The scan is performed in a raster fashion. At each step in the scan, the LIDAR range, mount position and laser spot location in each image are stored in an eXtensible Markup Language (XML) file along with the current UNIX timestamp. At each point, the software blocks first until the mount has signalled that it has stopped and then until each measurement is taken in parallel. The same timestamp is also applied to both stereo images that are saved, allowing easy synchronisation with ranging data. Point values are stored in RAM and then saved to the hard disk upon each altitude step, i.e. every row. The resulting XML file, combined with any saved images, comprises the data output for a scan.

The storage requirements for scan data alone is modest, since each point requires 8 floating point numbers to be stored. A high resolution scan therefore is on the order of megabytes. Storing all intermediate images requires significantly more disk space with a worst case of 1.3MB per image. The use of image formats that support lossless compression such as Portable Network Graphic (PNG) however even with low exposure images, a significant amount of intensity variation is present and typically only a reduction in file size of 50% is possible. A very low intensity threshold of 5 enables much more efficient compression and a file size reduction of two orders of magnitude while retaining enough information to accurately locate the laser spot. An example of this is shown in Figure 4.9; in this case the raw image has a compressed file size of 494 kB, the thresholded image has a file size of 4 kB. This is comparable to the operating system's file allocation size, the minimum allowable size for a file.

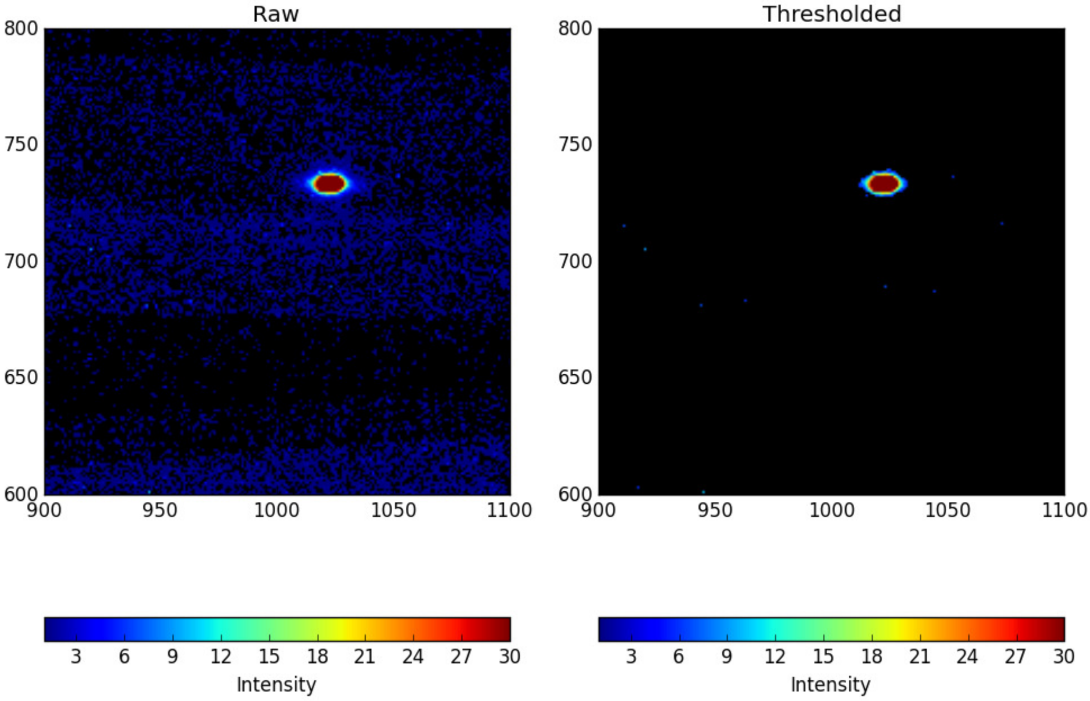


Figure 4.9: Crop from an intermediate scan image, showing the location of the LIDAR laser spot. Left: raw image, colour-mapped with a maximum intensity of 30 for emphasis. Right: threshold of 5 applied.

4.5 Theoretical System Performance

4.5.1 Stereo

Stereo depth resolution is arbitrarily small for sub-pixel matching, provided the matching algorithm returns the disparities in a floating point representation. In reality stereo performance is limited by two factors: the correlation accuracy ϵ_d and the calibration accuracy ϵ_c (for a fixed camera geometry). Calibration accuracy determines how well characterised the optics of the camera are and limits the spatial resolution of the 3D measurements as:

$$\Delta x = \epsilon_c \frac{z}{f} \tag{4.2}$$

$$\Delta y = \epsilon_c \frac{z}{f} \tag{4.3}$$

Stereo depth resolution is given by Equation 2.19. The focal length and baseline are known from the camera calibration (Table 4.2), but the disparity or correlation accuracy is dependent on the stereo matching algorithm used.

Measuring correlation error directly is difficult as correlation performance is dependent on image texture. Often only an heuristic estimate is possible, defined for a particular set of experimental data. Direct measurement is possible if ground truth is available, but structured light methods typically introduce similar uncertainties that scale quadratically with distance. An alternative method is to measure the depth accuracy on a cooperative target as a function of distance and use that to indirectly measure ϵ_d via Equation 2.19.

Stereo imaging a well-textured planar surface provides a straightforward way of estimating depth accuracy. This is a common method used to evaluate 3D imaging performance, and standards such as VDI/VDE 2634 Pt. 2 [9] which (although not internationally accepted) provide guidelines for measuring known geometric objects such as spheres and planes. Ahmadabadian et al. (2013), for example, used calibration spheres with known radii and a calibration cube which allowed both planarity and perpendicularity measurements. Beraldin (2009) highlighted the need for a set of common terminology and standards for uncertainty measurements. Experimental results (measuring flatness) were shown using lapped planar target with varying surface reflectances. More complex targets may also include features such as stepped geometries or closely spaced blocks, typically manufactured using precision CNC machining (Hess et al., 2014).

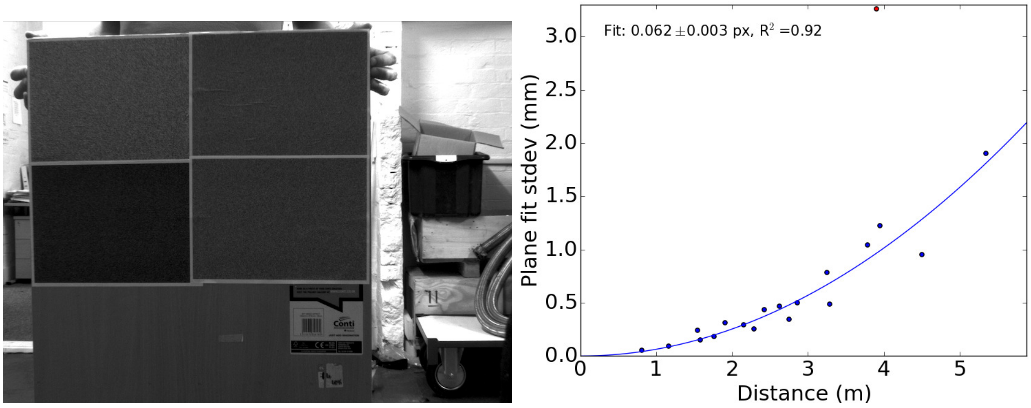
In the case of planar fitting, a plane is fitted to the reconstructed points and the standard deviation of the point-plane distances used as the depth error, ΔZ. This method was used to estimate ϵ_d for the system used in this research.

A random dot pattern was printed onto white paper and fixed to a wooden board. Rectified stereo images of the pattern were acquired at distances in a range of 0-5.3 m. The images were matched using Gotcha with default settings: maximum eigenvalue 100, patch size 12 and 8-neighbour matching (justification is given in (Shin and Muller, 2012)).

The data were modelled using Equation 2.19, with b and f given by the stereo calibration, results are shown in Figure 4.10. A best fit of $\epsilon_d = (0.062\pm0.003)$ px was calculated with an R^2 value of 0.92 showing a good model fit. It should be noted that this result was obtained under ideal conditions, i.e. matching with strong texture and good illumination, and represents an optimistic

[9] VDI/VDE 2634/Part2, 2002. Optical 3-D Measuring Systems – Optical Systems based on Area Scanning.

estimate of the disparity measurement error.



(a) (b)

Figure 4.10: (a) Planar random dot patterns (b) Stereo disparity measurement error obtained by imaging pattern at various distances. With a focal length of 8 mm and a baseline of 0.46 m, sub-mm accuracy is possible up to nearly 4m. The characteristic quadratic error curve is clearly visible.

Given the calibrated field of view, the projected size δs of a pixel in either camera is $\delta s = Z \cdot \tan(0.025°)$.

4.5.2 LIDAR

Both the LIDAR and mount have significantly higher resolutions than their specified accuracy. It is assumed that for the range of distances of interest, under 10 m, the LIDAR signal will have a high enough SNR that the nominal data sheet accuracy is attainable. In cases where the LIDAR does not return a distance, it is more likely that this is due to very low SNR or an ambiguous range. Unlike single-view stereo, it is possible to take repeated LIDAR measurements to obtain a variance at each point.

All LIDAR measurements are an average of the distances covered by the laser footprint. The laser size at the exit aperture is not specified by Dimetix, but footprint sizes at various distances (5m, 10m, and 30m) are provided. By making an exponential fit to these footprint sizes, an estimate of the laser spot size in pixels as a function of distance can be made. Figure 4.11 shows that the expected laser spot radius is less than 10 px for a typical scan. The actual imaged spot size is generally larger than this due to the intensity of the reflected light, even at short exposures.

While the laser footprint is several pixels wide at close range, the LIDAR can be stepped at a pitch that is smaller than the extent of a pixel. Additionally, the majority of the return signal

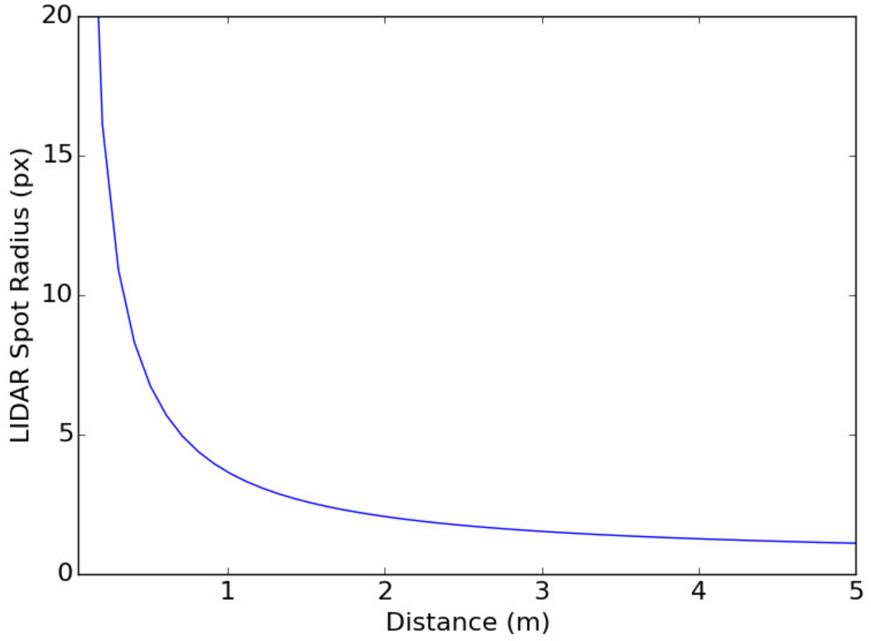


Figure 4.11: LIDAR laser spot size as a function of distance. At larger distances, for this arrangement > 5m, LIDAR spatial resolution begins to exceed that of stereo.

is concentrated into an area that may only span one or two pixels. Producing a LIDAR map of a scene with a higher resolution than that of the cameras is possible using this system, but the result would be smoother at close range than an ideal stereo reconstruction.

Errors on measurements made with the LIDAR system are likely to be dominated by the accuracy of the LIDAR range (± 1 mm). The mount accuracy is sufficiently high, $0.004°$ that even at 5m the expected x-y deviation is only 0.35 mm which is comparable to the LIDAR repeatability of 0.3 mm.

4.6 Summary

This chapter provided an overview of the 3D imaging system used in the research. Sensors and actuators were selected with accuracy as a first priority. Although the acquisition speed of the system is poor compared to a commercial scanning LIDAR, the expected accuracy is superior and greater control over individual measurements is possible.

Multithreaded scanning software has been developed to acquire data with user-specified resolution (altitude/azimuth stepping) and angular extent. The software enables a scan of a scene that includes intermediate images of the LIDAR spot at each ranged point. Thresholding followed by lossless compression is used to limit the storage requirements for each scan.

Laser spot detection has been demonstrated using two techniques: naive threshold and maximum filtering, and a least squares Gaussian fit. In both cases, short exposure times are used to suppress background noise. For scenes with significant specular features, dark frame subtraction is suggested as an effective method for robust spot location determination.

Both systems were calibrated with the results presented here. Standard calibration techniques were used to provide a reference camera calibration. A geometric model for the LIDAR was suggested and calibration results given. The expected accuracy and resolution of each system has been evaluated, including an estimation of the correlation accuracy of Gotcha on cooperative targets. This estimation suggests that sub-mm accuracy is possible using the stereo system up to around 2m.

Stereo Camera and LIDAR Inter-Calibration 5

5.1 Overview

Camera calibration is a necessary and important step in stereo 3D reconstruction. This chapter discusses methods for camera calibration, stereo rig calibration and stereo-LIDAR cross calibration. One large drawback with most calibration algorithms is the need to use calibration targets and, usually, human interaction.

Two common single-camera calibration methods are discussed: the direct linear transform (DLT) (Abdel-Aziz and Karara, 2015)[1] and Zhang's multiple-view algorithm (Zhang, 2000). Next, an overview of prior close range camera-LIDAR cross-calibration is given including accuracy estimates.

Finally a novel calibration approach using a visible scanning LIDAR is proposed. It is shown that unlike previous methods using LIDAR, this technique is capable of recovering both intrinsic and extrinsic camera parameters without an explicit calibration target. Results and accuracy measurements are given from real-world scenes and compared to ground truth. Extrinsic using this method is comparable to the state-of-the-art in terms of point cloud fitting error. It is shown subpixel reprojection errors are achievable using the LIDAR for intrinsic parameter determination.

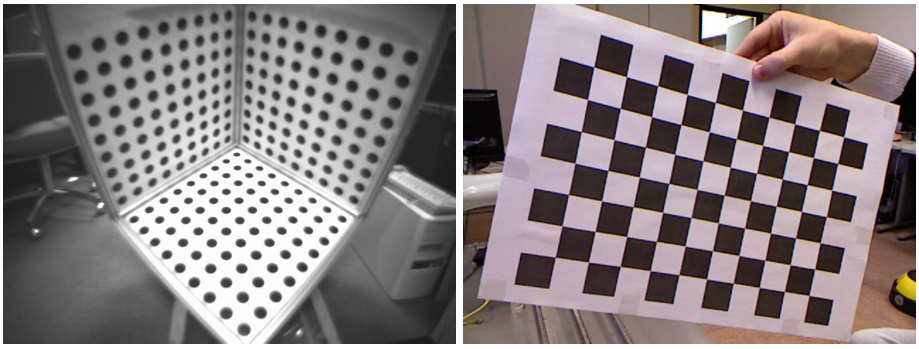
5.2 Camera calibration

Camera calibration is the procedure which determines the camera intrinsic and extrinsic parameters described in Section 2.3.1. For accurate reconstruction, a camera or stereo rig should be recalibrated every time a mechanical adjustment is made. This includes focus adjustments on most lenses as rotation of an optical element may change the location of the projection centre. Many

[1] A reprint of the classic paper (Abdel-Aziz and Karara, 1971) in a modern format. The content is unchanged.

algorithms exist in both the photogrammetry (Duane, 1971; Faig, 1975) and computer vision communities (Tsai, 1987; Heikkila et al., 1997; Zhang, 2000).

For accurate calibration, a set of image-object correspondences $\mathbf{x}_i \leftrightarrow \mathbf{X}_i$ must be known. The image points, $\mathbf{x}_i$ are the locations of calibration features within the image. The object points, $\mathbf{X}_i$ are the locations of these calibration features in the world. The image-object correspondences are usually generated by imaging a cooperative target with known geometry. Methods involving 1D (Zhang, 2004), 2D (planar) and 3D targets exist. 2D targets are usually boards with a printed geometric pattern such as a checkerboard or grid of circles (Heikkila, 2000). Figure 5.3 shows setups for 3D and planar 2D calibrations.



(a) (b)

Figure 5.1: Calibration algorithms require known correspondences between the world and the image. Most methods use either (a) 3D object points (single-view) or (b) 2D planar calibration targets like checkerboards. The 2D case usually involves calculating the homography that maps the world plane to the sensor plane; several views are required.

3D targets may be formed of multiple planes, objects with a known shape or control points fixed to the scene. If a 2D target is used, normally multiple views of the target are required to avoid degeneracy. If a 3D target is used then calibration can be performed using a single image. While there are reconstruction methods which can recover stereo geometry up to an unknown scale factor without requiring calibration points with known world coordinates (Hartley and Zisserman, 2003), they will not be discussed here.

Calibration accuracy is usually measured using the geometric reprojection error. Suppose $\hat{\mathbf{x}}_i$ are the image locations of object points $\mathbf{X}_i$ under a particular camera model and $\mathbf{x}_i$ are the actual image locations. The geometric error d is then:

$$d(\hat{\mathbf{x}}_\mathbf{i}, \mathbf{x}_\mathbf{i}) = ||\hat{\mathbf{x}}_\mathbf{i} - \mathbf{x}_\mathbf{i}|| \tag{5.1}$$

The sum over these distance errors is commonly used as the objective function when performing an iterative optimisation of the camera parameters. RMS errors of 0.1-0.2px are easily achievable using calibration toolkits like OpenCV. However taking the raw error figure does not take into account possible mistakes such as a degenerate solution (e.g. if the scene is entirely coplanar) or if the calibration points are not distributed well enough to accurately model lens distortion.

5.2.1 The direct linear transform

If accurate image-object correspondences are known, then the direct linear transform (DLT) is a simple and effective means to determine the camera parameters (Sutherland, 1974; Abdel-Aziz and Karara, 2015). The basic DLT algorithm solves a series of (overdetermined) linear equations of the form $\mathbf{Ap} = \mathbf{0}$ where $\mathbf{p}$ is a 12×1 vector representing the 3×4 camera projection matrix, $\mathbf{A}$ is a $2n \times 12$ matrix with the ith element:

$$\mathbf{A}_i = \begin{bmatrix} 0 & 0 & 0 & w_i & -w_iX_i & -w_iY_i & -w_iZ_i & -w_i & x_iX_i & x_iY_i & x_iZ_i & w_ix_i \\ w_iX_i & w_iY_i & w_iZ_i & w_i & 0_i & 0_i & 0_i & w_i & y_iX_i & y_iY_i & y_iZ_i & w_iy_i \end{bmatrix} \tag{5.2}$$

where $(x, y, w)_i$ are the ith homogenous image coordinates ($w = 1$) and $(X, Y, Z)_i$ are the ith corresponding object or world coordinates. At least 6 correspondences are required. The linear solution for $\mathbf{p}$ is found using singular value decomposition (SVD). If desired, the elements of $\mathbf{p}$ can then be optimised using an iterative algorithm like Levenberg-Marquadt (LM) (Marquardt, 1963) with the geometric (reprojection) error used as the objective function and the linear solution as an initial guess. Lens distortion is not considered in the initial, linear, stage of the DLT, but can be included in the non-linear optimisation stage (Hatze, 1988). The projection matrix can be decomposed into the intrinsic and extrinsic parameter matrices via RQ-factorisation (Press, 2007).

The main difficulty of this approach is determining the image-object correspondences and in taking accurate independent 3D measurements of the object points.

5.2.2 Calibration using planar targets

A more practical calibration method is to image a single or multi-planar target with known geometry and a printed calibration pattern (Tsai, 1987; Zhang, 2000). The calibration pattern is a grid of squares (checkerboard) or circles. The object points are then measured to high accuracy in 2D (nominally the Z coordinate is set to zero). Similarly corner finding in images is highly accurate if the images are well exposed, yielding image points with sub-pixel accuracy (< 0.05 px) Krüger and Wöhler (2011). Neither method makes any assumption of orientation of the pattern board, although Tsai recommends that the board be at least 30°with respect to the sensor plane. An example of a planar calibration target is shown in Figure 5.3.

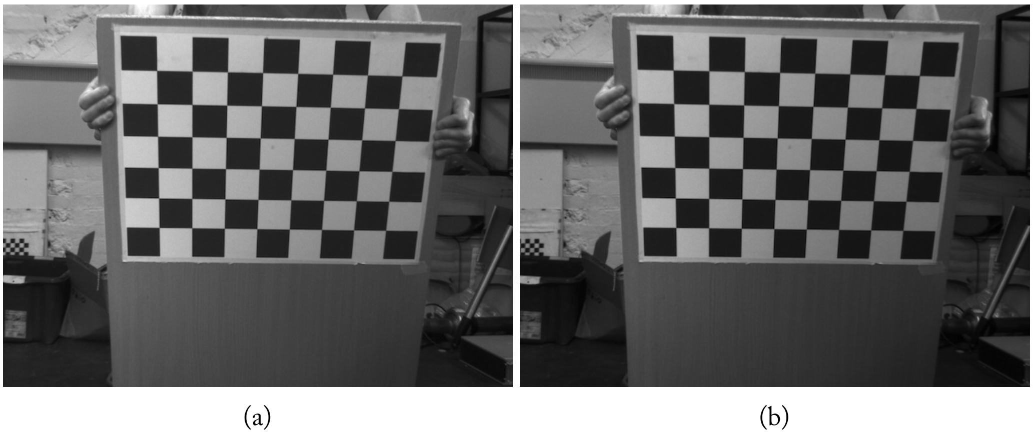


Figure 5.2: (a) Planar calibration target for use with (Zhang, 2000). Multiple views with the target in different orientations are required for calibration. (b) With lens distortion removed. This particular lens has very low radial distortion, the difference is most obvious on the top of the calibration pattern.

Tsai's method differed from contemporary algorithms which typically involved either nonlinear optimisation in a large parameter space or involved only linear equations without lens distortion. The proposed algorithm is a two-stage process combining both classes of algorithm. In the first step, calibration parameters are linearly approximated (similarly to the DLT). In the second, the parameters are optimised non-linearly, but only for one or two iterations. The calibration targets are planar and 3D objects points can be generated by translating the plane on a z-stage.

Zhang's method, or a derivative of it, is used in several computer vision toolkits including OpenCV and Matlab [2]. The popularity of this method stems from its user-friendliness and its high accuracy. The calibration target is a checkerboard pattern that is printed and fixed to a planar surface. A COTS laser printer is sufficient. The user then acquires several images of the target from various angles and at various distances to the camera. Prior knowledge of the pose of the target in each view is not necessary.

[2] http://www.vision.caltech.edu/bouguetj/calib_doc/

For each view of the pattern, a homography is calculated between the planar target and its image. By considering two separate constraints that these homographies place on the intrinsic parameters, a set of linear equations is formed and solved. Then, including distortion, the parameters are optimised using LM. Radial and tangential correction is possible. The algorithm is able to determine the pose of the calibration target in each view.

Accuracy measurements by Sun and Cooperstock (2005) found that Zhang's method outperforms Tsai in terms of accuracy by a factor of three. A minimum of two views is required, but in practice to accurately model radial distortion more views with calibration points distributed across the image is preferred. Zhang recommended between 8-15 images, with at minimum of five.

5.2.3 Stereo camera calibration

Stereo calibration seeks to obtain the epipolar geometry that describes the stereo rig. This amounts to computing the fundamental matrix, F, commonly performed using the 8-point algorithm (Longuet-Higgins, 1981). This algorithm assumes that image correspondences between the two views are already known, as is usually the case when calibrating a stereo rig with a common target. The procedure for estimating **F** is straightforward.

First, recall that for corresponding image points $\mathbf{x}'_i \leftrightarrow \mathbf{x}_i$ between the two views, $\mathbf{x}'^T_i\mathbf{F}\mathbf{x}_i = 0$, defining **F** (equation 2.16). If the image points are expressed as homogenous coordinates, then it is possible to expand out the equation for each correspondence:

$$xx'F_{11} + xy'F_{21} + xF_{31} + yx'F_{12} + yy'F_{22} + yF_{32} + x'F_{13} + y'F_{23} + F_{33} = 0 \tag{5.3}$$

or equivalently:

$$\begin{bmatrix} xx' & xy' & x & yx' & yy' & y & x' & y' & 1 \end{bmatrix}\mathbf{f} = 0 \tag{5.4}$$

where **f** is a 9×1 vector representing **F**. These may be stacked together to form a linear set of equations $\mathbf{Af} = 0$. Subject to the constraint that $||\mathbf{f}|| = 1$, at least 8 correspondences are required. The algorithm is sensitive to the relative magnitudes of the input correspondences unless normalisation is applied first (Hartley, 1997). In order to accurately model lens distortion, these points should be distributed across the sensor since distortion is usually much more significant near image edges.

If the calibration method uses a 3D target then the determination of the position of one camera with respect to the other is trivial since the pose of each camera relative to the common coordinate system is known. Otherwise, the perspective-n-point (PNP) algorithm (Quan and Lan, 1999) can be used to determine the relative position of each camera with respect to each view of the calibration target. This requires synchronised stereo image pairs of the target which must be fully visible in both views.

Once the calibration is complete, the stereo images are first undistorted and then rectified as described in Section 2.3.3.

5.3 Stereo-LIDAR extrinsic calibration

Fusing the output from different range sensors requires extrinsic calibration between them. There has been significant prior research into Stereo-LIDAR extrinsic calibration, largely motivated by robotic or autonomous vehicles. Cross calibration is also required for generating point clouds with real colour, derived from visible imagery.

Stereo-LIDAR calibration algorithms can be classified into methods which either image the laser beam directly or scan a calibration target with known geometry.

5.3.1 Direct LIDAR-Stereo correspondences

If the laser beam is visible to the camera, then it is possible to identify the location of the beam from the image providing 2D-3D point correspondences. Prior work has largely focused on 2D scanning LIDAR since these scanners project a fixed stripe in the image.

Kwak et al. (2011) proposed using a V-shaped cardboard target and a LIDAR scanner with an IR beam. Line and point features are manually extracted from the camera image and many images (at least 50) are required with the target in varying locations and orientations. A second IR camera is used to locate the laser beam on the target. Reprojection errors of under 5px were reported. Yang et al. (2012) proposed a similar method using the corner of a room. This method yielded state-of-the-art reprojection errors of 0.5-2px using only 15 scan-image pairs. IR pass filters were necessary to compensate for the relatively poor QE of the cameras in the IR compared to the ambient visible light. Both methods require planar features in order to achieve good calibration.

5.3.2 Calibration using a planar target

In most cases, the LIDAR beam is not visible to the cameras, as most cameras are fitted with IR blocking filters. A common solution is to introduce a calibration target that is simultaneously imaged and scanned with the LIDAR. By locating the target in the image and the LIDAR scan, it is possible to determine the relative pose of each system.

Zhang and Pless (2004) were the first to propose using a planar checkerboard target. Starting with a calibrated camera, the target is imaged and scanned with the LIDAR. The camera pose with respect to the board is calculated using Zhang's camera calibration algorithm. The position of the board is detected in the LIDAR scan and used to calculate the rotation and translation between the camera and LIDAR. With 20 views, the translation error between the camera and the LIDAR was reported to be 3mm. Vasconcelos et al. (2012) built on this work and proposed an improved, minimal solution which requires only 3 views rather than 5. Mirzaei et al. (2012) also considered a minimal solution which included intrinsic calibration of a 3D LIDAR (a Velodyne HDL-64E).

Naroditsky et al. (2011) used a calibration plane with a stripe printed on it. The stripe was identified in the LIDAR scan by its reflectivity. Reprojection errors were not given, but the translation error between the camera and LIDAR was given to be 1.9 mm in 'real-world' data.

Gong et al. (2013) used an arbitrary trihedral target which does not need to be orthogonal. Planes found in the scene can be used for calibration, such as wall-floor intersections.

Park et al. (2014) use a polygonal planar board. The entire board does not need to be scanned, only as much as is necessary to reconstruct its vertices. The vertices are then used as calibration points. Reprojection errors of 4px are reported using more than 5 views of the target.

All the methods above assume that the camera is pre-calibrated, although if the calibration routine involves imaging a series of checkerboards in various orientations then Zhang's algorithm could be used. User interaction is typically required if multiple views of the target are needed and in some cases manual feature identification is required.

5.4 Calibration using a visible scanning LIDAR

The proposed new calibration method uses a visible LIDAR and exploits the fact that the LIDAR spot is visible in the image. This therefore falls into the first class of algorithms. By scanning the LIDAR across the scene and imaging the laser spot at each step in the scan, a list of accurate object-image correspondences is generated. From these correspondences, the stereo-LIDAR system can be calibrated using the DLT with least-squares refinement. Figure 5.3 shows the LIDAR-camera arrangement.

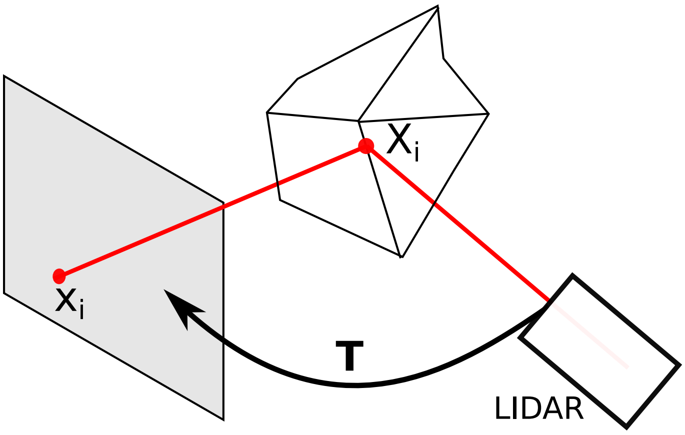


Figure 5.3: Using a visible beam LIDAR, it is possible to generate accurate object points, X_i, that are visible in the camera, x_i. The camera world coordinates and LIDAR world coordinates are related by a rotation and translation, shown here as $\mathbf{T}$. No explicit calibration target is required.

Although this method was developed in the context of a fused stereo/LIDAR system, it also presents the possibility of accurate, automatic camera intrinsic calibration. The method has several advantages:

1. As each LIDAR spot is imaged individually, it is straightforward to uniquely map object and image points.
2. By using a visible beam LIDAR, there is no need for an expensive additional IR camera or filters and simultaneous visible stereo imagery can also be acquired.
3. Extrinsic calibration with a pre-calibrated stereo rig reduces to a rigid body transform. If a single camera is used, then PNP may be used to obtain camera pose.
4. Using a steerable LIDAR, it is possible to generate calibration points throughout the image. This allows for combined intrinsic and extrinsic camera calibration without the need for a

calibration target.

5. The only required user interaction is choosing the angular range over which to scan and selecting an appropriate threshold for the LIDAR spot detection.

In the following sections, first LIDAR intrinsic/extrinsic calibration is detailed. This is important as the accuracy of the object points is dependent on the performance of the LIDAR. This method is also used for extrinsic calibration with a stereo rig. Next, a method for fully calibrating a combined stereo-LIDAR system, including camera intrinsic parameters, is given. Real-world data is used for demonstration and compared to a reference camera calibration. Two errors are considered: reprojection error of the LIDAR points into the camera images and the fitting error of stereo 3D points with the LIDAR scan.

5.4.1 LIDAR intrinsic calibration

Determination of the LIDAR intrinsic parameters $(\epsilon_x, \epsilon_y, \epsilon_r, \epsilon_\theta, \epsilon_\varphi)$ requires a set of 3D calibration points. As the laser beam is visible, these can be produced by using a well calibrated stereo rig. The LIDAR spot is located in the rectified stereo imagery and triangulated in the stereo coordinate system. Spot location was described in more detail in section 4.3.2. A 2D Gaussian peak is fitted to the image of the LIDAR spot and its centre is taken as the location of the object point in the image.

Imaging the LIDAR spot yields a set of coordinates in stereo world coordinates X_{Si} and LIDAR world coordinates X_{Li}. Corresponding coordinate pairs are ideally related by a rigid body transformation:

$$X_{Li} = \mathbf{R}X_{Si} + \mathbf{T} + \eta \tag{5.5}$$

where **R** is a 3D rotation and **t** a 3D translation that maps the stereo point cloud onto the LIDAR point cloud. This transformation is the extrinsic geometry between the LIDAR system and the stereo rig. In reality there will be a distance error, η, which accounts for errors in the stereo calibration, LIDAR spot location and the LIDAR signal-to-noise ratio. The effect of non-zero intrinsic parameters is to introduce an additional systematic error between the two coordinates following the transformation:

$$\sigma_i = X_{Li} - (\mathbf{R}X_{Si} + \mathbf{T}) + \eta \tag{5.6}$$

It is assumed that $\sigma_i >> \eta$ for poorly chosen intrinsic parameters. Using this distance error, determination of the optimal intrinsic parameters is expressed as a minimisation problem:

$$\underset{(\mathbf{R},T,\epsilon_x,\epsilon_y,\epsilon_r,\epsilon_\theta,\epsilon_\varphi)}{\arg\min} \sum_i |(X_{Li} - (\mathbf{R}X_{Si} + T))| = \underset{(\mathbf{R},T,\epsilon_x,\epsilon_y,\epsilon_r,\epsilon_\theta,\epsilon_\varphi)}{\arg\min} \sum_i \sigma_i \tag{5.7}$$

The rotation and translation are initially estimated as an affine transform using the OpenCV function `estimateAffine3D` which uses RANSAC (Fischler and Bolles, 1981) for robustness. The other parameters are set to zero initially. Non-linear optimisation of the parameters is then performed using a method such as LM. Although imaging the LIDAR spot tends to produce few false correspondences, RANSAC is used to select a sample from the point clouds as even a single blunder point is enough to produce a poor fit.

5.4.2 Extrinsic calibration with a stereo rig

Once the LIDAR intrinsic parameters are known, recalibration with respect to a new (calibrated) stereo rig is straightforward. In this situation, the aim is to compute only the rotation and translation that maps the LIDAR point cloud into the stereo coordinate system or vice versa. It is therefore assumed that both the LIDAR and stereo point clouds are distortion free, but exhibit normally distributed errors.

As before, the transformation is estimated as an affine transform using OpenCV and then refined using LM with the objective function:

$$\underset{(\mathbf{R},T)}{\arg\min} \sum_i |(X_{Li} - (\mathbf{R}X_{Si} + \mathbf{T}))| \tag{5.8}$$

5.4.3 Extrinsic calibration of a single camera

In this case, the aim is to compute the pose of a single calibrated camera with respect to the LIDAR point cloud. This method could also be applied to a stereo rig, considering each camera separately. With a single camera, this type of calibration is normally used to overlay an image onto a point cloud. The solution in this case is to use the PNP algorithm. Point cloud colouring can then performed by mapping LIDAR points back into the image to determine which pixels should be sampled.

5.4.4 Intrinsic calibration of an unknown camera

Finally, there is the general case of an uncalibrated camera where both intrinsic and extrinsic parameters are to be determined. As with the previous methods it is assumed that a list of LIDAR-image coordinate correspondences are known. The algorithm consists of two stages. First, the projection matrix, P, is approximated using the DLT. Then, lens distortion is included in the model and the parameters are optimised using LM. Only the first two radial distortion terms are considered (k_1, k_2) and tangential distortion is not included. Additionally the skew, s is fixed to be zero and the aspect ratio is fixed to be one ($f_x = f_y = f$). In order to improve the robustness of the calibration, RANSAC is used to select calibration points.

The minimisation function is:

$$\underset{(\mathbf{R},\mathbf{t},f,c_x,c_y,k_1,k_2)}{\arg\min} \sum_i |\mathbf{x}_i - M(\mathbf{R}X_{Li} + \mathbf{T})| \,. \tag{5.9}$$

where $\mathbf{x}_i$ are the imaged points and M represents the proposed camera model including lens distortion. The radial distortion parameters are initially set to zero with the remaining parameters taken from the estimated projection matrix, P.

5.5 Calibration Results

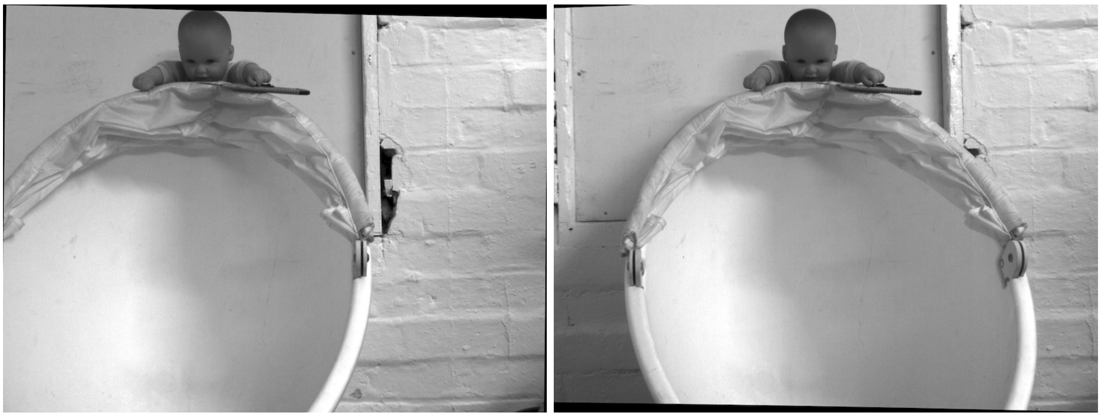
5.5.1 Experimental Setup

An initial stereo camera calibration using OpenCV's `calibrateCamera` and `stereoCalibrate` functions was performed to act as a benchmark and to produce point clouds from the LIDAR spot location data. The LIDAR was also calibrated with the results shown in Table 4.5.

For demonstration, two scenes were chosen. The only constraining factor was that the scene must not be entirely coplanar; fortunately in real-world environments this is rare. Ambient illumination was provided by both fluorescent lighting and diffuse sunlight through a window. The exposure time for both cameras was set to 1/750 s. This was also necessary to prevent saturation of the image of the laser. A threshold intensity of 75 was chosen for spot detection. Any image with a maximum intensity of < 75 was considered occluded. Similarly any measured points where the LIDAR returned an error were discarded.

The extent of the LIDAR scan was such that measurements were acquired across the field of view of both left and right cameras. This was necessary to capture any radial distortion presented by the lenses. The resolution of the scans varied between 0.1 degrees and 0.25 degrees. Results from two scenes are given.

The first demonstration scene, ‘wall’, (Figure 5.4) was a wall with a plastic chair placed in front of it. The chair provides points that are not coplanar and most surfaces in the scene are white providing good reflectivity for the laser beam.



(a)

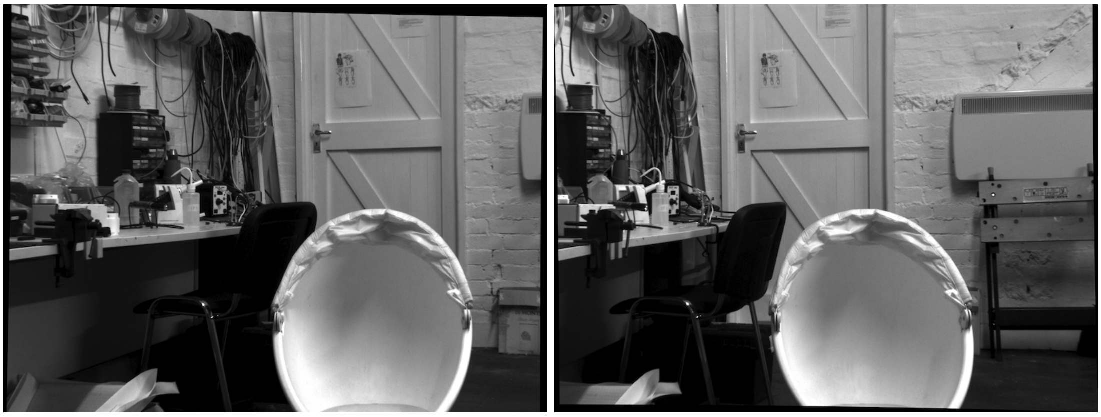
(b)

Figure 5.4: Wall calibration scene, rectified images.

The second demonstration scene, ‘unstructured’, (Figure 5.5) was an unstructured workshop environment containing surfaces with a wide range of reflectances. The scene also provided calibration points at distances between 2-5m in order to demonstrate that the algorithm works at different distance scales.

5.5.2 Calibration Results and Discussion

For clarity, the calibration routine is described in detail using the first scene (‘wall’) as an example and results from the other scene follow. Two main accuracy metrics were considered. The first is point cloud matching accuracy, which is defined as the RMS distance error between the LIDAR point cloud and the point cloud generated by projecting and transforming the stereo image points. Note this is the error that is minimised in equation 5.8. This metric is useful for data fusion since it describes how well the two datasets can be combined. Note that this does require a stereo



(a) (b)

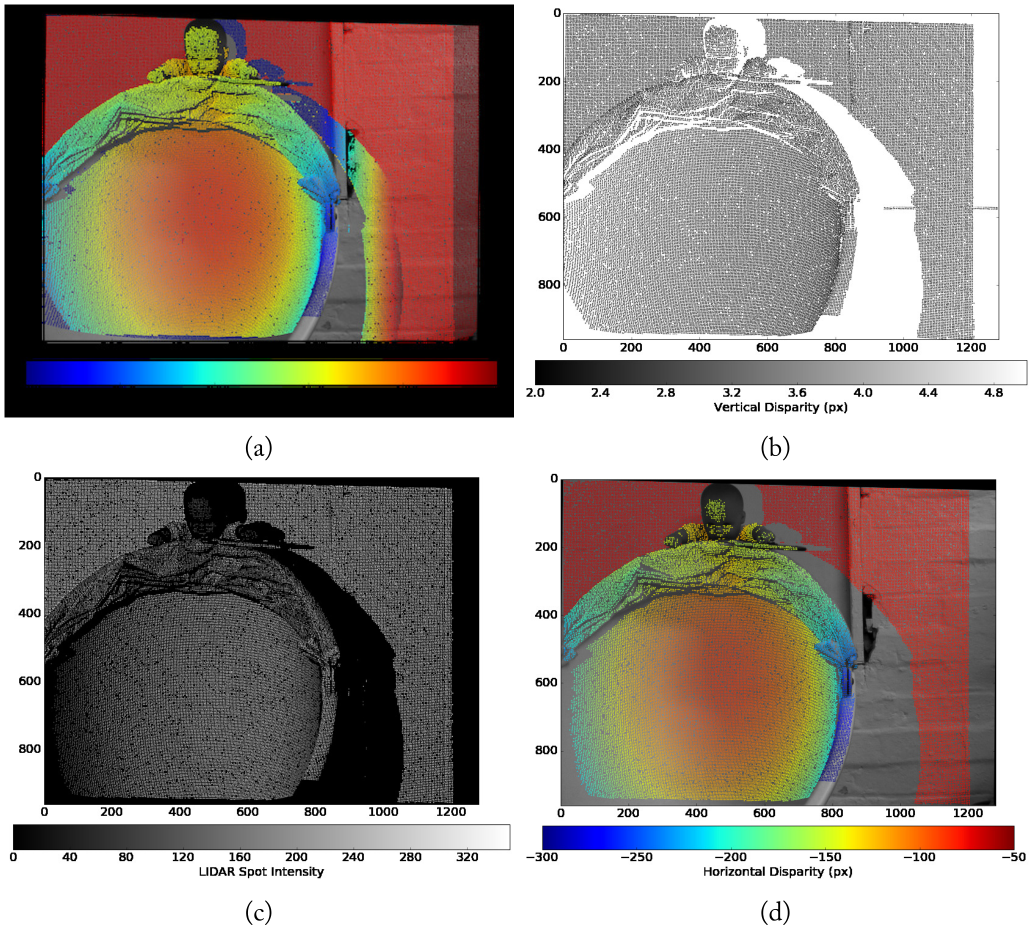
Figure 5.5: Unstructured calibration scene, rectified images.

calibration of the cameras. The second is reprojection error which is a more common metric for comparing camera calibration algorithms (equation 5.9). The reprojection error expresses how accurately a 3D point can be mapped onto a particular pixel.

5.5.2.1 Extrinsic Calibration

Extrinsic calibration requires corresponding points in the stereo and LIDAR world coordinate frames. It was necessary to filter out LIDAR points which were occluded in each image. Since the laser was detected in each view, it was possible to compute horizontal and vertical disparity values for each LIDAR point. Figure 5.6 shows the calculated disparity maps for 'wall'.

A straightforward way of detecting occlusions is thresholding the vertical disparity image; if the laser is visible in both images the disparity should be < 1 px. In the horizontal disparity map (Figure 5.6a), the overlap between the left and right views is seen as a vertical discontinuity at around $x = 1200$ px. There are several blunder points in the centre of the chair and the region of the wall that has been occluded by the chair is also visible. In the vertical disparity map (Figure 5.6b), thresholded at 2 px, the occluded regions of the image are well segmented. Some blunder points are seen around $y = 580$ px where the error is coincidentally zero. By also considering only points with a laser spot intensity (Figure 5.6c) above the threshold (75), the filtered disparity map is obtained (Figure 5.6d). Note that there are still blunder points in the intensity information near depth discontinuities. These are caused by the LIDAR beam grazing the edge of an object. Thus, both vertical disparity and LIDAR spot intensity are required for robust filtering. Although some points were incorrectly discarded, such as those on the doll's head, for calibration it is more



(a) (b) (c) (d)

Figure 5.6: Calibration scene 'wall', left viewpoint (a) horizontal disparity map, (b) vertical disparity map (thresholded), (c) LIDAR spot intensity, (d) filtered horizontal disparity map.

important that all false positive correspondences are removed.

The filtered disparity map was used to generate a point cloud, using the intrinsic parameters obtained from Zhang's method. The unaligned stereo and LIDAR point clouds are shown in Figure 5.7.

As the stereo and LIDAR system are largely co-aligned, the point clouds are mostly separated by a translation. The extrinsic parameters were computed using the OpenCV function `estimateAffine3D` with a RANSAC confidence threshold of 0.999. The parameters are shown in Table 5.1.

Extrinsic Parameter	**Value**
Rotation (x, y, z) (deg)	$(-1.45, 0.17, -0.15)$
Translation (x, y, z) (mm)	$(-235.26, -237.41, -206.91)$

Table 5.1: Calibrated LIDAR-stereo extrinsic parameters. The transformation maps the LIDAR points into the stereo coordinate system (referenced to the left camera).

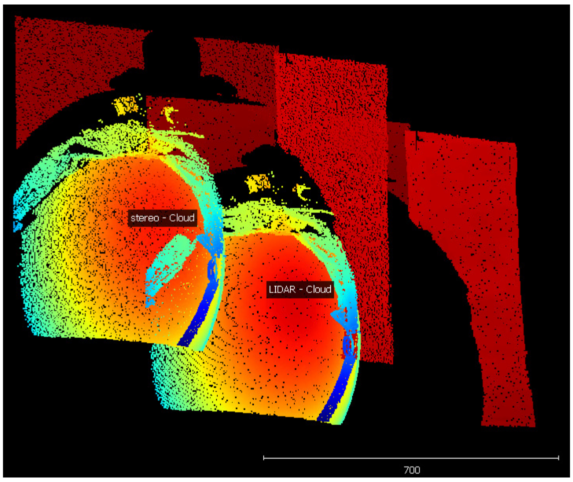
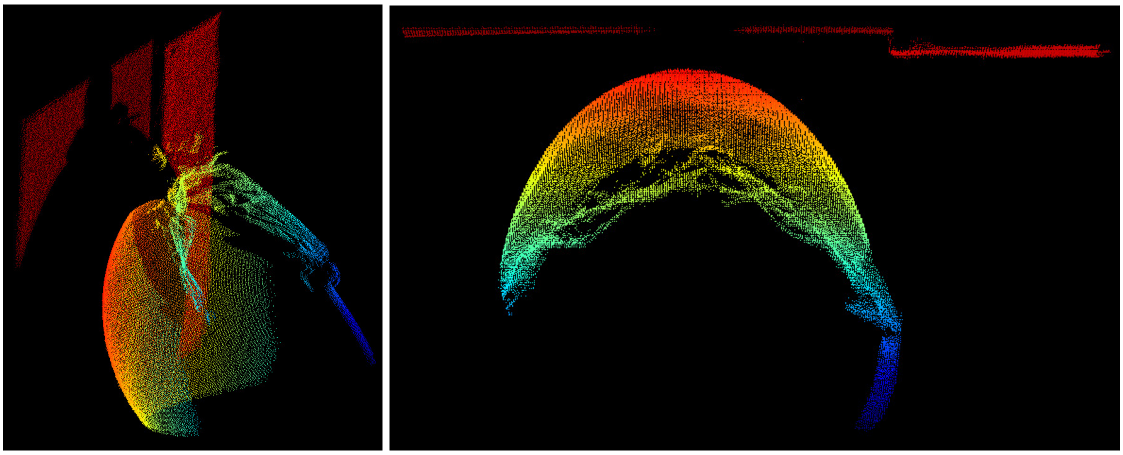


Figure 5.7: Unaligned point clouds from calibration 'wall'. Point cloud colouring denotes distance from the sensor. The stereo point cloud shows more noise on the rear wall (red planar surface).

Although the ground truth location of the LIDAR with respect to the cameras was unknown, the retrieved parameters were sensible: the LIDAR was close to being co-aligned with the optical axis of the left camera and was positioned above, to the right of, and behind the camera. The system baseline was 0.46 m and the LIDAR was positioned approximately in the middle of the stereo system. The aligned point clouds are shown in Figure 5.8.



(a) (b)

Figure 5.8: Aligned point cloud from calibration scene 1. (a) side view (b) top view showing good alignment of the rear wall. The apparently poorer alignment in the top left is a view artefact - the wall is slightly concave towards the ground.

The RMS fitting error in each axis was found to be $(e_x, e_y, e_z) = (-0.13 \text{ mm}, 0.21 \text{ mm}, -0.27 \text{ mm})$ mm.

5.5.2.2 Intrinsic Calibration

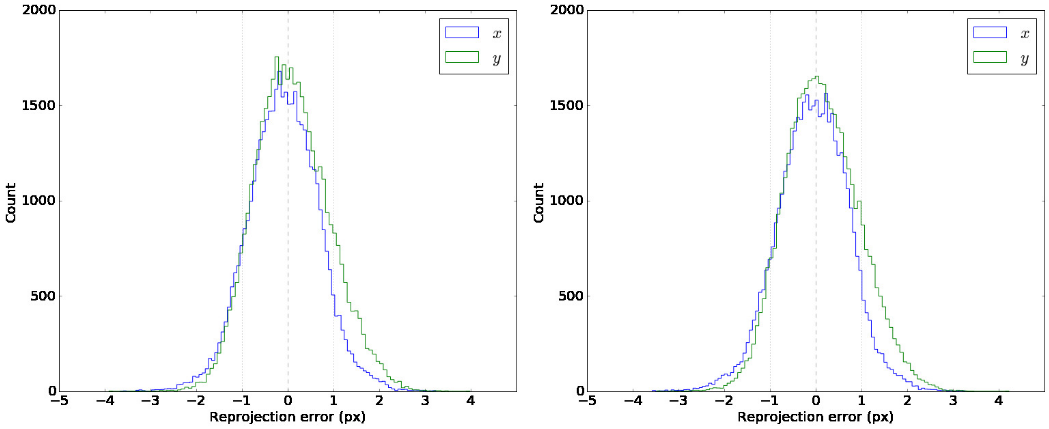
For intrinsic calibration, the vertical disparity is unknown so the calibration relies on robust LIDAR spot detection. The number of outliers is expected to be small, such that most random samples of calibration points only contain inliers (and therefore that only a few iterations of RANSAC are required to reach a good calibration). The algorithm was applied to un-rectified imagery representing a real-world calibration scenario. A random sample of 20 points was used in each iteration of RANSAC to fit a prospective camera model. The RMS reprojection error for a correspondence to be classified as an inlier was set to 1 px. Results are shown in Table 5.2. Paramters are reported here in pixels, rather than metric units

Parameter	Left Truth	Left Fit	Right Truth	Right Fit
Focal length (mm)	8.31	8.32	8.29	8.32
Camera centre (px)	(662.5, 479.8)	(648.74, 486.1)	(657.7, 476.1)	(650.0, 490.7)
k1	-0.09	-0.13	-0.08	-0.14
k2	-0.42	-0.54	-0.49	-0.57
RMS error (px)	0.10	0.61	0.10	0.62
Pos. (x,y,z) (mm)	(0,0,0)	(0,0,0)	(443.3, 6.26, 50.0)	(463.8, 2.36, -0.34)
Rot. (x,y,z) (deg)	(0,0,0)	(0,0,0)	(-0.99, -10.6, 1.39)	(-0.17, -11.8, 1.19)

Table 5.2: Unrectified camera intrinsic and extrinsic parameters derived from LIDAR correspondences in the 'wall' scene, compared to truth values derived from a planar checkerboard target.

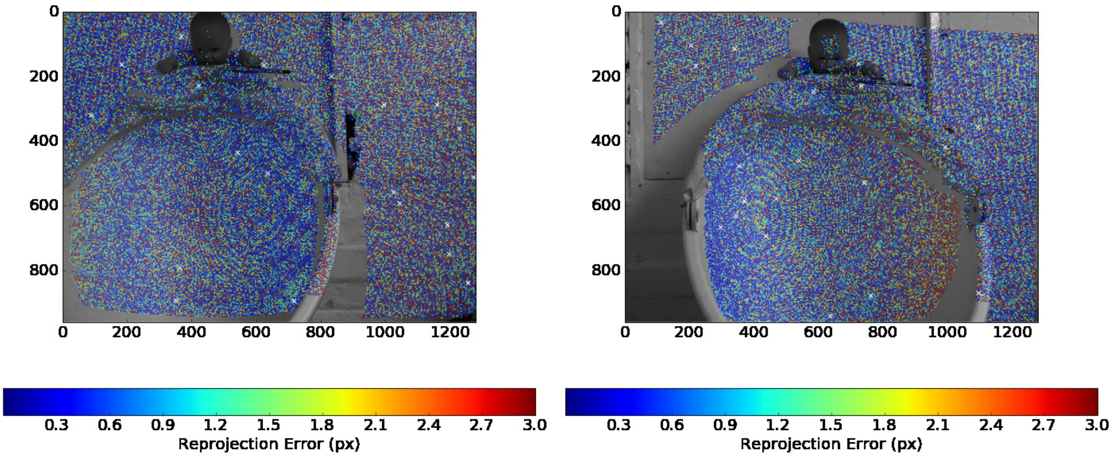
In general the calculated intrinsic values were close to the ground truth values. The camera centre was correctly estimated to be near the centre of the image sensor and the dominant radial distortion coefficient, $k1$, was also close to the truth value. Differences between the truth value and the fit can be explained by the fact that any valid calibration is not unique. A difference in camera centre may be compensated for by shifts in other parameters, for instance. The RMS reprojection error using the LIDAR was several times larger than when using a checkerboard, but was still sub-pixel. Reasons for this increased error are suggested in the following discussion. The truth position and rotation values were derived from the OpenCV stereo calibration routine (prior to rectification). Figure 5.9 shows the distribution of reprojection error magnitudes in both image axes.

The error distribution for both cameras appeared to be Gaussian, centred around zero in both axes, indicating a good model fit. As Figure 5.10 shows, the reprojection error is uniform across the field of view. There are some regions of higher error, particularly on the right edge of the chair. This is likely due to poor detection of the LIDAR spot on the plastic surface.



(a) (b)

Figure 5.9: Reprojection error distribution for 'wall' calibration scene (a) left camera (b) right camera.

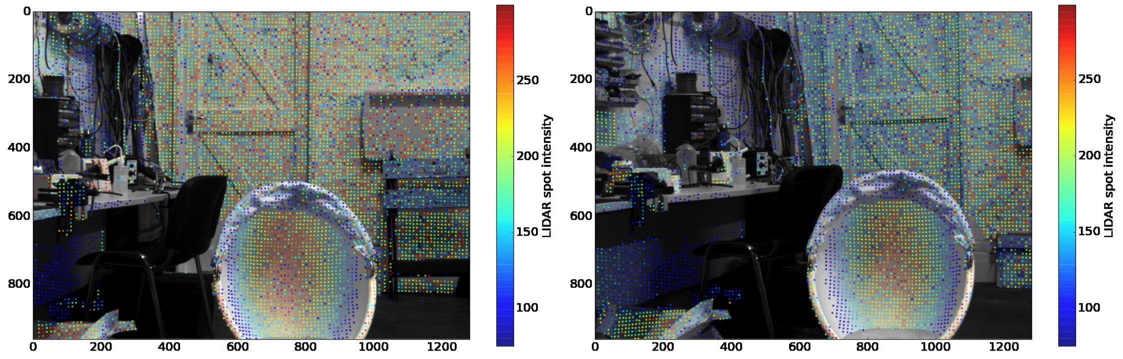


(a) (b)

Figure 5.10: Detected and filtered LIDAR spots for the (a) left and (b) right views of the 'wall' scene. Points are coloured by their reprojection error.

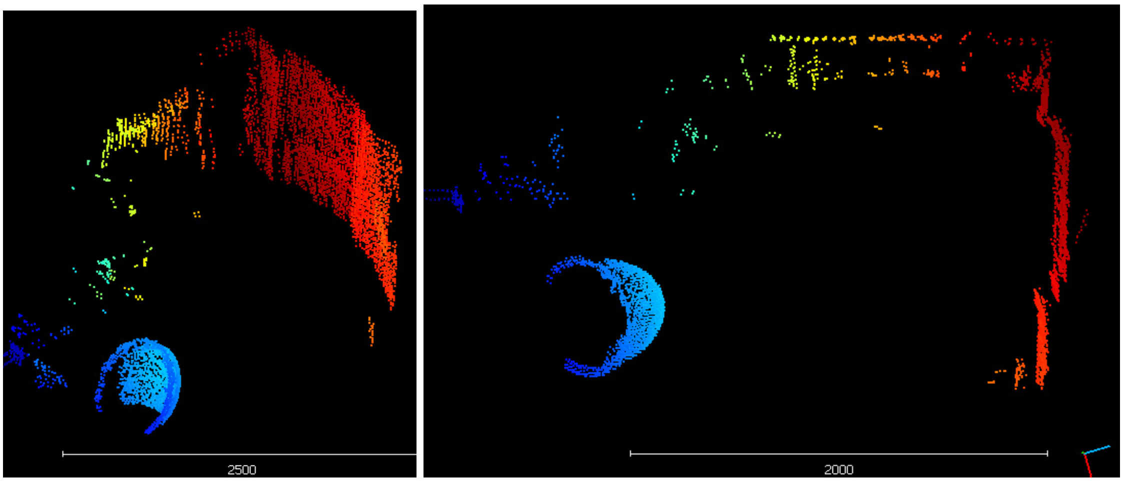
The second scene, 'unstructured' provides a more difficult calibration target. The volume of interest is larger, with the furthest points up to 5m away (compared to 2m for 'wall'). Additionally there are a wide variety of natural surfaces which post a challenge for spot detection. The LIDAR backscatter intensity maps, Figure 5.11, show that the LIDAR spot was not located in several darker regions of the images such as the office chair.

The intensity maps also highlight specularity within the scene. For instance, on the rear wall, the left image has an average backscatter intensity almost 100 counts higher than the right image. The image exposures are otherwise similar, ruling out any significant difference in camera setup. The aligned point clouds are shown in Figure 5.12.



(a) (b)

Figure 5.11: Detected LIDAR spot intensity for the 'unstructured' calibration scene.



(a) (b)

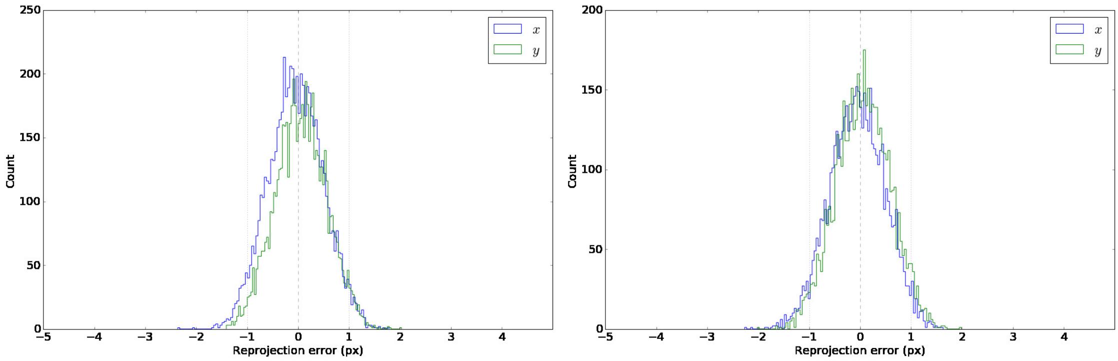
Figure 5.12: Aligned point cloud from 'unstructured' calibration scene. (a) side view (b) top view. The distance from the camera to the nearest point in the cloud is 2.2m

Following the same procedure as for 'wall', the derived intrinsic parameters are shown in Table 5.3.

Again, the LIDAR-derived intrinsic parameters were in close agreement with the truth values. As with the other scene, there were differences between individual parameters, but the RMS reprojection error was less than half a pixel for both cameras. This error is encouraging, but it should be expected that the error is lower at longer distances as each LIDAR spot fills fewer pixels in the image. This dataset also contains fewer points than 'wall' and the filtering process might have removed more outliers. The RMS reprojection error distribution is shown in Figure 5.13 and errors overlaid onto the image are shown in Figure 5.14.

Parameter	Left Truth	Left Fit	Right Truth	Right Fit
Focal length (mm)	8.31	8.32	8.29	8.29
Camera centre (px)	(662.5, 479.8)	(656.2, 481.2)	(657.7, 476.1)	(647.8, 496.0)
k1	-0.09	-0.13	-0.08	-0.09
k2	-0.42	0.05	-0.49	0.095
RMS error (px)	0.10	0.37	0.10	0.39
Pos. (x,y,z) (mm)	(0,0,0)	(0,0,0)	(443.3, 6.26, 50.0)	(446.4, 0.08, -1.36)
Rot. (x,y,z) (deg)	(0,0,0)	(0,0,0)	(-0.99, -10.6, 1.39)	(-0.41, -11.72, 2.06)

Table 5.3: Unrectified camera intrinsic and extrinsic parameters derived from LIDAR correspondences in the 'unstructured' scene, compared to truth values derived from a planar checkerboard target.



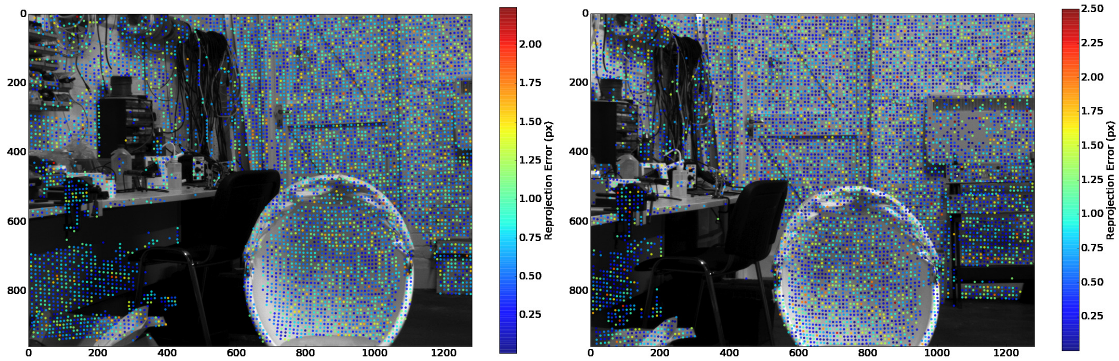
(a) (b)

Figure 5.13: Reprojection error distribution for 'unstructured' calibration scene (a) left camera (b) right camera.

5.5.3 Conclusions

For extrinsic calibration, using visible LIDAR points for calibration is accurate and simple. With robust point selection to avoid outliers, point clouds from both demonstration scenes were merged with accuracies on the order of 1-2mm. This result is reasonable given that the LIDAR has a specified accuracy of 1mm. Accuracy measurements on cooperative targets (section 4.5) suggest that the stereo rig should be able to achieve sub-mm accurate distance measurements up to 3m, however this assumes sub-pixel accurate correspondences. The LIDAR spot spans several pixels; it is possible that the LIDAR and stereo points are simply not derived from the same point on the surface, particularly if the surface is specularly reflective.

When used for intrinsic camera calibration, the algorithm is more sensitive to both outlying points and the distribution of calibration points within the field of view. The problem of choosing a suitable set of calibration points was solved by using RANSAC and robust filtering. In both scenes, intrinsic parameters derived from the LIDAR-camera correspondences were in agreement



(a) (b)

Figure 5.14: Detected and filtered LIDAR spots for the (a) left and (b) right views of the 'wall' scene. Points are coloured by their reprojection error.

with the ground truth calibration. There were deviations from the truth values, but sub-pixel reprojection errors were achieved in both calibration scenes. When using a calibration target based on a printed pattern, the combination of high accuracy object and image points leads to accurate intrinsic parameter estimation. With a LIDAR, spot detection is sub-pixel, but the object point accuracy is lower and the reprojection error reflects this.

Although the achieved calibration accuracy was not as good as a method involving a target (0.6 px vs 0.1 px), it is significantly simpler to perform. Using a faster scanning system such as a galvanometer would allow hundreds to thousands of points to be captured per minute. The expectation when using this calibration for combined stereo and LIDAR data capture is that it could be performed automatically for each scan. If a particular scene would benefit from an adjusted baseline or focus then calibration data may be derived directly from the scan. This calibration method would also be well suited for mass-production of stereo imaging systems where each system requires an individual calibration.

5.6 Summary

Camera calibration is necessary for accurate stereo reconstruction and there are a large range of algorithms to calculate the intrinsic and extrinsic parameters. Most methods rely on explicit calibration targets which provide highly accurate object and image points. A large downside of these methods is the need for user interaction.

A novel method of camera calibration was shown using a scanned visible-beam LIDAR. Unlike previous methods, no human interaction is required and no explicit calibration targets are required. By exploiting direct correspondences between the LIDAR 3D points and the 2D image of the laser spot, it is possible to calculate both intrinsic and extrinsic camera parameters.

Using this method, extrinsic parameters were recovered to high accuracy from two calibration scenes in real-world environments. For extrinsic calibration, the calculated reprojection and point cloud fitting errors were comparable to the state-of-the-art. When used for intrinsic parameter estimation, sub-pixel reprojection errors were achieved and individual parameters were in agreement with ground truth. The overall accuracy is strongly dependent on the performance of the LIDAR and scanning mechanism.

Stereo and LIDAR data fusion 6

In Chapter 3, previous methods for data fusion of stereo image matching with an additional (active) source of range data were compared. The vast majority of published techniques have used range data from ToFCs. ToFCs are attractive as they are increasingly affordable and offer real-time dense 3D over a wide field of view. However the accuracy of the data is low, the cameras are sensitive to ambient illumination and calibration is challenging. LIDAR solves many of the problems of ToFCs: measurements have higher relative accuracy, longer ranging is possible and measurements are robust to external illumination. These advantages come at the expense of acquisition time and data sparseness.

This chapter presents a data fusion algorithm that combines a scanning LIDAR with a region growing stereo matching algorithm: Gotcha (Shin and Muller, 2012). By using LIDAR to generate unambiguous seed points for the region growing process, dense matching is possible in low texture regions. First the relationship between the number of LIDAR seed points and the number of matched pixels is explored. Then, a more efficient scanning method is described that aims to reduce redundancy during data capture. This allows for a significant reduction in the number of LIDAR points while still producing dense matching results.

Results are shown from indoor scenes designed to be challenging for stereo matchers as well as outdoor data taken from the KITTI dataset (Geiger et al., 2013). Reference match results were produced using Gotcha and SGM, along with LIDAR-derived ground truth.

6.1 Introduction

Data fusion has been proposed as a viable technique for improving stereo matching by considering an additional source of range information such as a scanning LIDAR or ToFC. These active sensing techniques perform well in the poorly textured regions that passive stereo systems struggle to match. Previous methods (Section 3.3.2) fall into two categories: *a priori* methods which consider the (usually coarse or sparse) range data as a constraint during image matching; and *a posteriori* methods which produce combined point clouds from stereo and the range data. In both cases, it is usually necessary to acquire range information across the entire scene.

One way of improving match performance in textureless regions is to project some kind of pattern into the scene (see Chapter 7). The pattern might be a static, random, pattern or a series of structured patterns. While pattern projection is very effective at close range, there are a number of limitations: First, these systems are sensitive to ambient illumination. Infrared projection systems, even with spectral filtering, work poorly in sunlit or outdoor scenes. Projected pattern illumination decreases with distance according to the inverse square law, limiting usable range as the SNR decreases. Secondly, pattern spatial resolution decreases with increasing distance which can limit matching accuracy.

The proposed data fusion method aims to produce accurate, dense, disparity maps while requiring only a comparatively sparse LIDAR scan. Specifically, a region growing stereo matcher (Gotcha) is used (introduced in Section 3.2.7). Unlike prior research, the efficient algorithm presented in Section 6.5 aims to minimise the number of LIDAR points required.

This efficient technique was developed specifically for systems that offer control over the direction of the LIDAR. 3D scanning LIDAR systems are still expensive, costing tens of thousands of pounds. A lower cost alternative is integrating a 1D LIDAR with a scanning platform such as a gimbal mount or galvanometer (Chapter 3). These systems have a much slower scanning speed compared to the 1 Mpt/s achievable from commercial systems[1]. On the other hand, 1D systems can be more accurate and scanning platforms can offer superior angular resolution. Accurate, visible-beam, 1D LIDAR are available with acquisition speeds of around 100 Hz (e.g. the Jenoptik LUMOS [2]). Higher speed systems are available using infrared lasers, but these tend to have a lower accuracy. Since the scan time is directly proportional to the number of acquired points, any reduction in the number of points is beneficial.

In the following section, an overview of the standard Gotcha stereo matching algorithm is

[1] Note this is typically over a full hemisphere, so the number of points captured within the field of view of a camera per frame is generally less.

[2] https://www.jenoptik.com/products/metrology/laser-distance-sensors/lumos-laser-distance-meter-for-radiating-objects, accessed 20/9/2016

given, before a description of its extension is given to include LIDAR seed points.

6.2 Gotcha (Gruen-Otto-Chau ALSC)

Gotcha differs from most stereo matchers in that it does not (initially) consider every pixel in the images. The algorithm takes as an additional input a list of potential pixel-to-pixel correspondences between the left and right images. These correspondences (seedpoints) are generated by matching features from a detector like SIFT (Lowe, 2004). The algorithm is structured around Gruen's Adaptive Least Squares Correlation (ALSC) (Gruen, 1985) which is used to determine whether two image regions are similar enough to be considered matched. Otto and Chau (1989) described a region-growing approach where the disparity map is grown by applying ALSC to the initial seedpoints, then to the nearest 4- or 8-connected neighbours of those which match and so on until no further matches are possible. The current version of the algorithm is a 5th generation implementation which includes multi-processor support and, on an Intel 6700K (4 GHz) CPU with 8 virtual cores, takes approximately 1 minute per megapixel. Although this is not yet fast enough for realtime performance, the runtime is comparable with many of the top performing algorithms on the KITTI dataset [3]. A flowchart of the algorithm is shown in Figure 6.1.

This section describes firstly the mathematical basis behind ALSC and then the region growing process that Gotcha employs.

6.2.1 Adaptive Least Squares Correlation

Individual (prospective) correspondences are verified using Adaptive Least Squares Correlation (ALSC). The result of the algorithm is either a tiepoint with a refined location or a rejected tiepoint. The 'adaptive' moniker refers to the fact that the shape of the match window is changed on each iteration of the algorithm.

Like traditional correlation algorithms, Gotcha compares the local neighbourhoods (patches) around the pixels under consideration. The algorithm then attempts to minimise the correlation error between the left and right patches by performing small transformations to the right patch which include affine warping and translation.

[3] http://www.cvlibs.net/datasets/kitti/eval_scene_flow.php?benchmark=stereo, accessed 20/9/2016

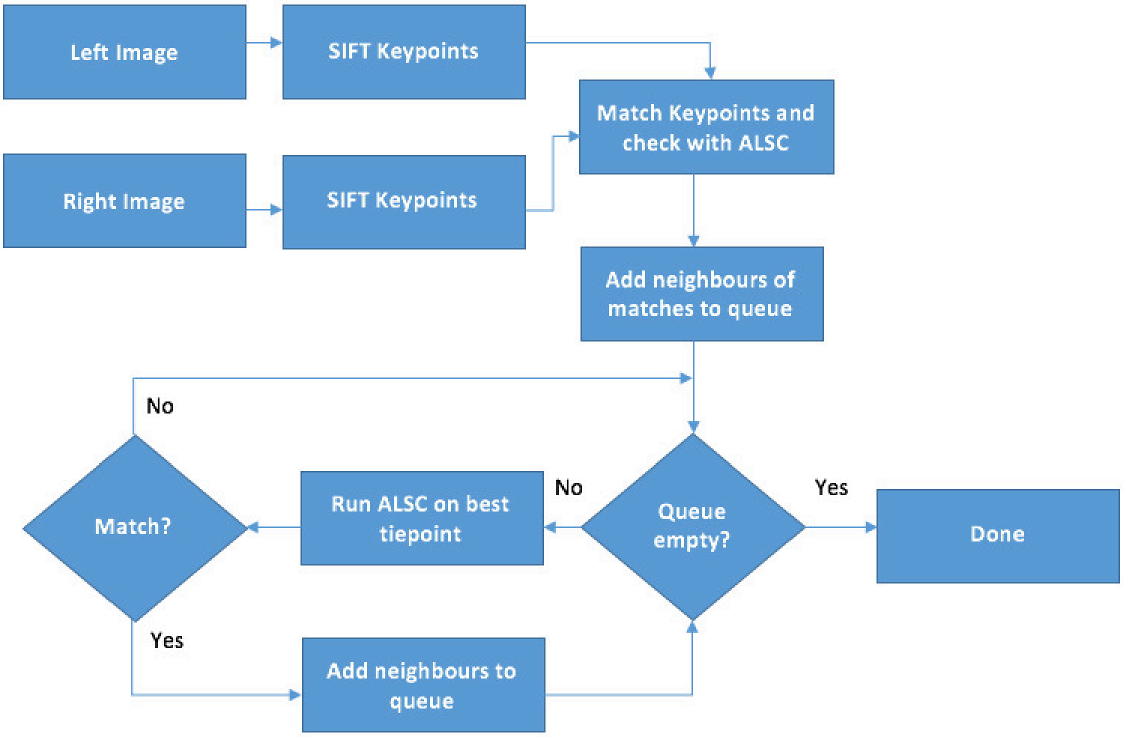


Figure 6.1: High level Gotcha algorithm flowchart. Here, SIFT keypoints are used as seedpoints, but in general any list of correspondences can be used for initialisation, including manually selected points.

Let the left image patch be defined as a discrete function $f(\mathbf{x}^L)$, centred on a pixel $\mathbf{x}^L = (x^L, y^L)$ and the right image patch as $g(\mathbf{x}^R)$, centred on $\mathbf{x}^R = (x^R, y^R)$. It is assumed that the patches are square with size $m \times m$ pixels, where m is odd. The values of the functions are obtained by sampling the image around these origins, $-m/2 > x^{L,R} > m/2$ and $-m/2 > y^{L,R} > m/2$. In the implementation used in this thesis, an optimised version of (Shin and Muller, 2012), sampling is performed using bilinear interpolation. The initial centre of the right patch is denoted with a tilde, e.g. $\widetilde{\mathrm{x}}^R$. The goal of the algorithm is to determine the location of g such that ideally $f(\mathbf{x}^L) = g(\mathbf{x}^R)$.

If it is assumed that the image functions are continuous and differentiable, then using Taylor expansion the relationship between the two patches can be written:

$$f(\mathrm{x}^{\mathrm{L}}) = g(\widetilde{\mathrm{x}}^{\mathrm{R}}) + \frac{\partial g(\widetilde{\mathrm{x}}^{\mathrm{R}})}{\partial x}dx + \frac{g(\widetilde{\mathrm{x}}^{\mathrm{R}})}{\partial x}dy + e(\widetilde{\mathrm{x}}^{\mathrm{R}}) \tag{6.1}$$

where $e(\widetilde{\mathrm{x}}^{\mathrm{R}})$ is an error function accounting for noise in each image.

If the right patch is allowed to be affine-distorted (straight lines are preserved) as well as translated then the relationship between the two patches is:

$$\mathbf{x}^L = \begin{bmatrix} a_{11} & a_{12} & t_x \\ a_{21} & a_{22} & t_y \\ 0 & 0 & 1 \end{bmatrix} \widetilde{\mathbf{x}}^R + e(\widetilde{\mathbf{x}}^R) = \mathbf{A}\widetilde{\mathbf{x}}^R + e(\widetilde{\mathbf{x}}^R) \tag{6.2}$$

assuming that the coordinates are expressed in homogenous form. Expanding and differentiating these equations gives closed form expressions for dx and dy.

$$dx = dt_x + a_{11}\widetilde{x}^R + a_{12}\widetilde{y}^R \tag{6.3}$$

$$dy = dt_y + a_{21}\widetilde{x}^R + a_{22}\widetilde{y}^R \tag{6.4}$$

So equation 6.1 can be written as:

$$f(\mathbf{x}^{\mathbf{L}}) - g(\widetilde{\mathbf{x}}^R) - e(\widetilde{\mathbf{x}}^R) = \partial_x g(\widetilde{\mathbf{x}}^R)(dt_x + a_{11}\widetilde{x}^R + a_{12}\widetilde{y}^R) + \partial_y g(\widetilde{\mathbf{x}}^R)(dt_y + a_{21}\widetilde{x}^R + a_{22}\widetilde{y}^R) \tag{6.5}$$

which can be expressed in least squares form:

$$\begin{bmatrix} f(\mathbf{x_1^L}) - g(\widetilde{\mathbf{x}}\mathbf{_1^R}) \\ \vdots \\ f(\mathbf{x_{m^2}^L}) - g(\widetilde{\mathbf{x}}\mathbf{_{m^2}^R}) \end{bmatrix} = \mathbf{As} + e(\mathbf{x^R}) \tag{6.6}$$

$$l = \mathbf{As} + e(\mathbf{x^R}) \tag{6.7}$$

where $\mathbf{A}$ is an $m^2 \times 6$ matrix with the ith row:

$$\mathbf{A}_i = [\partial_x g(\widetilde{\mathbf{x}}\mathbf{_1^R})x, \partial_x g(\widetilde{\mathbf{x}}\mathbf{_1^R})y, \partial_x g(\widetilde{\mathbf{x}}\mathbf{_1^R}), \partial_y g(\widetilde{\mathbf{x}}\mathbf{_1^R})x, \partial_y g(\widetilde{\mathbf{x}}\mathbf{_1^R})y, \partial_y g(\widetilde{\mathbf{x}}\mathbf{_1^R})] \tag{6.8}$$

and the solution vector $\mathbf{s} = [da_{11}, da_{12}, dt_x, da_{21}, da_{22}, dt_y]$. Solving for $\mathbf{s}$:

$$s = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{1} \tag{6.9}$$

Since $\mathbf{A}^T\mathbf{A}$ is positive-definite, real and symmetric its inverse may be calculated efficiently using Cholesky decomposition (Otto and Chau, 1989). The solution vector contains updated shift and warp parameters for the right patch. These parameters are used to resample the right patch and the process iterates, refining the match to sub-pixel accuracy. The largest eigenvalue of the correlation matrix, M^TM is used to determine whether a match has converged. This eigenvalue is referred to as the similarity. The threshold similarity must be set by the user and is typically defaulted to around 100 which is suitable for a wide variety of scenes. Gotcha is capable of refining seedpoints with a significant initial error, as much as ten pixels (Shin and Muller, 2012).

6.2.2 Region growing

Starting with an initial list of seedpoints, Gotcha considers each in turn, applying ALSC to verify the matches. The ALSC similarity score (a scalar value) is used to insert these initial seeds (and their neighbours) into a priority queue[4] sorted by similarity. In order to improve the quality of the disparity map, the seedpoint with the highest ASLC similarity is considered at each iteration, a 'best-first' strategy. Intuitively this implies that the regions are grown in the order of match confidence. The algorithm terminates when the priority queue is empty. The outputs of Gotcha are vertical and horizontal disparity maps and a confidence map, containing the ALSC similarity for each matched pixel.

Additional checks can be used to determine whether a match is valid, such as the vertical disparity (which should be close to zero for well-rectified images). Limitations are also placed on the number of itertions of ALSC performed per potential match and the distance between the initial and final patch locations.

Unlike most stereo matchers, Gotcha places no constraint on minimum or maximum disparity. Stereo pairs with large disparity ranges do not take any longer to match than pairs with small ranges. Assuming each ALSC process takes a constant amount of time, the runtime of the algorithm is directly proportional to the number of pixels in the image regardless of the range of disparities present. This assumption is valid provided the number of iterations within one round of ALSC is fixed or capped (12 iterations is set by default in the software).

Seed points are typically generated using SIFT. A SIFT keypoint is defined as a location in the image combined with a descriptor vector. SIFT descriptors are 128-dimensional vectors calculated from local image gradients. A list of keypoints is generated for each of the left and right images,

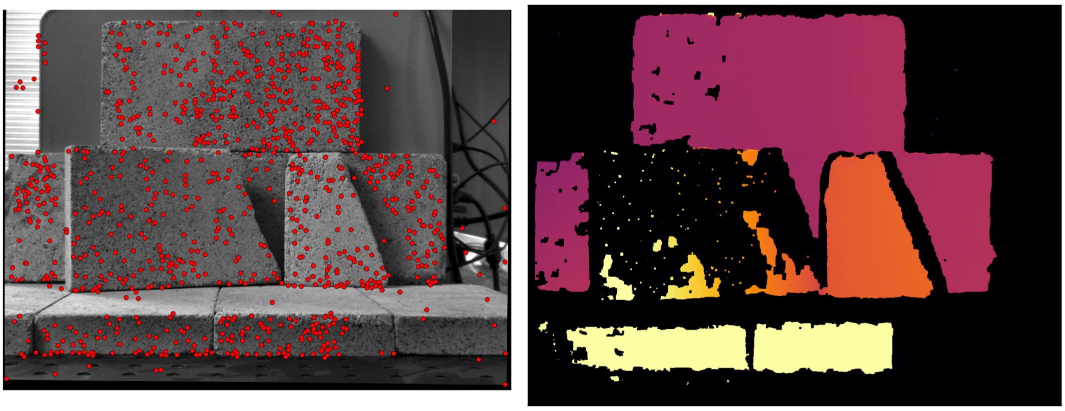
[4] A priority queue is a data structure which has the property that each element has a score. The order of the queue is preserved when new elements are inserted and the element with the highest (or lowest) score is always at the front.

often several thousand points are produced. The similarity between a set of keypoints is defined to be the Euclidean distance between their descriptors, i.e. for keypoint descriptors K_j^L and K_k^R in the left and right images, the distaance is:

$$d_{jk} = \sqrt{\sum_{i=0}^{i=127} (K_{ji}^L - K_{ki}^R)^2} \tag{6.10}$$

A brute force approach calculates d_{jk} over all the possible pairs of keypoints. The potential match for each keypoint is its nearest neighbour in the other image. Lowe proposed using the ratio of distances between the nearest and second-nearest neighbours to prune false matches. If the ratio of distances between the neighbours is greater than some threshold, t, then the match is rejected. An empirical threshold of $t = 0.8$ is widely used in the literature. If the ratio is very low, then the nearest neighbour is much more similar than the second best candidate and therefore likely to be an inlier. If the ratio approaches 1, then the match is ambiguous and should be rejected. This threshold is important when considering homogenous regions, where many descriptors may have similar values. As a result, seed points generated using this approach tend to be restricted to uniquely textured regions. For epipolar rectified imagers, to increase the robustness of the matching process, keypoint pairs are only considered if their y-difference is < 2 px. This constraint is not applicable if non-epipolar imagery is used.

While Gotcha is able to match image regions accurately, this relies on there being initial seed points nearby and strong texture for the disparity map to grow into. This is illustrated in Figure 6.2 on a test scene (Bricks).



(a) (b)

Figure 6.2: Bricks scene. (a) matched SIFT keypoints marked in red (b) Gotcha disparity map.

The distribution of seed points correlates to high texture image regions. These regions are isolated by low texture regions, which the disparity map is unable to grow into. It is possible that the image contains texture that is sufficient to allow region growing, but not unique enough to allow unambiguous keypoint matching. The next section proposes using LIDAR measurements to generate unambiguous seed points in the image regions.

6.3 LIDAR seed points

As described in the previous section, Gotcha produces a disparity sheet by region growing, initialised by a set of seed points generated by a feature detector. The matching process is robust, but these matched seed points are generally only found in highly textured image regions. This does not necessarily mean that the remainder of the image cannot be matched, simply that it was not possible to generate unambiguous seed points in those areas. Therefore, the following hypothesis is proposed: if unambiguous seed points are provided in these low texture regions, there may be sufficient local texture to allow dense matching. These seed points may be generated using a LIDAR system.

In Chapter 4 it was discussed how a scanning LIDAR and camera system can be cross-calibrated. This allows LIDAR 3D points to be projected onto the 2D camera sensor. There is a choice between using seed points directly imaged by the camera or points derived from the LIDAR-camera calibration. Direct imaging of the LIDAR spot is somewhat more robust, as occlusions can be detected by thresholding the image to determine whether the LIDAR spot is visible or not. However, this limits the acquisition rate to that of the camera which is generally several tens of frames per second. Also this relies on the choice of a suitable threshold for spot detection which varies given the dynamic range present in any particular scene. An alternative is to calculate the expected location of the laser spot using the cross-calibration, which allows the use of much faster LIDAR systems. The projected spot locations may be occluded in either image, but ALSC can be employed for robust filtering.

The Gotcha algorithm is used unchanged, except that the input seed points are derived from LIDAR measurements instead of, or in addition to, SIFT keypoints. The following section presents results using this approach in indoor and outdoor environments.

6.4 Results

6.4.1 Indoor scenes

Stereo imagery and LIDAR scans were acquired for five challenging indoor scenes. Ground truth was derived from the LIDAR scans by direct observation of the LIDAR spot in each image. Gotcha was used to stereo match the scenes using default settings (section 4.5). Feature points were generated using SIFT. Match results from SGM are also presented for comparison using 8-way dynamic programming. The four scenes (Bricks, Corner, Potplant and Chair) and their ground truth images are shown in Figures 6.3, 6.4, 6.5 and 6.6. Although some efforts were taken to keep image brightness constant between views, through methods such as histogram equalisation, there are still some illumination differences. These differences were kept as they provided a more realistic dataset. Small holes, < 3 px, in the disparity maps were filled using grayscale morphological opening and closing as a post-processing step. All images were 1.3M px.

For each scene, 10,000 random LIDAR points were withheld for ground truth evaluation and were not used for disparity map seeding. For each scene, LIDAR points were randomly sampled from the scan and used as seed points for Gotcha. These results give an indication of the expected improvement in the number of matched pixels with respect to the number of seed points used.

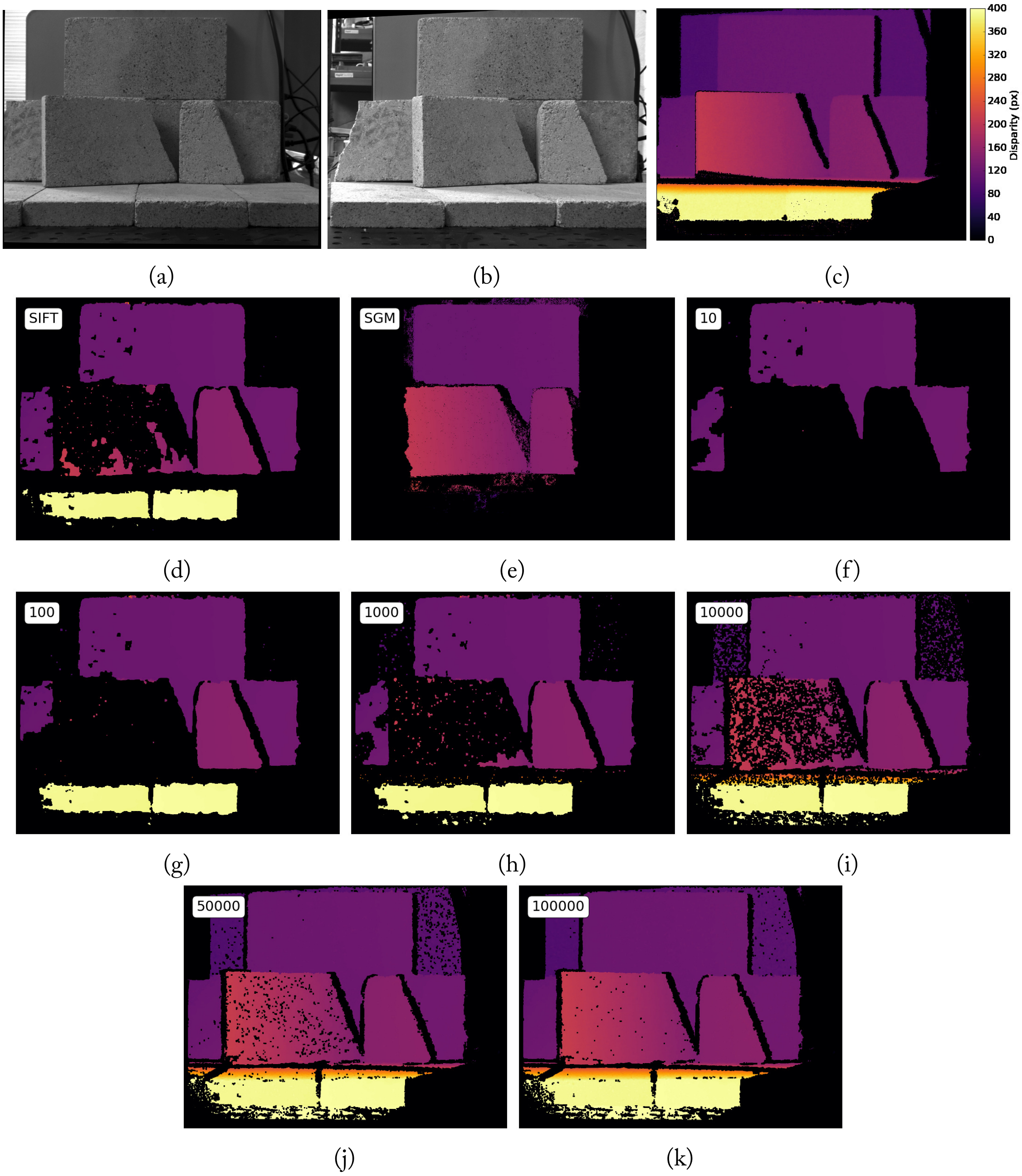
Figure 6.8 shows the relationship between the number of random LIDAR seeds and the overall number of matched pixels.

Table 6.1 lists the number of matched pixels for each scene, comparing Gotcha using SIFT alone and using SIFT and LIDAR seeds. These results use the full LIDAR scan for seed points, but note from Figure 6.8 that in most cases the number of matched pixels had plateaued by around 50-100k LIDAR seed points.

Scene	Area (deg^2)	Res. (deg)	Scan points	SIFT only (px)	SIFT+LIDAR (px)
Bricks	777	0.05	310,800	495,676	778,688
Corner	868	0.1	86,800	233,245	761,204
Chair	632	0.1	63,200	94,694	723,823
Biba	736	0.05	294,400	105,351	729,214
PotPlant	735	0.05	294,000	93,510	465,698

Table 6.1: Number of matched pixels compared to LIDAR seed points used for disparity map enhancement in each test scene. The rightmost two columns compare the number of matched pixels between standard Gotcha and Gotcha using SIFT and LIDAR seeds.

Figure 6.9 shows the mean disparity error and error standard deviation for each scene. The



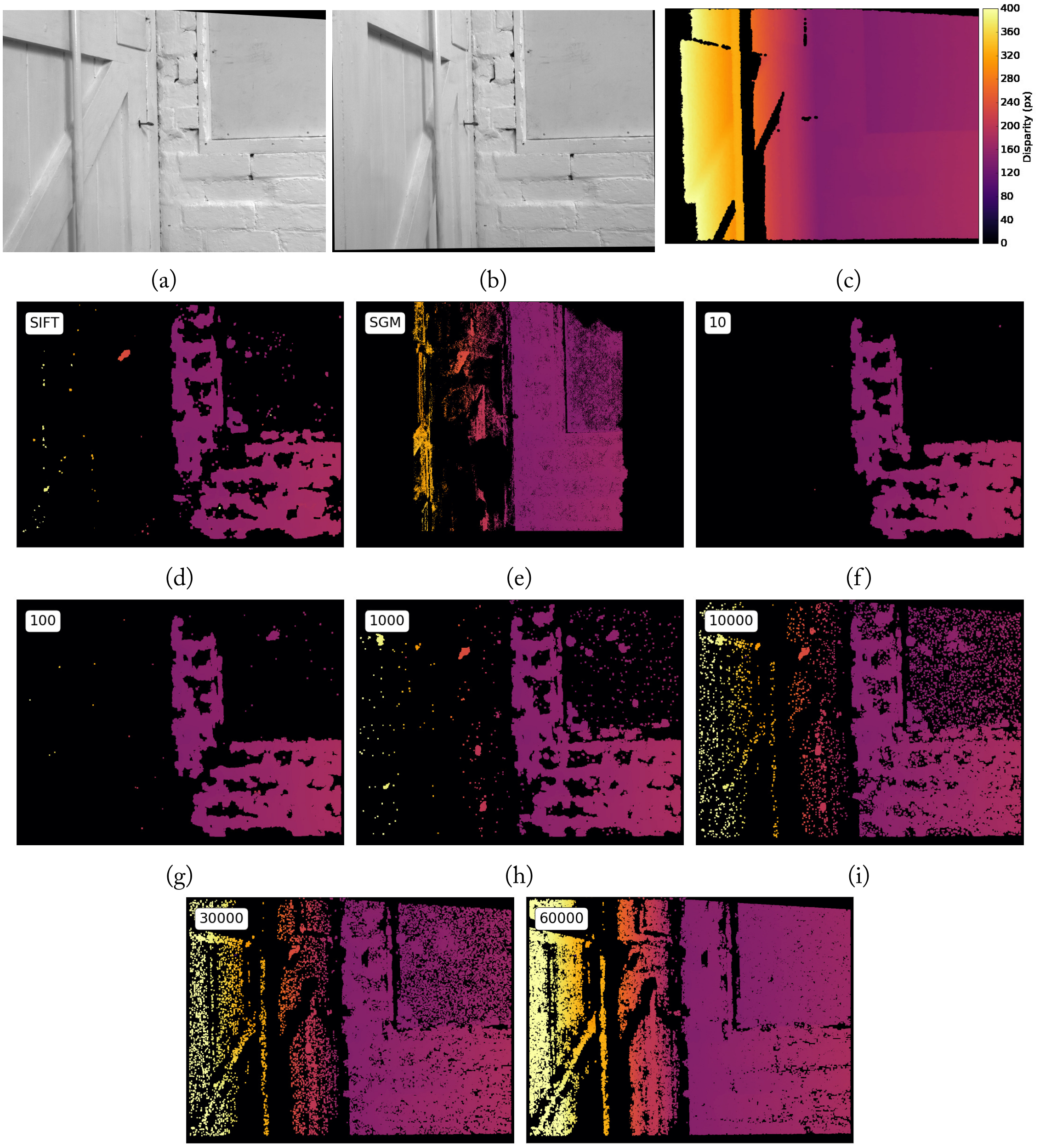
(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

(j) (k)

Figure 6.3: Bricks disparity map. (a, b) Left and right stereo pair (c) LIDAR-derived ground truth (d) Gotcha disparity using 1080 SIFT seed points. (e) SGM disparity (f - j) Gotcha disparity using 10, 100, 1000, 10000, 100000 random LIDAR seed points.



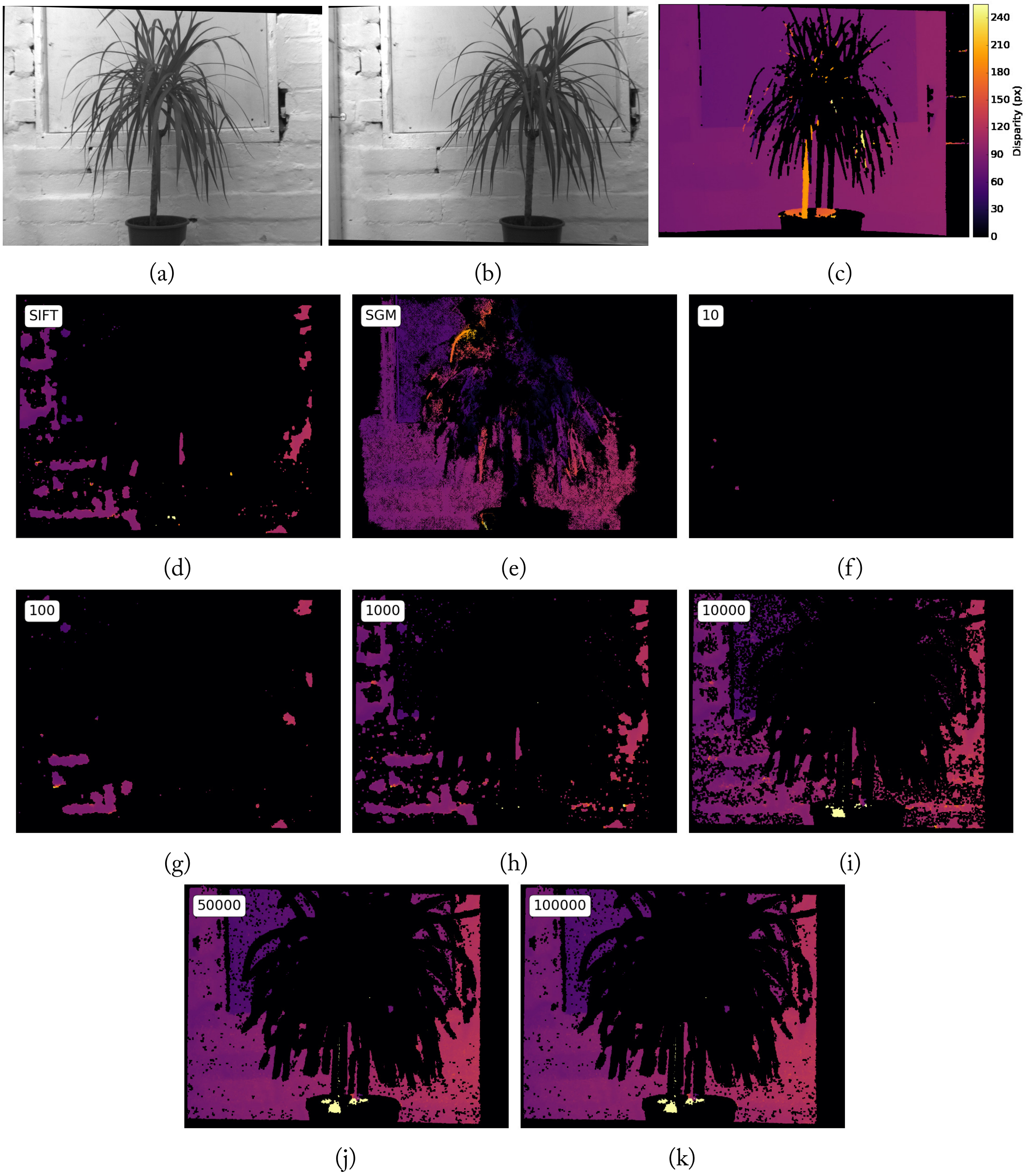
(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

(j) (k)

Figure 6.4: Corner disparity map. (a, b) Left and right stereo pair (c) LIDAR-derived ground truth (d) Gotcha disparity using 834 SIFT seed points. (e) SGM disparity (f - j) Gotcha disparity using 10, 100, 1000, 10000, 30000, 60000 random LIDAR seed points.



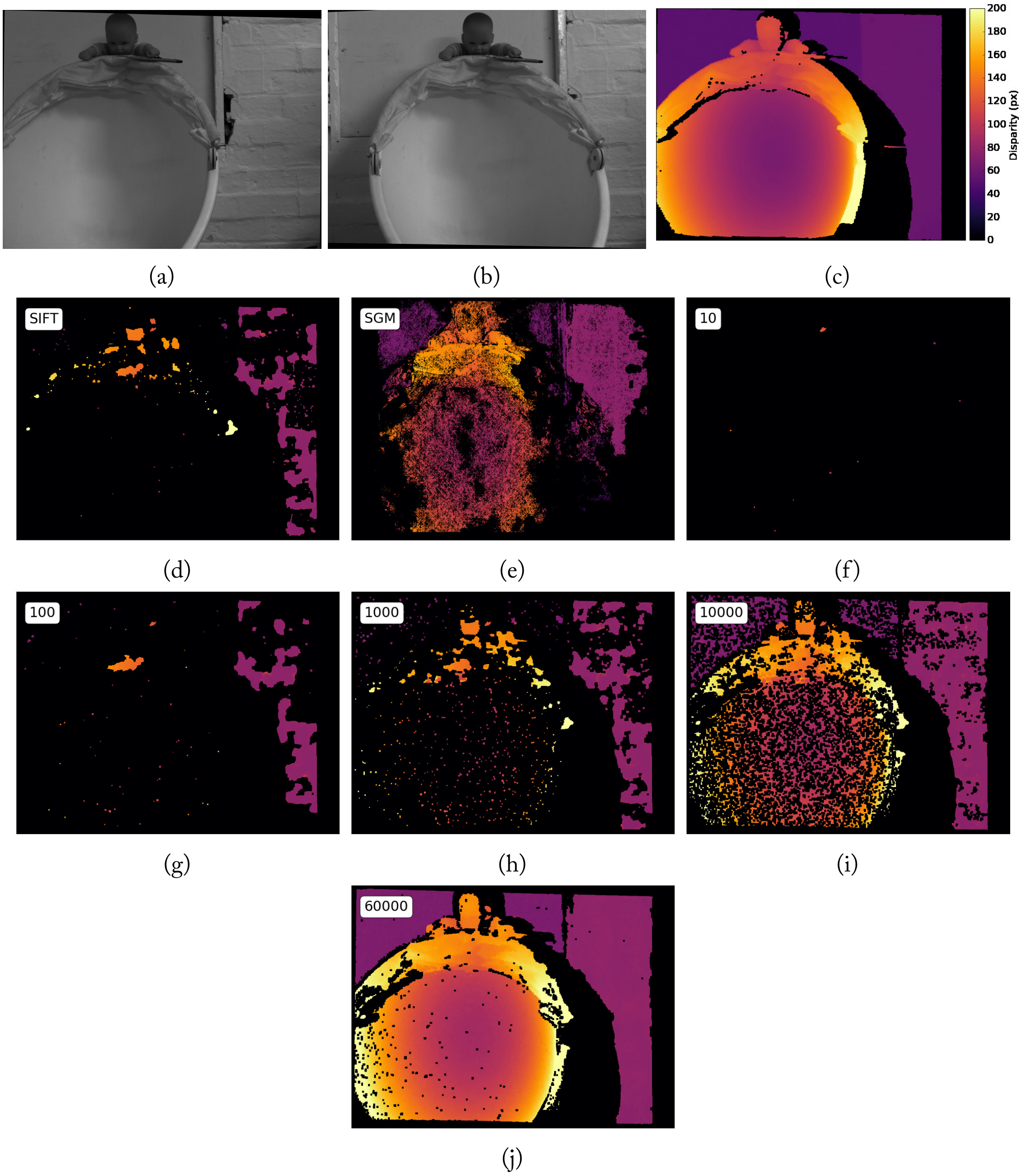
(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

(j) (k)

Figure 6.5: Potplant disparity map. (a, b) Left and right stereo pair (c) LIDAR-derived ground truth (d) Gotcha disparity using 658 SIFT seed points. (e) SGM disparity (f - j) Gotcha disparity using 10, 100, 1000, 10000, 50000, 100000 random LIDAR seed points.



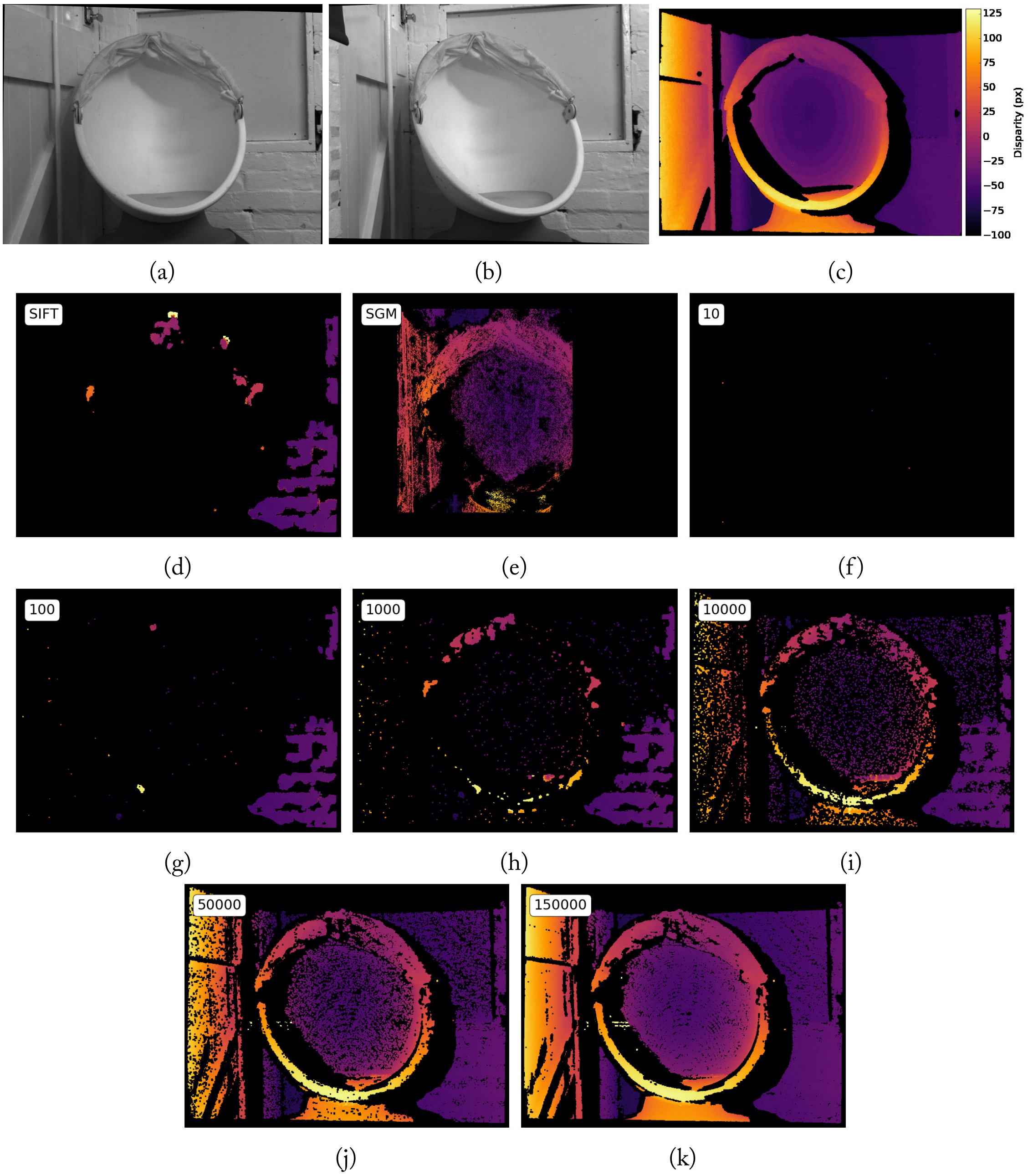
(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

(j)

Figure 6.6: Chair disparity map. (a, b) Left and right stereo pair (c) LIDAR-derived ground truth (d) Gotcha disparity using 595 SIFT seed points. (e) SGM disparity (f - j) Gotcha disparity using 10, 100, 1000, 10000, 60000 random LIDAR seed points.



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

(j) (k)

Figure 6.7: biba disparity map. (a, b) Left and right stereo pair (c) LIDAR-derived ground truth (d) Gotcha disparity using 96 SIFT seed points. (e) SGM disparity (f - j) Gotcha disparity using 10, 100, 1000, 10000, 10000, 50000 and 150000 random LIDAR seed points.

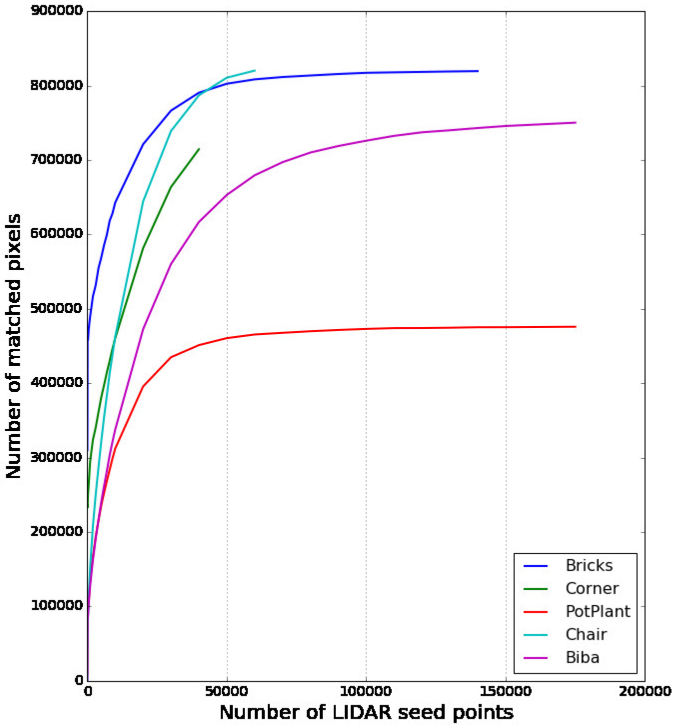


Figure 6.8: Relationship between the number of LIDAR seed points and the number of matched pixels in the disparity map.

disparity error was calculated by comparing the fused disparity maps against the LIDAR ground truth. To avoid bias, errors were only calculated against the withheld LIDAR measurements (i.e. those which were not used to seed Gotcha).

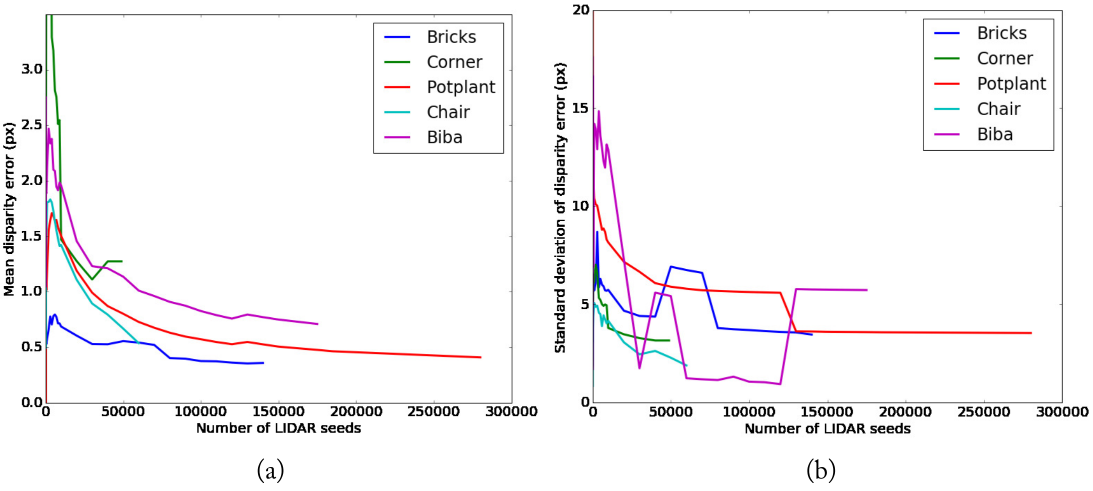


Figure 6.9: Indoor test scenes (a) Mean disparity error and (b) Standard deviation of disparity error as a function of the number of LIDAR seed points used.

6.4.2 Outdoor scenes (KITTI)

The KITTI dataset provides cross-calibrated stereo and LIDAR data (see Chapter 2) acquired from a moving vehicle in a variety of urban scenes. The onboard Velodyne LIDAR on the KITTI vehicle only covers the lower half of the frame and while around 100 k LIDAR points are captured per video frame, only around 20 k are in view of the forward-facing stereo cameras.

Raw data were taken from the 2011_09_26 run comprising 107 stereo pairs. Raw datasets[5] from KITTI include rectified stereo imagery, LIDAR returns in the LIDAR coordinate system and an external calibration between the LIDAR and camera coordinate systems. LIDAR-derived seed points were generated for each image using the calibration files provided. These LIDAR points therefore represent ground truth, and are used to derive the official KITTI ground truth data.

The KITTI organisation also provide benchmark data sets without ground truth, which may be used for independent stereo matching assessement. Since this assessment relies on withholding the LIDAR data, it is not currently possible to use the online benchmark utility for stereo matching algorithms that involve data fusion, even though in practice this is a sensible route for robust scene reconstruction.

In order to assess matching accuracy, 60% of the LIDAR data were withheld to act as a ground truth. The remaining 40% of the LIDAR data were used as seeds for stereo matching. The data were sampled randomly to produce this split. Therefore there were around 8k ground truth points and 12k seeds per stereo pair. The downside of splitting the data is that GOTCHA is not able to exploit the full set of input seed points and is likely to match fewer pixels. For interest, matching was also performed using the full set of LIDAR seed points with the caveat that accuracy measurements were not possible.

For each image, the following inputs to Gotcha were used:

- Matched SIFT keypoints alone
- Matched SIFT keypoints and a reduced (40%) set of LIDAR seedpoints, leaving 60% for ground truth (accuracy) evaluation.
- Matched SIFT keypoints and all available LIDAR seedpoints

[5] Datasets are located at: `http://www.cvlibs.net/datasets/kitti/raw_data.php`, accessed 11/11/2016

The workflow for preparing the raw data and matching is shown in Figure 6.10.

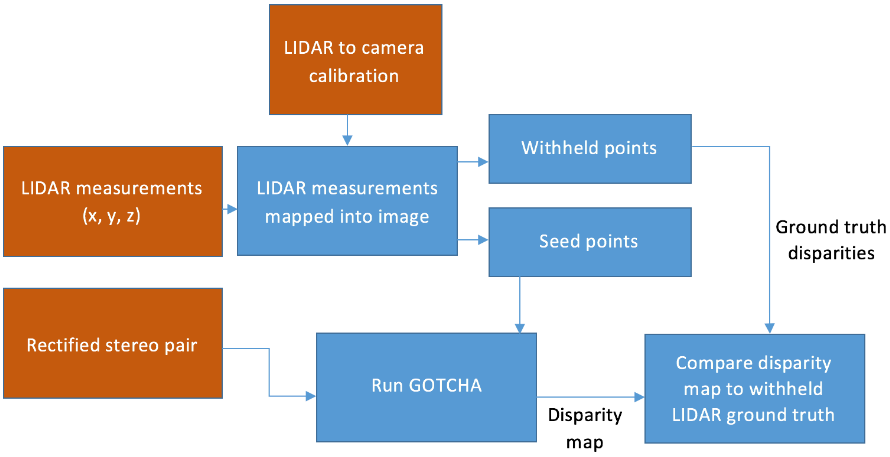

LIDAR to camera calibration
LIDAR measurements (x, y, z)
LIDAR measurements mapped into image
Withheld points
Seed points
Ground truth disparities
Rectified stereo pair
Run GOTCHA
Disparity map
Compare disparity map to withheld LIDAR ground truth

Figure 6.10: Processing workflow for KITTI data using GOTCHA for matching.

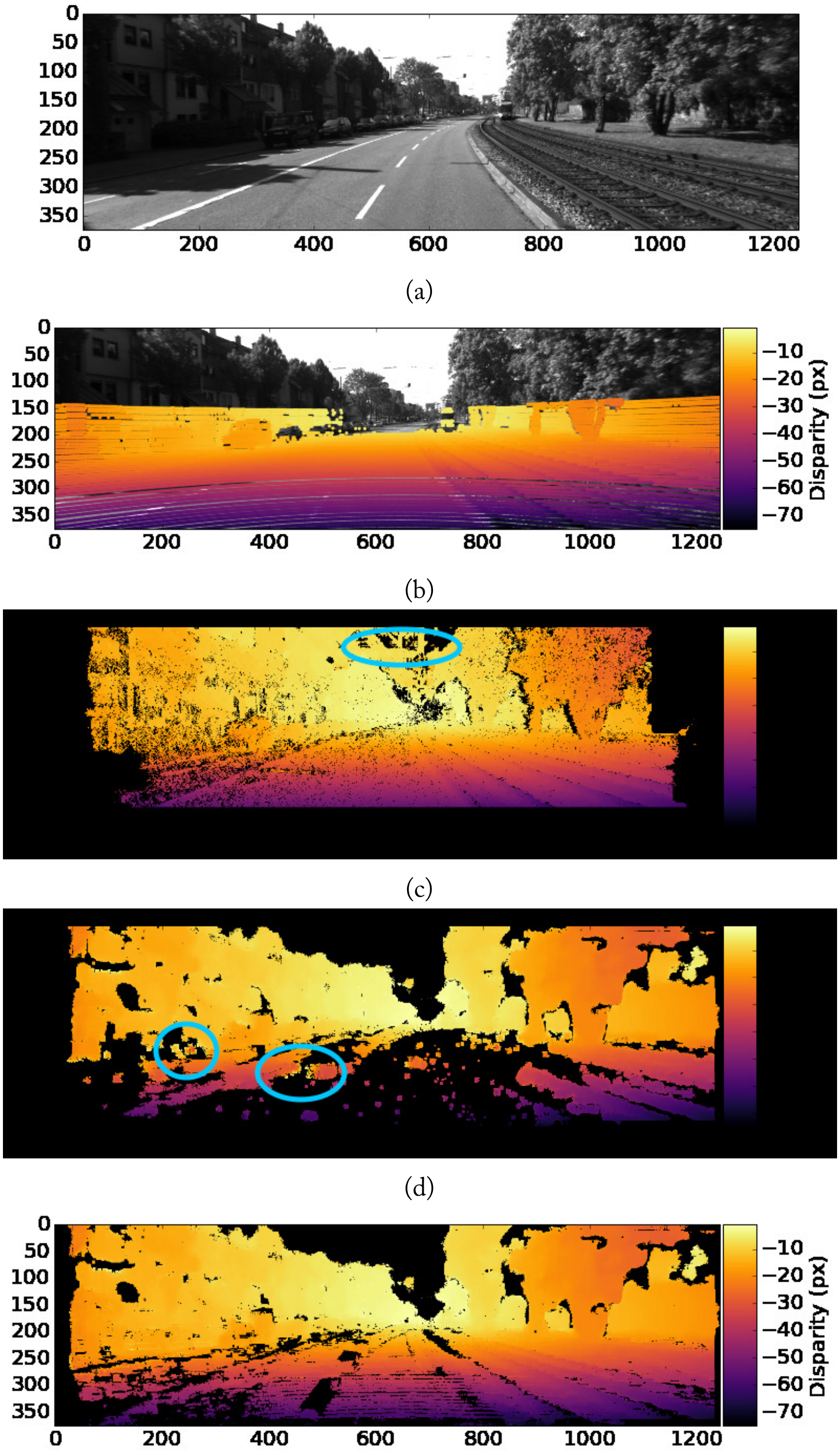
All images were also matched using SGM using a census cost with a 9px window for comparison. Figure 6.11 shows a typical KITTI image along with the LIDAR scan location and matching results from SGM, Gotcha and Gotcha with LIDAR seed points.

For clarity and ease of comparison, LIDAR measurements were converted into disparities. This examples highlights the difficulty of outdoor imaging. There is a high dynamic range in the scene, from the overexposed sky to deeply shadowed regions at the side of the road. The road itself is a large, low-texture region. Since there was no control over the LIDAR acquisition rate, or where the Velodyne LIDAR was aimed, all LIDAR points were used as seeds for evaluation.

There are several blunders in the SIFT-only disparity map (Figure 6.11d), which have been circled. These errors are not present in the SIFT+LIDAR result. The results from SGM were typically very good, showing excellent reconstruction of the road surface and the rest of the image. However, SGM (Figure 6.11c) also (incorrectly) matched the sky and the disparity map does not extend to either edge of the image.

Figure 6.12 shows the error statistics for the various matchers and seed point combinations. Figure 6.12a shows a histogram of the number of matched pixels over all the images for SIFT, SIFT+All LIDAR and SGM (this does not take into account matching accuracy). The remaining subfigures use a reduced set of LIDAR seed points. Figure 6.12b shows the proportion of matched pixels[6] for which ground truth was available, with an error of less than 2 px. Figure 6.12c shows the

[6] Using the ratio of 'good'/total matches allows for different numbers of matched pixels in each image with different matchers



(a)

(b)

(c)

(d)

(e)

Figure 6.11: Example KITTI image (a) Left stereo image (b) Left stereo image with LIDAR points overlaid. Each LIDAR measurement is shown with a large 7 px spot for clarity (c) SGM disparity map (d) Gotcha disparity map using SIFT keypoints (e) Gotcha disparity map using SIFT and LIDAR seed points. Some obvious disparity map errors are ringed in blue.

absolute error over all 107 scenes and Figures 6.12d and 6.12e show the per-scene error mean and standard deviation respectively. Only matched pixels for which ground truth was available were used for statistical accuracy analysis.

6.4.3 Discussion

The five indoor stereo pairs presented here proved challenging for stereo matching. SGM tended to perform better than Gotcha using SIFT seeds alone. Gotcha only performed well in the Bricks scene, but failed to match one of the brick surfaces (centre-left of Figure 6.3d). The reason for this poor performance is the previously discussed limitation of using image features for region growing. In poor texture regions, there are too many similar features for reliable initial matching.

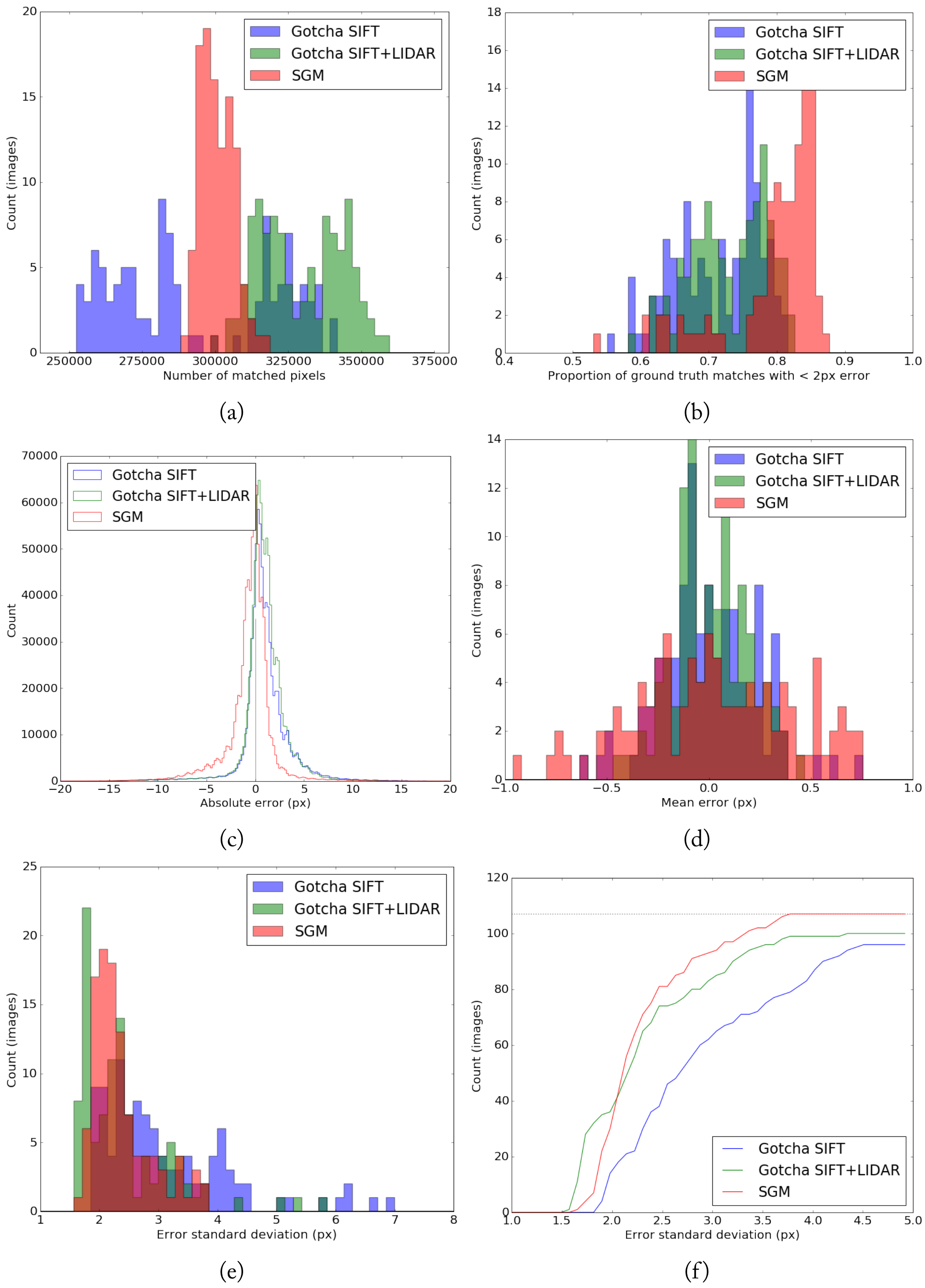
In some scenes, there was sufficient texture that only 10 LIDAR seeds were required to produce a disparity map comparable to using SIFT keypoints (e.g. Corner and Bricks). This hints at the problem of redundancy, since these densely matched regions did not become significantly improved until tens of thousands more LIDAR points were used.

The relationship between the number of seed points used and disparity map density was asymptotic (Figure 6.8) and this effect was observed for all scenes. Significant disparity map improvement was observed up to around 100 k seed points, at which point there was a rapidly diminished improvement (e.g. PotPlant, Bricks and Biba). Although the Chair and Corner scenes were scanned at a lower resolution than the others, the data suggests the same trend.

The eventual plateau in the number of matched pixels was consistent with the amount of occlusion in each scene as shown in the ground truth images. In other words, this plateau represents a disparity map that cannot be improved further. This suggests that an appropriate strategy for LIDAR seeding would be to scan the scene using at progressively higher resolution until the rate of change of disparity map density slows below a pre-defined threshold. Experimentally it was not necessary to scan at a higher resolution than 0.05 degrees. Scanning at higher resolutions using the gimbal system would have resulted in prohibitively long acquisition times.

As with the number of matched pixels, the disparity map error (Figure 6.9a) showed an asymptotic decrease with a plateau above 50-100k seed points. The mean disparity error was sub-pixel (or approaching sub-pixel) for most scenes, with the exception of Corner.

In the case of outdoor data, obtained from the KITTI dataset, SIFT+LIDAR seed points



(a)

(b)

(c)

(d)

(e)

(f)

Figure 6.12: (a) Histogram of the number of matched pixels for 107 KITTI stereo pairs using different matching methods. (b) Proportion of matched ground truth pixels with an error < 2px. (c) Pixel-wise disparity errors compared to ground truth (d) Mean disparity error for each stereo pair (e) Disparity error standard deviation for each stereo pair. (f) Cumulative distribution function for error standard deviation.

provided an improvement in the number of matched pixels over both SGM and Gotcha using SIFT seeds alone (Figure 6.12a). In terms of accuracy, SGM performed best overall, with around 10% more 'good' points with <2 px error than Gotcha. Using Gotcha with LIDAR seed points provided a slight improvement over SIFT alone (Figure 6.12b). The fused result also avoided the blunders produced by Gotcha or SGM alone. Unlike SGM, Gotcha was able to match to the edge of the frame and contained generally fewer blunders such as the sky area. All three matching methods showed similar error characteristics with an overall error distribution centred around zero pixels and per-scene mean errors around zero pixels with a spread of ±1-2 px. There was some evidence for a negative bias when using SGM and a positive bias with Gotcha, both less than 2 px, but it is not clear from where this originated. Considering the error standard deviation, all three methods show a similar distribution throughout the dataset. SGM and SIFT+LIDAR both outperform SIFT alone, which is more clearly visible in the cumulative distribution function (CDF) in Figure 6.12f.

LIDAR and stereo fusion is therefore a viable technique for dense and robust outdoor matching, in environments where it would be impossible to use competing techniques such as ToFCs.

In the case of autonomous driving, the benefit of using a combined stereo and LIDAR approach is that the LIDAR scan does not need to have a large vertical range. However for some applications like mapping, it is more useful to reconstruct the entire frame and this is more easily achieved using stereo. LIDAR-seeded region growing does not place any particular requirements on the choice of sensing hardware and could therefore be integrated into existing robotic platforms without modification.

6.5 Efficient LIDAR scanning

Data fusion naturally introduces redundancy, which may or may not be desirable. Redundancy occurs if a distance is obtained both from stereo matching and an additional ranging system. An advantage of redundancy is validation: if two independent measuring systems return the same distance for a particular point, there is evidence that the measurement is valid. This also enables one source of range data to act as a filter for the other, and points where there is disagreement can be removed. A disadvantage of redundancy is that it is inefficient, since active ranging is only required where stereo matching fails. While this is not so much of an issue for ToFCs and other area sensors, for scanned systems the acquisition time is typically linearly dependent on the number of points to be acquired.

In this section, a coarse-to-fine approach is presented which allows scenes to be matched efficiently, with a reduced number of LIDAR points compared to scanning the entire scene at full resolution. Naively, the number of points, n, required to scan a scene at a particular resolution, r, is $n = A/r$ for a scan area A. However parts of the scene will be occluded and others might be easily matched using stereo. By avoiding scanning image regions that are already matched or occluded, the number of LIDAR measurements is reduced.

6.5.1 Selective LIDAR scanning

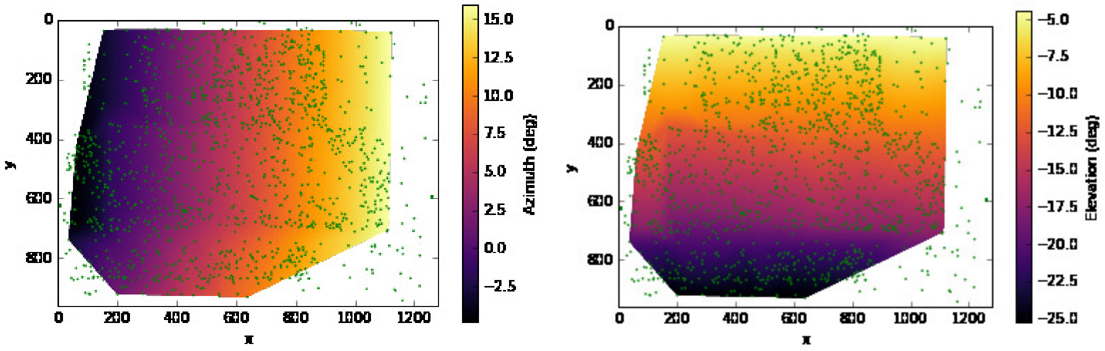
The goal of coarse-to-fine scanning is to progressively improve the disparity map by scanning the scene, selectively, at increasingly fine resolutions in unmatched regions. This selective scanning requires a mapping between pixel coordinates in the image and the direction that the LIDAR should be pointed. One to one mapping requires the distance for each pixel to be known. Since the distance for each unmatched pixel is by definition unknown prior to reconstruction, this mapping must be estimated.

An effective prior is obtained by acquiring a coarse scan of the scene and interpolating between un-occluded points to calculate the transformation between the two systems. Occlusion is detected either by direct detection (visible beam LIDAR) or by ALSC verification of the calculated position of the LIDAR spot.

The mapping is calculated via bilinear or bicubic interpolation over the points scattered throughout the image. There are two maps, one for each of the axes of the scanner. An example from a 1.6 degree resolution scan of Bricks is shown in Figure 6.13.

The estimation is less accurate near depth discontinuities, but is sufficient for regions with smoothly varying depth. As the scene is scanned at higher resolutions, low accuracy regions of the map are refined.

One advantage of obtaining these maps is that the scan can subsequently be limited to the field of view of the image. An initialisation strategy that has worked well in practice is to define the scan limits to be well outside the field of view of the cameras, perform a scan at low resolution and then use the map to automatically determine the optimal scanning range.



(a) (b)

Figure 6.13: Predicted per-pixel scan angles for the Bricks scene, interpolated from known values (green points). These points were generated by scanning the scene at a 1.6 degree resolution. Several occluded points were identified and not included in the interpolation.

6.5.2 Identifying unmatched regions

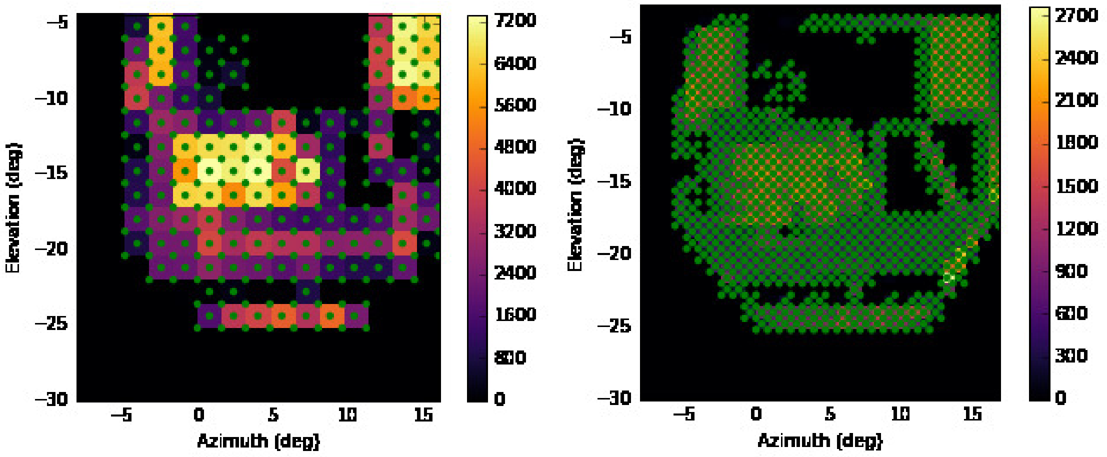
Once the disparity map is obtained, it must be analysed to determine where the LIDAR should be scanned next. For each pixel that is unmatched, the estimated scan angles (section 6.5.1) necessary to illuminate it with the LIDAR are added to a 2D histogram. This is effectively a transformation of the disparity map into the (binned) coordinate system of the scanner. The histogram bin size is determined by the desired scan resolution. A threshold is used to specify which bins represent unmatched regions of the image. At higher scan resolutions, it is possible to detect potentially occluded regions in the image by noting where no LIDAR points are detected. This reduces the possibility that occluded areas are repeatedly scanned when there is no way they can be matched. Experimentally a good resolution threshold for using an occlusion mask was ≤ 0.2 degrees. The mask is applied by multiplying a binary occlusion map (Chapter 3) element-wise with the disparity map before transforming into scanner coordinates.

For each bin that is labelled unmatched, the centre and corners are added to a list of points to be scanned. Two histograms for the Bricks scene are shown in Figure 6.14.

The points to be scanned during the next iteration are marked in green. Matched or occluded scan angles are shown in black and these areas are left unscanned on the next iteration.

6.5.3 Results

In this section, results are presented for the indoor scenes shown in Section 6.4. This approach was not performed for the outdoor data since it was assumed that there was no control over the scan



(a) (b)

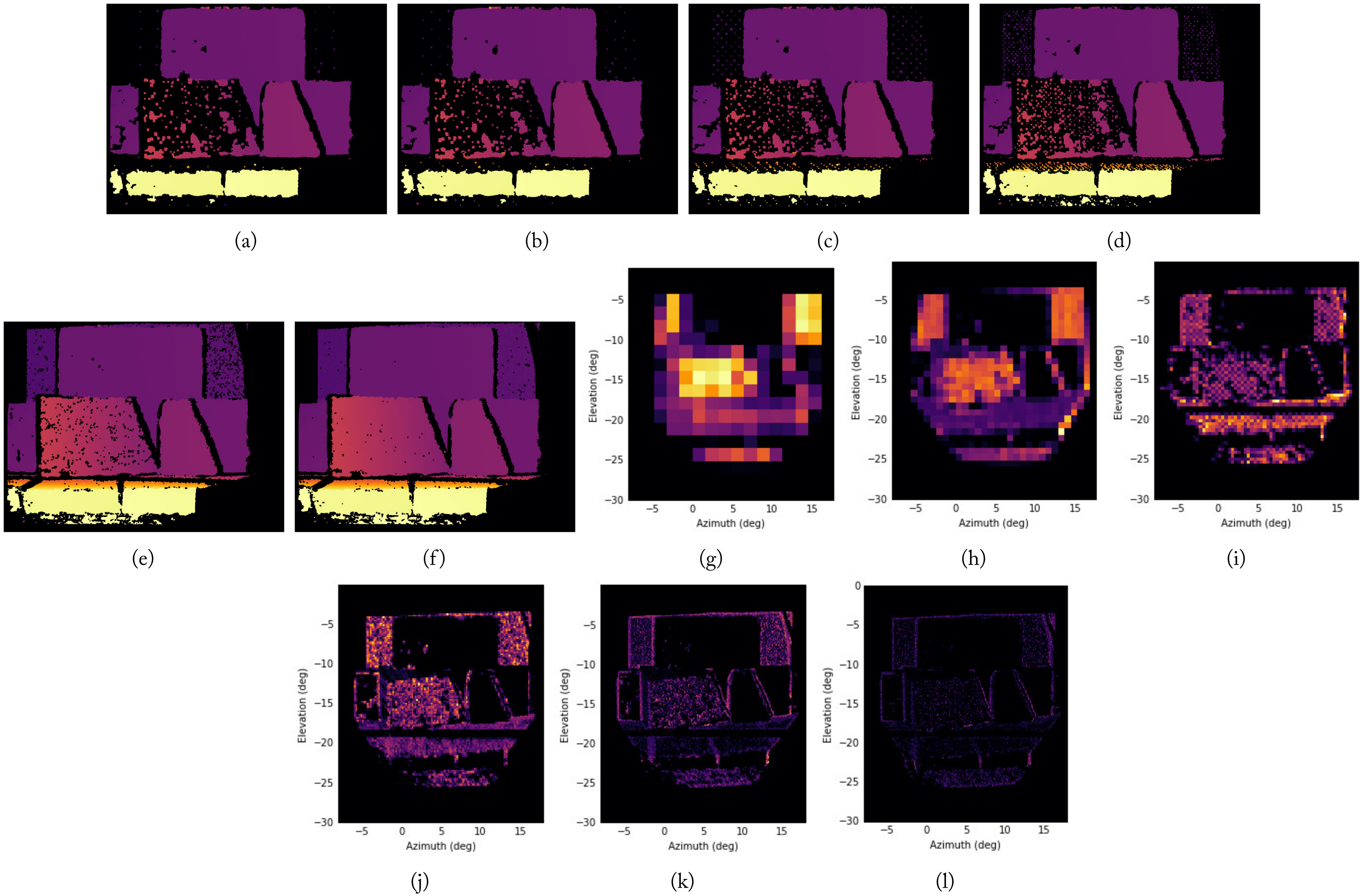
Figure 6.14: Histograms showing the locations of unmatched pixels in terms of the LIDAR scan angle. In each image, points to be scanned by the LIDAR on the next iteration are marked in green. (a) Bricks scene scanned at 1.6 degree resolution (b) Bricks scene scanned at 0.8 degree resolution.

pattern. For each scene, the intermediate scans and resulting disparity maps are shown, along with the estimated location of unmatched pixels in terms of the LIDAR scan angle. For the example scenes, a minimum scan resolution was specified which provided a limit on the number of points acquired and the number of iterations performed. The matching parameters used were the same as in the previous section. The top row of images in each figure shows the disparity map after each iteration of the scanning process. The lower row of images shows which regions of the image are unmatched, transformed into altitude-azimuth space. These images therefore highlight where the LIDAR should be scanned in the next iteration. Results are shown for Bricks, Chair, Corner, Biba and Potplant in Figures 6.15, 6.16,6.17, 6.18 and 6.19 respectively.

Table 6.2 shows the improvement (reduction) in the number of LIDAR points when using a progressive scan versus a full scan. For comparison, results from using progressive scanning are given with and without occlusion detection.

Table 6.3 shows the number of matched pixels using a progressive scan, compared to using a full scan. Results with and without occlusion detection are given.

Table 6.4 compares the accuracy of the disparity maps generated via progressive scanning (in the best case), compared to using a full set of LIDAR seed points.


(a) (b) (c) (d)

(e) (f) (g) (h) (i)

(j) (k) (l)

Figure 6.15: Gotcha match results for Bricks scene using SIFT keypoints and a progressive LIDAR scan pattern. Left to right (a-f, g-l): with 1.6°, 0.8°, 0.4°, 0.2°, 0.1° and 0.05° resolution respectively.

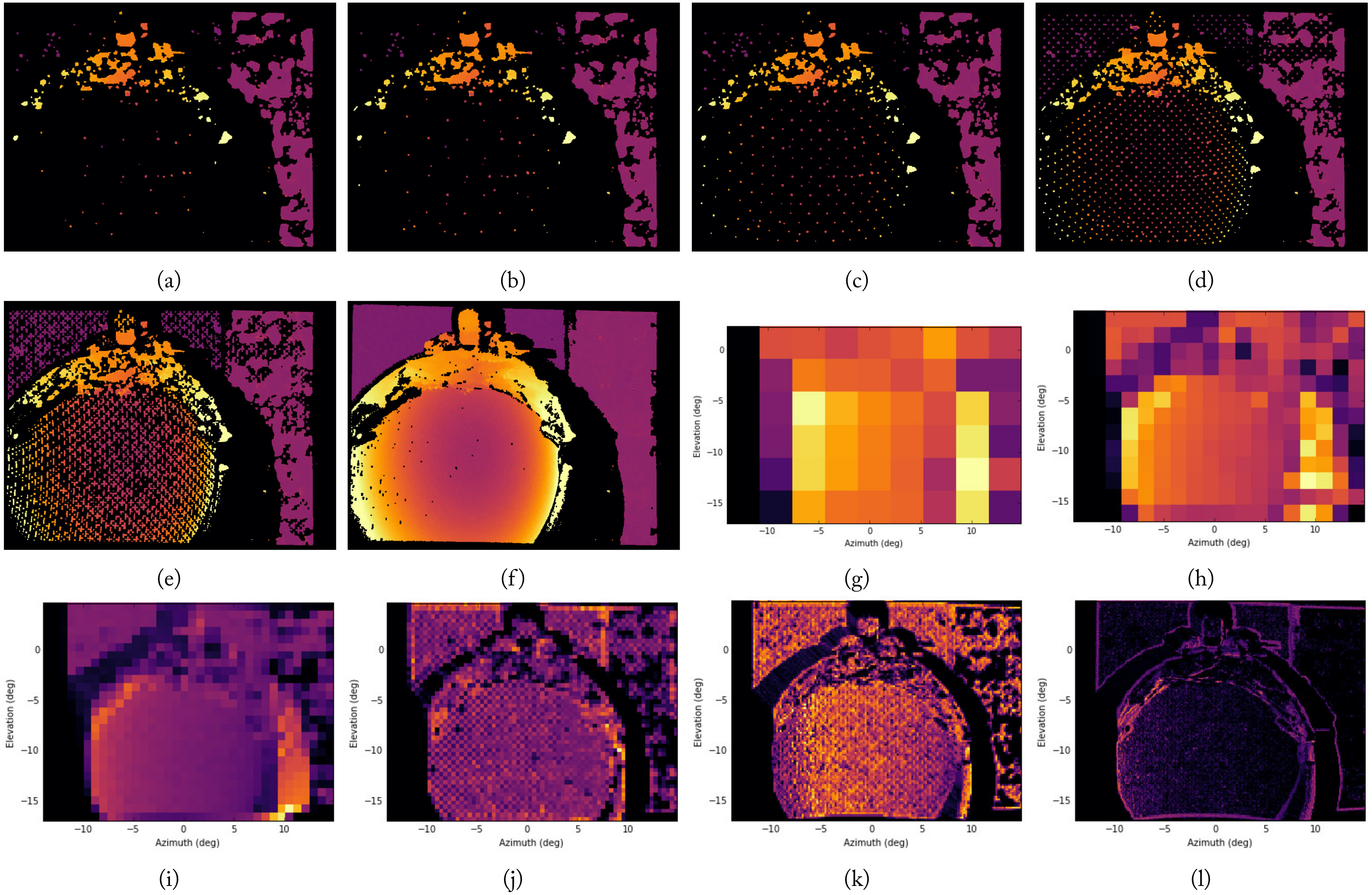


Figure 6.16: Gotcha match results for Chair scene using SIFT keypoints and a progressive LIDAR scan pattern. Top row (a-f): with 3.2°, 1.6°, 0.8°, 0.4°, 0.2° and 0.1° resolution respectively. Bottom row (g-l): unmatched pixels mapped to scanning angles.

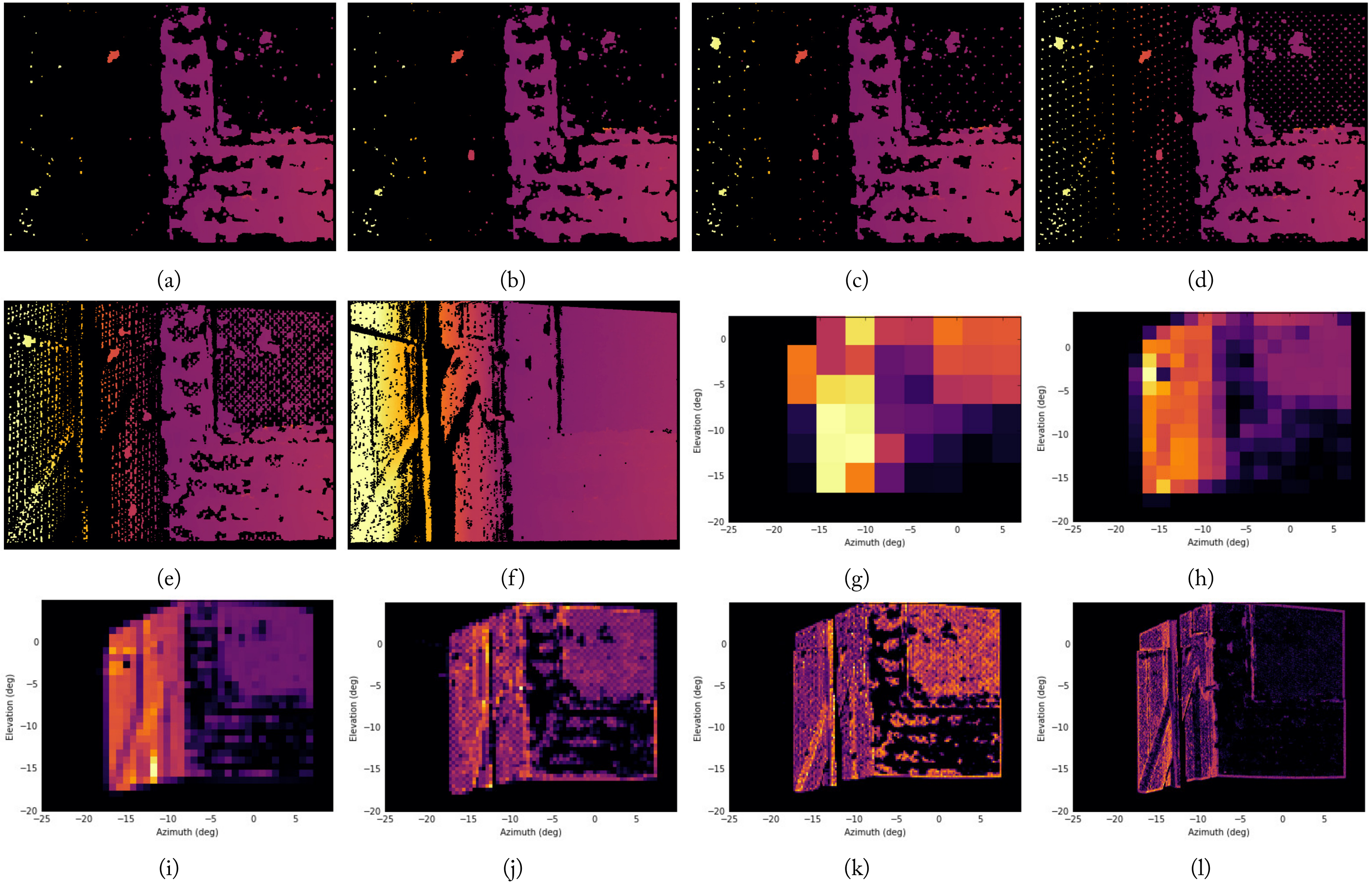
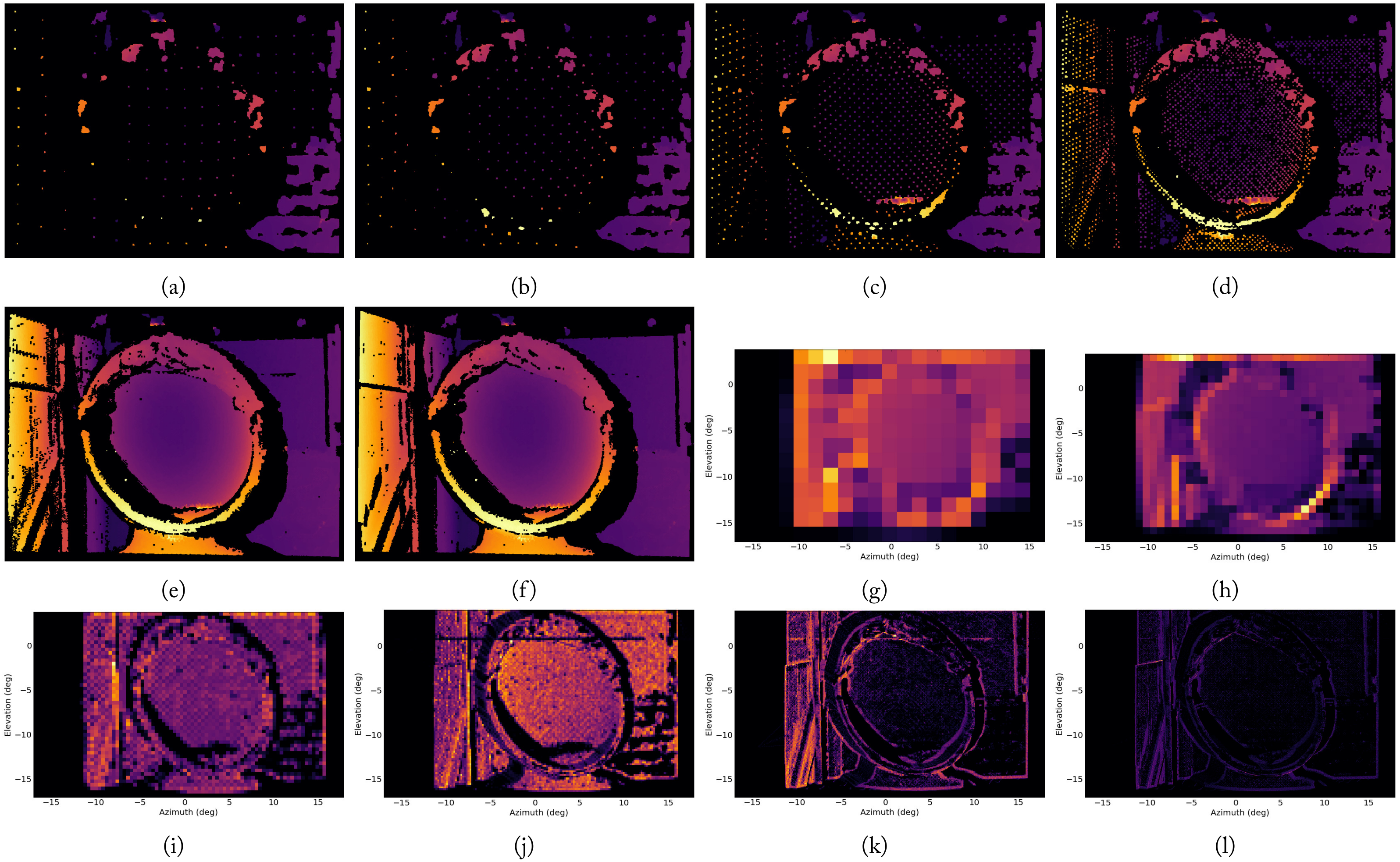


Figure 6.17: Gotcha match results for Corner scene using SIFT keypoints and a progressive LIDAR scan pattern. Top row (a-f): with 3.2°, 1.6°, 0.8°, 0.4°, 0.2° and 0.1° resolution respectively. Bottom row (g-l): unmatched pixels mapped to scanning angles.



(a) (b) (c) (d)

(e) (f) (g) (h)

(i) (j) (k) (l)

Figure 6.18: Gotcha match results for Biba scene using SIFT keypoints and a progressive LIDAR scan pattern. Top row (a-f): with 1.6°, 0.8°, 0.4°, 0.2°, and 0.05° resolution respectively. Bottom row (g-l): unmatched pixels mapped to scanning angles.

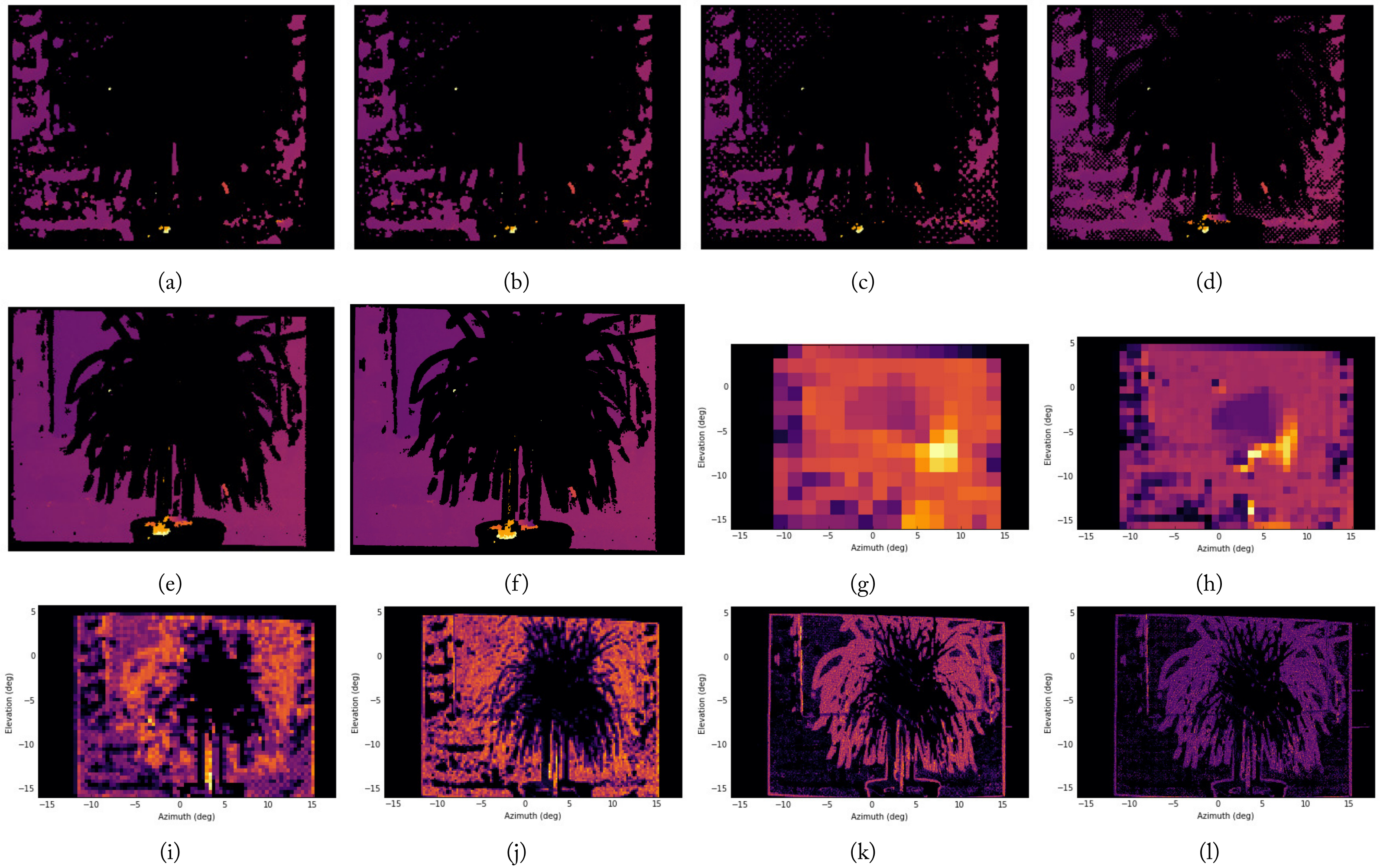


Figure 6.19: Gotcha match results for Potplant scene using SIFT keypoints and a progressive LIDAR scan pattern. Top row (a-f): with 1.6°, 0.8°, 0.4°, 0.2°, and 0.05° resolution respectively. Bottom row (g-l): unmatched pixels mapped to scanning angles.

Scene	Area (deg^2)	Res. (deg)	Full (pts)	Progr. (pts)	Progr.+Occl. (pts)
Bricks	777	0.05	310,800	28,116	21,417
Corner	868	0.1	86,800	25,673	24,019
Chair	632	0.1	63,200	31,838	26,910
Biba	736	0.05	294,400	50,694	39,133
PotPlant	735	0.05	294,000	72,884	50,144

Table 6.2: Number of LIDAR points used for disparity map enhancement in each test scene. Using a progressive scan significantly decreased the number of LIDAR points needed and further improvement was seen if occlusion detection is enabled.

Scene	Full (px)	Progr. (px)	Progr.+Occl. (px)
Bricks	778,688	727,772	732,933
Corner	761,204	756,205	749,130
Chair	723,823	706,156	701,743
Biba	729,214	631,215	637,623
PotPlant	465,698	402,651	406,431

Table 6.3: Number of matched pixels for progressive LIDAR scans vs full scans. Occlusion detection did not significantly change the number of matched pixels, but did reduce the number of LIDAR points needed (see Table 6.2).

Scene	Full accuracy (px)	Progr.+Occl. accuracy (px)
Bricks	0.35 ± 3.45	0.27 ± 4.32
Corner	1.27 ± 15.4	1.01 ± 19.62
Chair	0.53 ± 1.88	0.66 ± 2.06
Biba	0.70 ± 5.72	1.31 ± 9.00
PotPlant	0.40 ± 3.53	1.00 ± 7.6

Table 6.4: Stereo matching accuracy when using a full set of LIDAR seed points versus a reduced set of LIDAR seed points. Results are shown from the final output from the progressive scan. The same ground truth points were used to evaluate each method.

6.5.4 Discussion

In Section 6.4 it was shown that disparity map enhancement is possible by integrating a scanning LIDAR into a region growing stereo matcher. Using a simplistic approach, it was shown that increasing the number of LIDAR points used as seedpoints results in an asymptotic increase in the number of matched pixels until no further improvement in the disparity map is possible. However, those results did not take into account redundancy between the stereo matcher and the LIDAR scan and so many LIDAR measurements were unnecessary.

Using a progressive, coarse-to-fine scan, the number of LIDAR points required was reduced significantly. For scenes that contained regions of good texture, like Bricks, the number of LIDAR points was reduced by an order of magnitude (93%) with only a small (6%) reduction in the number of matched pixels. For images which were challenging to match with stereo alone, like Biba, more LIDAR points were required but there was still a large reduction (83%) compared to a full scan. The results also demonstrated that interpolating sparse LIDAR points to determine the scanning angle for a particular pixel is effective, as the final disparity maps were comparable to those using scans with many more points.

Occlusion detection allowed for additional robustness, as otherwise occluded pixels are repeatedly scanned. In the Potplant scene, this reduced the number of LIDAR points needed by almost a third. However, this scene was poorly matched overall. This is likely due to a number of factors such as the thin leaves, poor surface reflectance and perspective differences between the two image views. Obviously the reduction in the number of LIDAR points is only significant if there is a large amount of occlusion present in the scene. Applying occlusion detection to Corner, a largely un-occluded scene, only resulted in 7% fewer points. Since the estimated occlusion map is only accurate when the scene is scanned at high resolution, a suggested resolution threshold of 0.2 degrees was applied.

This technique was specifically developed in order to obtain faster scans using low-cost scanning systems. For a hypothetical system operating at 100 Hz, this would result in theoretical scan times of 3.5-10 minutes for all the indoor scenes presented. This speed is not suitable for realtime applications, but such a system could be used for static, high resolution inspection.

Comparing the accuracy of the two techniques, unsurprisingly using a full set of LIDAR seed points resulted in more accurate disparity maps. Given the reduction in the number of seed points, the disparity maps generated via an efficient scan were comparable: in the worst case, degraded by a factor of two (Potplant, from 3.53 px to 7.6 px). Whether or not this is an acceptable trade-off (scanning time vs accuracy) depends on the scene.

Commercially available 3D scanning systems usually offer control over scan resolution, but do not allow any control over where the beam is steered. This precludes any significant reduction in LIDAR points, due to redundancy, unless the scans at different resolutions are offset spatially. Using a coarse-to-fine approach might still reduce the scan time significantly since there is a possibility that a scan at the maximum resolution is not necessary.

6.6 Conclusions

In this chapter it was argued that passive stereo matching alone is not sufficient for robust matching of many typical indoor and outdoor environments. A data fusion method was proposed that integrates a scanning LIDAR into a region growing stereo matcher (Gotcha). Any type of LIDAR might be used, provided it is cross-calibrated with the cameras. The region growing process allows disparity map densification in low-texture regions.

Results were presented from challenging indoor scenes as well as a series of outdoor scenes taken from an autonomous driving dataset. Asymptotic improvement was seen in the disparity map as more LIDAR seed points were used, suggesting that no further improvement in the disparity map was possible. Including results on outdoor data, taken from the KITTI dataset, showed that Gotcha using LIDAR seeds outperformed both standard Gotcha and SGM in terms of the number of pixels matched with comparable accuracy.

Finally an efficient scanning algorithm was presented that employed a coarse-to-fine scan pattern. This method aimed to reduce redundancy where image regions were already matched and did not need to be scanned with the LIDAR. A method for estimating the scan angle required to illuminate a particular pixel with the LIDAR was given. By scanning the scene selectively in unmatched regions, it was shown that the number of LIDAR seed points required can be reduced by up to an order of magnitude.

Robust stereo matching using LIDAR-derived texture 7

Dense stereo matching is dependent on there being unique intensity variations (texture) in the images. Texture projection is an established method for enabling robust stereo matching of featureless images, but current methods are limited to short ranges, are untested outdoors, and are relatively inflexible to changes in the stereo system setup.

Chapter 6 introduced the idea of using data from a LIDAR to seed a region growing stereo matcher to improve performance in low-texture regions. This chapter extends this idea by proposing that the LIDAR be used to generate texture in images, since each step in the scan can be used to paint a dot in the stereo images. With many such images, a random pattern may be overlaid on the original stereo pair. Methods for pattern generation using both visible and NIR LIDAR systems are given.

Dense matching results from a number of indoor and outdoor scenes are given, using illumination patterns derived from directly imaging the LIDAR spot and from simulating images of the LIDAR spot. The number of dots required and simulated dot size are investigated. Finally, a method is given for intelligently texturing an image based on predicting which parts of an image will be matched. Intelligent texturing is shown to significantly reduce the number of LIDAR points required in a scan.

7.1 Introduction

The human vision system relies on textural cues to determine which parts of two images correspond (Marr et al., 1991). Yet, there is no accepted definition of image texture. A very broad definition of texture is that it describes the variation of intensity values in an image or image region. Although it is straightforward to describe textures using natural language notions such as colour, roughness and so on, it is comparatively difficult to express these concepts mathematically. A variety of different image textures are shown in Figure 7.1.

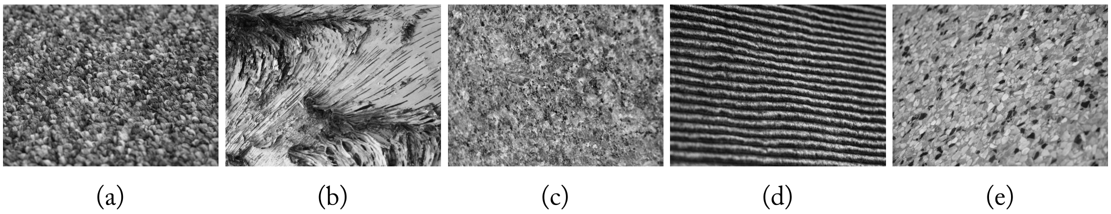


Figure 7.1: Examples of different textures, taken from the database provided by (Lazebnik et al., 2006) Left to right: Carpet, tree bark, marble, corduroy fabric, stone flooring.

Previous work pn texture analysis has focused on object identification, particularly in medical imaging, and scene segmentation. A common approach is to calculate textural features and apply some kind of clustering to determine which pixels correspond to the same object (Ojala et al., 1996). A textural feature or metric aims to convert a series of intensity values, ordered or un-ordered into a single scalar value (Tamura et al., 1978).

Texture is highly relevant for stereo matching, since all matching algorithms rely on comparing intensity variation between image regions. Image regions with low intensity variation are challenging to match because they introduce ambiguous matching costs. The problem is not limited to uniform texture, as repetitive texture can also introduce ambiguities. The low-texture problem is illustrated in Figure 7.2.

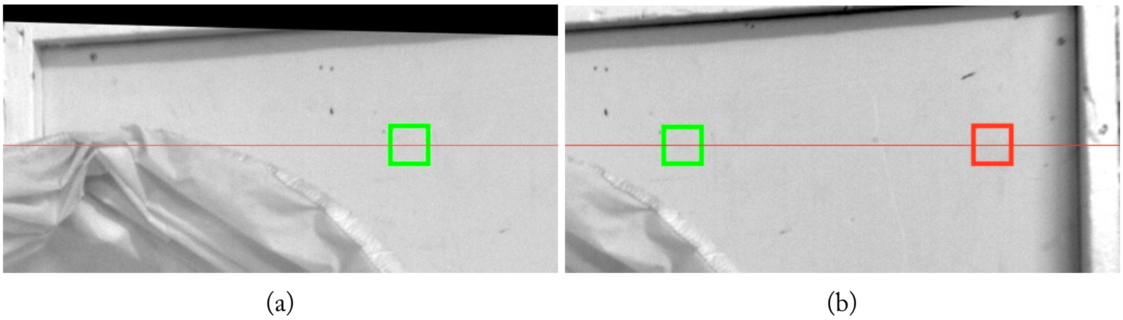
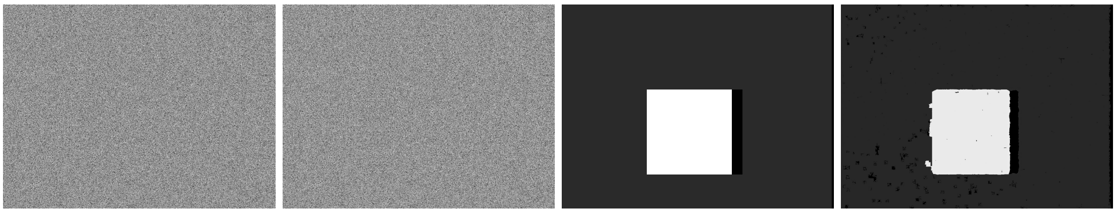


Figure 7.2: Crop from Biba scene. An epipolar line is marked in red. The green squares show the correct correspondence and the red square in the right image shows an incorrect correspondence. The matching costs for the two regions in the right image are very similar.

If local correlation were used to match this image, it would be difficult to determine which of the two marked image regions on the right correspond to the region on the left. More robust algorithms like dynamic programming and global/semi-global matching offer an improvement over naive correlation, they do not guarantee good match performance on all scenes. Other algorithms have been developed to cope with low texture regions, for example Sach et al. (2009) used edge detection to locate textureless regions and guide the cost aggregation step during matching.

It is possible to construct a stereo pair that is close to ideal for stereo matching. Here, ideal is taken to mean that any local region in the left image has an unambiguous match in the right image. Random dot stereograms approximately satisfy this definition (Julesz, 1960) and an example is shown in Figure 7.3.



(a) (b) (c) (d)

FIGURE 7.3: Random dot stereogram. (a) Left image (b) right image (c) ground truth (d) Gotcha match result

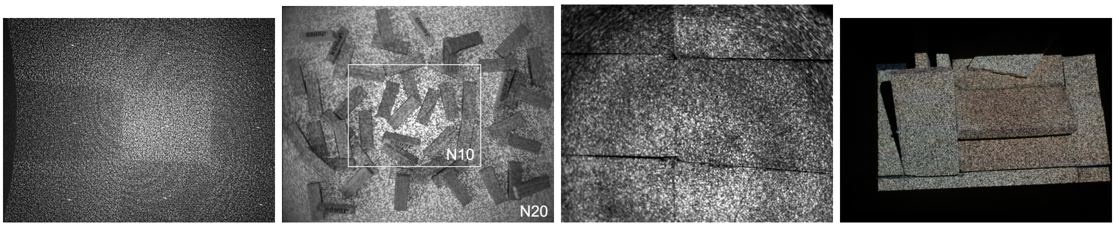
The left and right images are created by filling two arrays with identical random noise. The right image is a copy of the left image, shifted by 10 px. The central block is shifted by 60 px. The space left by moving the block is filled with random values. The Gotcha match result is very similar to the ground truth with only a few failed regions near the depth boundary. The stereo match result is not perfect (as shown in the lower left of Figure 7.3d), likely due to self-similarity of the random pattern.

If such a pattern is visibly projected onto a scene, it enables robust matching in textureless regions. Usually the pattern is a field of (pseudo) randomly located dots. This type of pattern can be generated using a number of techniques such as a diffractive optical element (O'Shea et al., 2003), data projector or from interference phenomena like laser speckle. This approach is exploited by the Microsoft Kinect v1 (see Chapter 2) to provide robust depth sensing, although it is not a stereo system.

Recently several manufacturers, such as IDS (Obersulm, Germany), have introduced 'active' stereo imaging systems with integrated illumination systems under the Ensenso brand[1]. Osela (Lachine, Canada)[2] manufacture diffractive optical elements with up to 100000 random dots. Some examples of texture projection techniques are shown in Figure 7.4.

[1] `https://en.ids-imaging.com/ensenso-stereo-3d-camera.html`, accessed 20/9/2016

[2] `http://www.osela.com/products/random-pattern-projector/`, accessed 20/9/2016



(a) (b) (c) (d)

Figure 7.4: Examples of texture projection methods: (a) Diffractive optical element (Kinect), (b) IDS Ensenso fixed 'gobo' pattern (c) laser speckle (d) random dot pattern from a data projector.

One issue with texture projection involving truly random patterns is the possibility that if two small patches of the pattern are examined, they will be similar or identical. Molinier et al. (2008) construct their pattern from 5x5 pixel blocks in such a way that all blocks are unique. Lim (2009) used non-recurring de Brujin sequences (de Brujin, 1975) to generate patterns which have no repetition along individual epipolar lines. A de Brujin sequence B = (k, n) contains all sequences of length n drawn from an alphabet, A, with length k, exactly once. For example if A = 0, 1 a valid sequence B(2, 3) is 00010111.

Konolige (2010) presented a technique for generating 'ideal' patterns for block matching algorithms also using De Bruijn sequences, but with an additional optimisation step. After an initial pattern is generated, simulated annealing is used to adjust it such that the average similarity between any two blocks is minimised. The process also aims to generate patterns that are less affected by blurring and phase noise introduced by spatial offsets between the camera and projector. These patterns exploit the fact that if the epipolar constraint is satisfied, and the images are rectified such that matches lie along horizontal scanlines, then the pattern need only be unique in the horizontal direction.

These studies quantified performance in different ways. Lim compared results to a ground truth disparity map, but it was not specified how the ground truth was generated. Konolige compared the drop-out (the percentage of unmatched pixels) for various patterns and also performed metric accuracy measurements on the final pattern using a planar target.

7.1.1 Examples of projection methods

For the Bricks scene, stereo images with other pattern projection methods were acquired for comparison. The methods used were the Kinect (IR) projector, a data projector displaying a binary white noise image and laser speckle generated using a 532 nm laser and a 400 μm fibre.

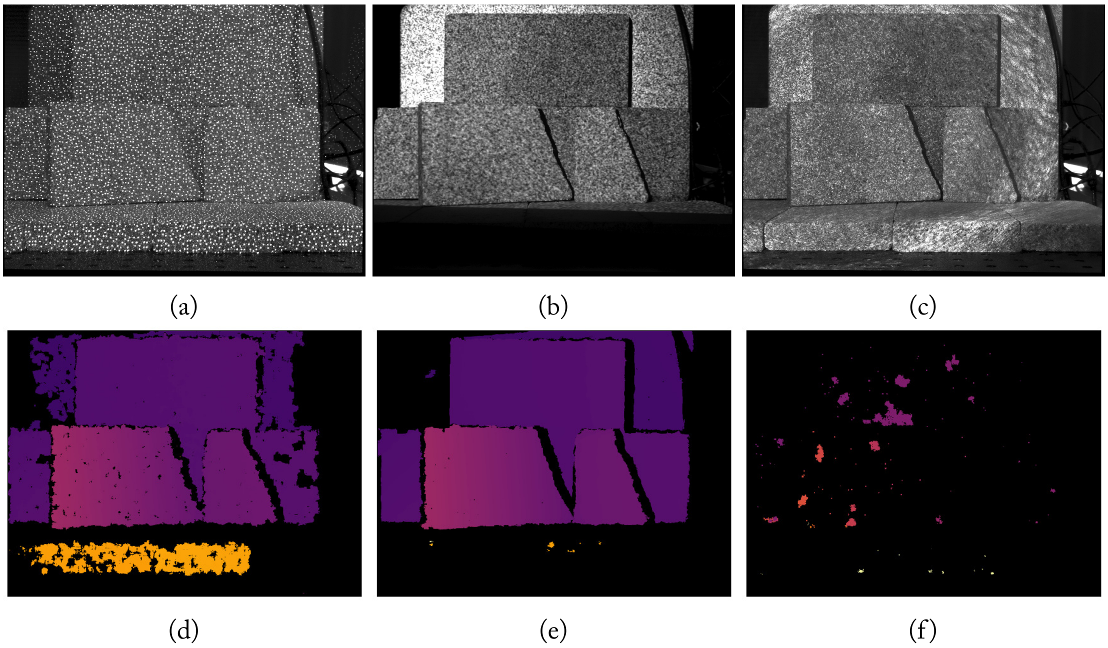


Figure 7.5: Bricks scene using other texture projection methods: (a) Kinect pattern (b) data projector displaying binary white noise (c) laser speckle. Gotcha match result from (d) Kinect, (e) data projector, (f) laser speckle.

Both the Kinect and projector patterns resulted in dense matching in most illuminated, unoccluded regions. The data projector produced the smoothest disparity map. However, due to the specular reflection in the upper left of the image, this region failed to match. Note the projector did not fully cover the field of view of the cameras. The results from speckle projection was poor, but this is largely due to the difficulty of generating suitable SIFT-derived seed points for Gotcha. The exact reason for this is unknown, but the authors suspect that it is due to local saturation of the image by the speckle pattern and interference effects causing different textures to be visible to each camera. The regions that were matched using laser speckle were consistent with ground truth.

7.1.2 Limitations of current methods

The usable range of any projection system is limited by the signal to noise of the projected pattern, imaged by the camera. Laser-based projection methods offer superior range compared to LED- or lamp-based illumination. Since texture projection relies on the pattern being visible to the camera, the projection system must be able to illuminate the scene sufficiently at any required distance. Commercial texture projection systems like the Ensenso are limited to a modest range of up to 3m.

The pattern projector must be customised for the camera system, since it must illuminate the same field of view. Any changes to the camera system necessitate changes to the pattern.

Outdoor operation in sunny conditions is usually not possible, particularly for systems which use NIR illumination. A visible blocking filter can be used to improve SNR at the expense of being able to image the scene in true colour. The Kinect avoids this by having a separate (IR) camera for depth measurements. However, although this improves SNR for the projected pattern, in sunny conditions the sensor is easily overwhelmed by background light.

7.2 LIDAR Random Dot Projection

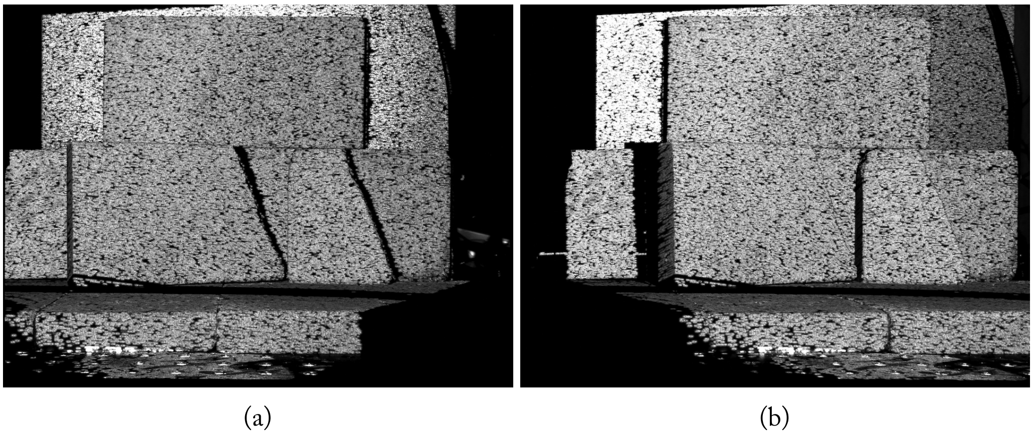
This section describes how a LIDAR system can be used to generate texture in an image. It is shown that LIDAR can overcome the limitations of conventional texture projection systems. Two strategies are described: first, a method where the random dot image is generated by imaging the LIDAR beam during a scan and second, a method where the random dots are 'painted' on the digital image after a scan, given a cross-calibration between the LIDAR and stereo camera system. Using a LIDAR rather than a static pattern allows for additional robustness since any unmatched regions can be progressively scanned using the LIDAR until the scene is satisfactorily matched.

Scanning LIDAR has the potential to overcome both SNR and pattern resolution limitations. LIDAR is able to operate over longer distances than projector-based methods. Since LIDAR measurements are 1D, the laser power is concentrated into a single spot. Visible spectrum systems are easily observed using cameras even in bright conditions with the use of very short exposure times. Most LIDAR scanning systems have a wide field of view so the choice of lens does not affect the ability to generate a pattern. Finally, LIDAR is able to operate robustly outdoors.

The data fusion approach described in Chapter 6 is also applicable when using a LIDAR to generate random texture. Specifically each LIDAR point can be passed as a seed for the Gotcha region growing process. If additional steps are taken (Section 7.3) to predict which parts of the image region will be matched using stereo alone, the LIDAR can be selectively scanned only in textureless regions. Overall this means that, depending on the amount of texture present in the scene, only a sparse LIDAR scan is required for dense, robust matching.

7.2.1 Random dot image generation

In order to overlay a random dot pattern on the scene using a visible LIDAR, multiple images of the LIDAR spot can be stacked together. At each point in the scan, a small region of the image is illuminated using the LIDAR. If the scan covers the entire image, then combining the intermediate images is equivalent to a single image where the whole scene is illuminated simultaneously. This requires synchronisation of the cameras with the LIDAR, as discussed in Chapter 4. An example of this is shown in Figure 7.6 for the Bricks scene.



(a) (b)

Figure 7.6: LIDAR-illuminated Bricks scene formed from a stack of 50000 images, each image corresponding to a single LIDAR measurement. Due to the very short exposure time (1/1000 s), ambient illumination is suppressed.

The stacking process is performed by first creating an empty image, I_s. For each acquired image, I_n, an element-wise maximum is performed between the image stack and the new image. That is, the stack image is defined as a recurrence relation:

$$I_{s,n}(u,v) = \max\left(I_{s,n-1}(u,v), {}_{n}(u,v)\right) \tag{7.1}$$

for n scan points where $I_{s,0}$ is an empty image and (u,v) are the pixel locations in the images. The maximum operation is preferred over simply summing the images as it prevents background noise from building up and saturating pixels over the several thousand images required to form the stack.

Either a full scan, containing measurements at all possible scan angles for a particular resolution, can be acquired or scan angles can be randomly skipped. In the case of a full scan, a random sample of points can be taken to generate the random dot image.

The resulting pattern can be matched as-is, but can also be merged with the ambient light image to make use of existing intensity variation. This is shown in Figure 7.7.

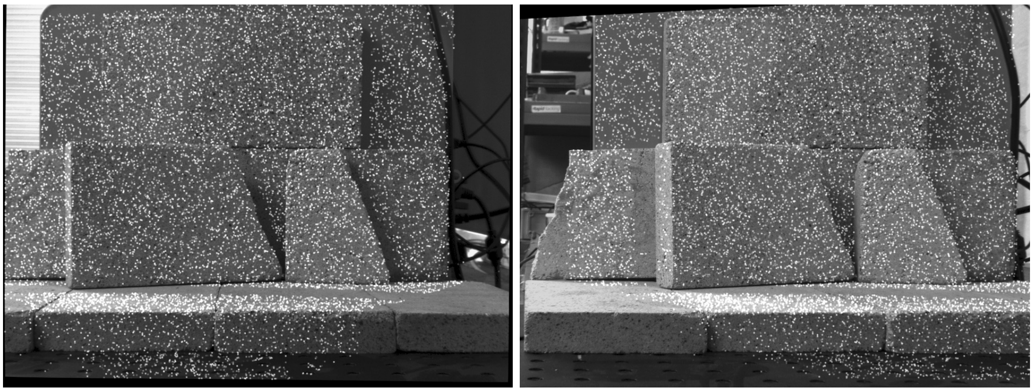

(a) (b)

Figure 7.7: Bricks scene ambient light stereo pair with a LIDAR-derived random dot pattern overlaid. There is some shadowing visible in the lower row of bricks near the centre of the image.

To avoid saturation of the original 8-bit images, both pattern image and ambient light image are cast as 16-bit arrays before being combined. The intensity of the combined image is then rescaled to 8-bit and histogram equalisation is performed to retain contrast in darker areas not illuminated by the pattern.

7.2.2 Simulated LIDAR spot images

In some cases it is not possible, or practical, to directly image the laser spot. Synchronisation with a camera limits the LIDAR acquisition speed to the camera frame rate which could lead to excessively long scan times. LIDAR systems which are very quickly scanned (or offer no control over the scan pattern) are also not suitable for direct imaging. Alternatively scanning LIDAR using IR lasers require cameras sensitive to the NIR which can degrade image quality due to unwanted additional (invisible) illumination.

In this section, it is suggested how LIDAR random dot images may be simulated. The aim is to paint texture into an image using only the 3D measurements provided by the LIDAR. If the LIDAR and stereo system are fully (intrinsically and extrinsically) cross-calibrated, then it is straightforward to map 3D measurements from the LIDAR into 2D locations on the sensor planes. This procedure is described in Chapter 5 and was used to generate seed points from KITTI LIDAR data (Section 6.4.2). Along with the location of the LIDAR spot, the visual representation of the spot also should be simulated.

The intensity profile of a LIDAR spot is approximately gaussian, which can be verified by examining individual images acquired synchronously during a scan (Chapter 3). Therefore an appropriate method for simulating a LIDAR random dot pattern is to paint a gaussian intensity pattern onto an image at each point the LIDAR should illuminate. Analogously to the previous section, a simulated pattern may be expressed as a sum over 2D gaussians:

$$I_n = I_s + \sum_{i=0}^{n} (G_i) \tag{7.2}$$

Where I_n is the final pattern intensity at (u, v), I_s is an initial image and G_i is a 2D Gaussian function centred on the ith spot.

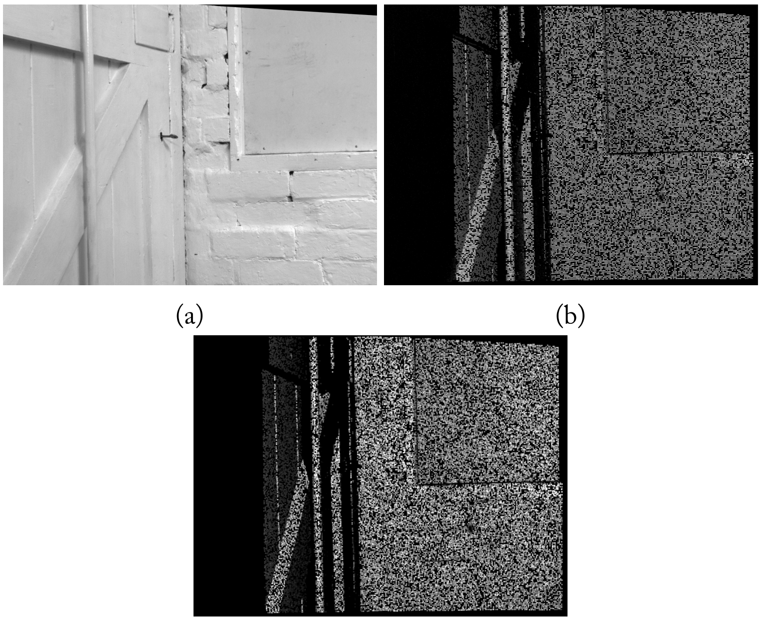
The peak intensity of the LIDAR spot is dependent on the reflectance of the surface at the wavelength of the laser, for a fixed exposure time. Some LIDAR systems allow measurement of the return signal power which can be used to scale the intensity. Additionally in real images, the laser spot may be distorted due to the angle of incidence between the surface and the laser beam. This cannot be emulated in simulated imagery without *a priori* knowledge of the scene's 3D structure. Nevertheless, a comparison between a real and simulated image, Figure 7.8, shows that despite these limitations the pattern looks very similar. The following section discusses the effect of spot diameter.

A Gaussian spot pattern was chosen to mimic LIDAR, however in principle any pattern (suitably localised) could be painted in the image. One option could be to paint locally unique textures to reduce the risk of ambiguous matching. This is less of an issue for Gotcha, since the pattern points may be used as seeds which inherently avoids initial ambiguity when matching.

7.2.3 Effect of spot diameter

The minimum physical diameter of the LIDAR spot varies with distance and is dependent on the divergence of the laser beam. However, the area covered by each pixel also increases with distance and the imaged spot diameter is dependent on surface reflectance, exposure time and instantaneous laser power. Even if the spot diameter is large, an arbitrarily small exposure time can be used so that only the peak intensity is visible (or detected above a threshold).

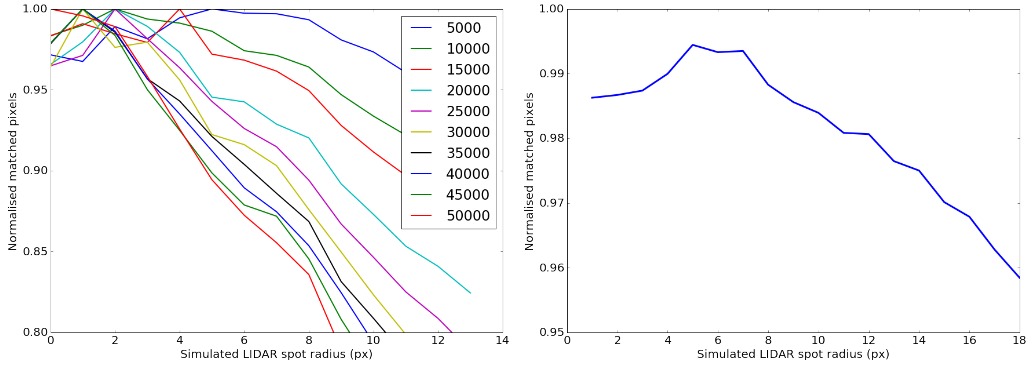
Figure 7.9 shows the change in the number of matched pixels as a result of changing the simulated spot radius for indoor and outdoor imagery.



(a) (b)

(c)

Figure 7.8: Real versus simulated LIDAR-derived random dot images. (a) Left ambient light image (b) Stack of 28000 LIDAR spot images, (c) simulated image of the same pattern using 5 px diameter spots.



(a) (b)

Figure 7.9: Relationship between simulated LIDAR spot diameter and the relative number of pixels matched. (a) Results using indoor scenes with varying number of LIDAR spots (b) Average outdoor results (KITTI imagery) with around 20 k spots per image.

In both indoor and outdoor imagery, there was a degradation in match performance with larger spot diameters, above 5 px. This is probably caused by overlapping patterns as the spot density increases. Below this threshold the best performance was observed with a spot diameters of 2 px for indoor imagery and 4-5 px for outdoor imagery. This suggests that the choice of diameter is scene specific and if a system is used to repeatedly acquire imagery of the same scene type, some experimentation to determine the best spot size is desirable.

7.2.3.1 Number of LIDAR points required

A statistical analysis of the number required of LIDAR points needed is challenging since it depends on a lot of factors that are scene-specific. For an image to be matched, there must be sufficient textural information present in a particular pair of correlation windows. Not every pixel in the windows needs to be directly illuminated by the LIDAR.

Consider an image of a fronto-parallel plane, uniformly illuminated using only a LIDAR. The probability that a pixel is (partially) illuminated is:

$$p_i = \frac{N\pi r_s^2}{A} \tag{7.3}$$

assuming that r_s is the spot radius, A is the image area and N is the number of LIDAR points. For $N = 10000$, $r_s = 2$ px and $A = 1.3$M px, $p_i \approx 10\%$. The probability of illumination is strongly dependent on the LIDAR spot size. Increasing r_s to 3 px more than doubles the probability of illumination. In reality, the spot is often ellipsoidal rather than circular, there are occlusions and the LIDAR might not scan the entire image. This approximation also breaks down when spots begin to overlap, but nevertheless serves as a good indication of how dense a pattern will be.

The illumination probability also suggests that a relatively small number of LIDAR points are required. It is possible to calculate the number of points required such that all pixels should theoretically be illuminated. For the 1.3M px example above, this occurs at around 100 k points. Since a projected pattern should have both dark and light regions, fewer points are required and this is investigated empirically in the next section.

7.2.4 Matching results

Results are presented from some of the scenes introduced in Chapter 6. Images from every point in the LIDAR scan were acquired for Bricks, Chair, Corner allowing a comparison between real and simulated random patterns.

For each scene, match results were obtained using different numbers of randomly sampled LIDAR points. Matching was performed using Gotcha using default settings (Section 4.5) and the only seed points used were derived from the sampled LIDAR points. Each random dot pattern was generated sequentially such that each contained all the points from the previous patterns. Simulated dot patterns were generated using a spot size of 2 px.

Matching is shown for real LIDAR random dot (RLRD), simulated LIDAR random dot (SLRD) and with/without combining the ambient light image. Figures 7.10, 7.11 and 7.12 show the number of matched pixels for Bricks, Chair and Corner respectively. Disparity maps are shown with various numbers of LIDAR points used in Figures 7.13, 7.14 and 7.15 for Bricks, Chair and Corner respectively.

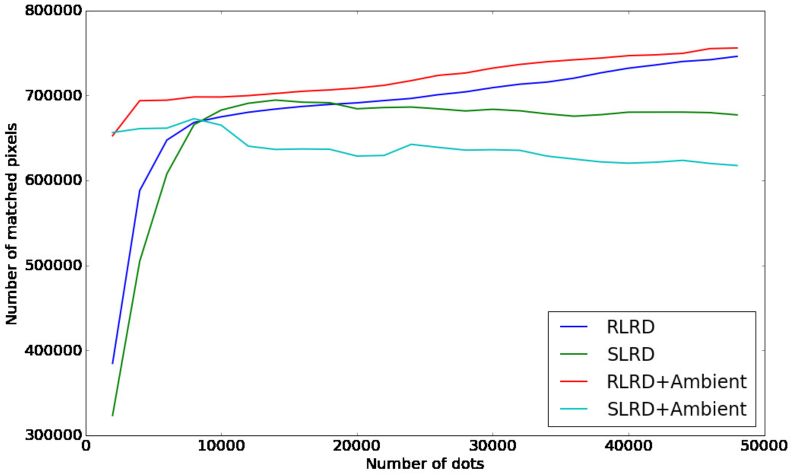


Figure 7.10: Bricks scene match results using random patterns with different numbers of dots.

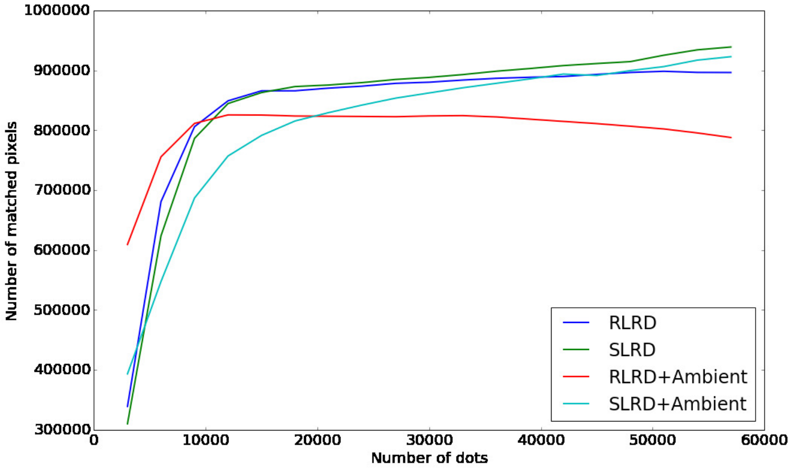


Figure 7.11: Chair scene match results using random patterns with different numbers of dots.

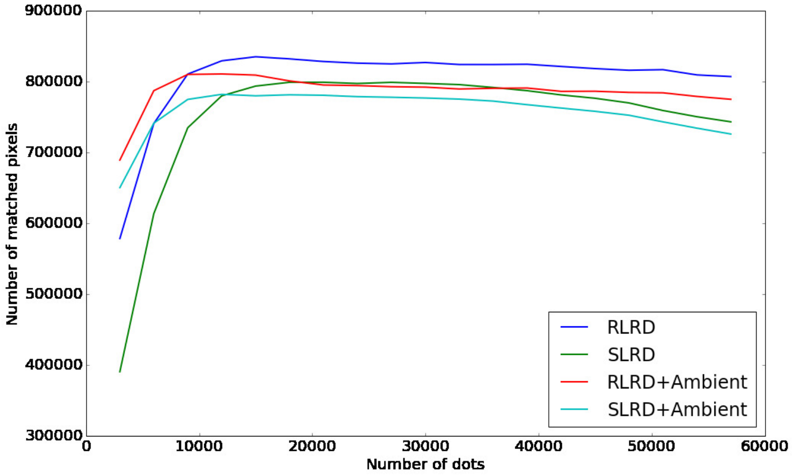


Figure 7.12: Corner scene match results using random patterns with different numbers of dots.

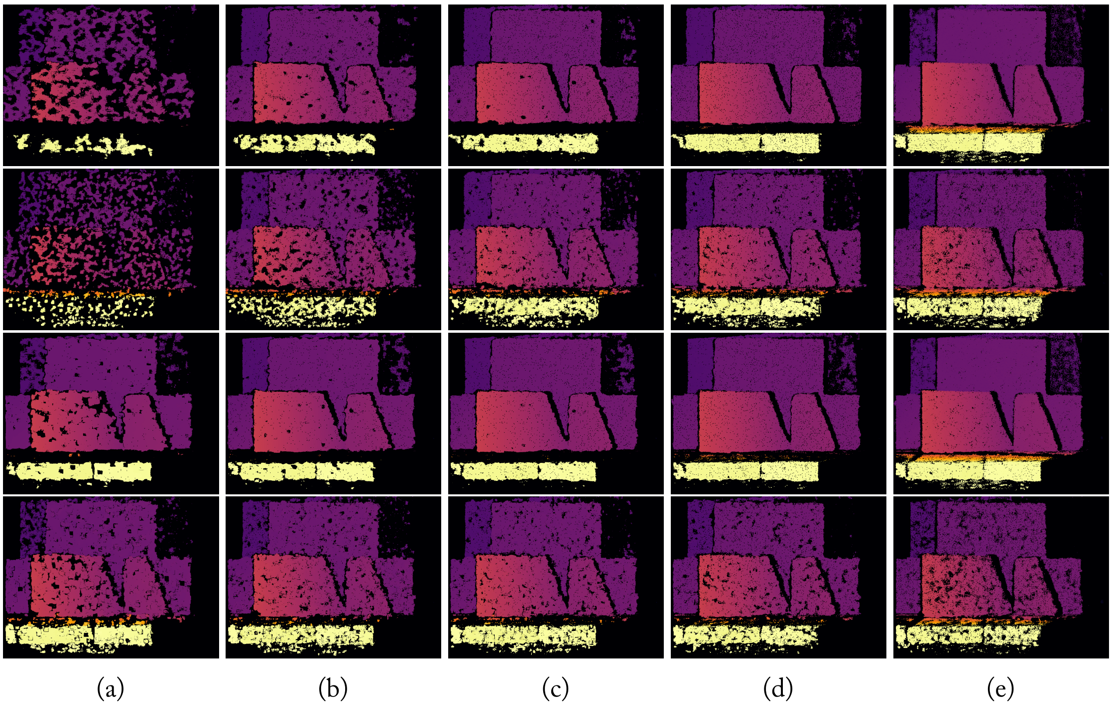


Figure 7.13: Bricks scene matched using Gotcha with various random dot patterns. From top to bottom, RLRD, SLRD, RLRD+Ambient, SLRD+Ambient. From left to right (a-e), 2000, 6000, 10000, 20000, 50000 LIDAR points.

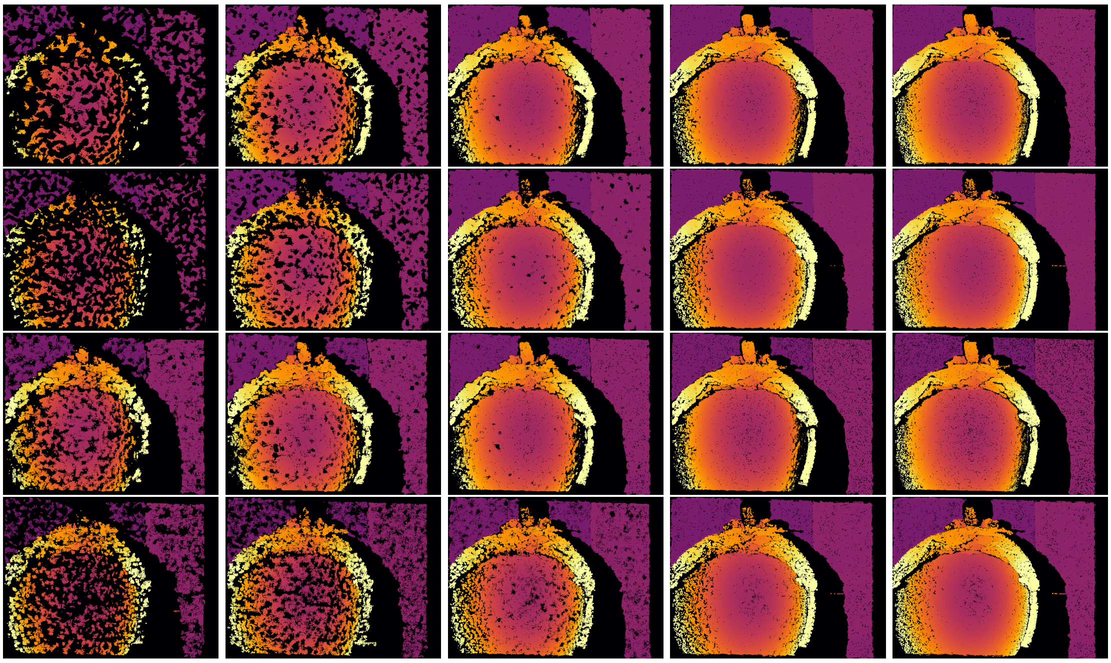


Figure 7.14: Chair scene matched using Gotcha with various random dot patterns. From top to bottom, RLRD, SLRD, RLRD+Ambient, SLRD+Ambient. From left to right (a-e), 3000, 6000, 12000, 24000, 36000 LIDAR points.

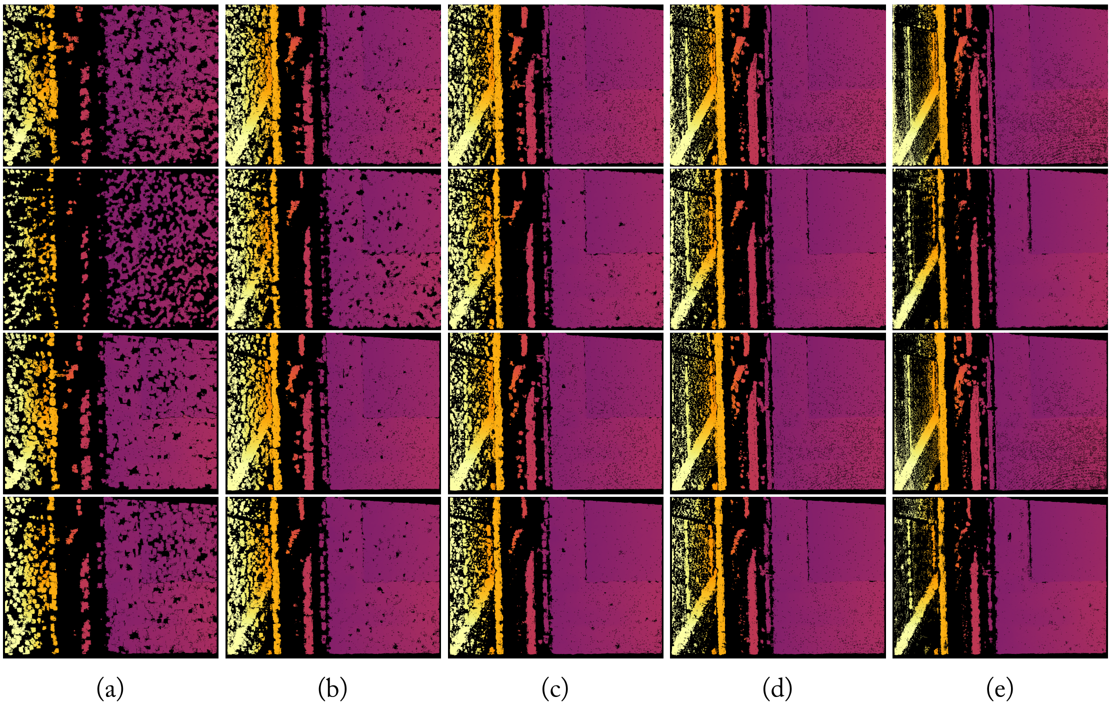
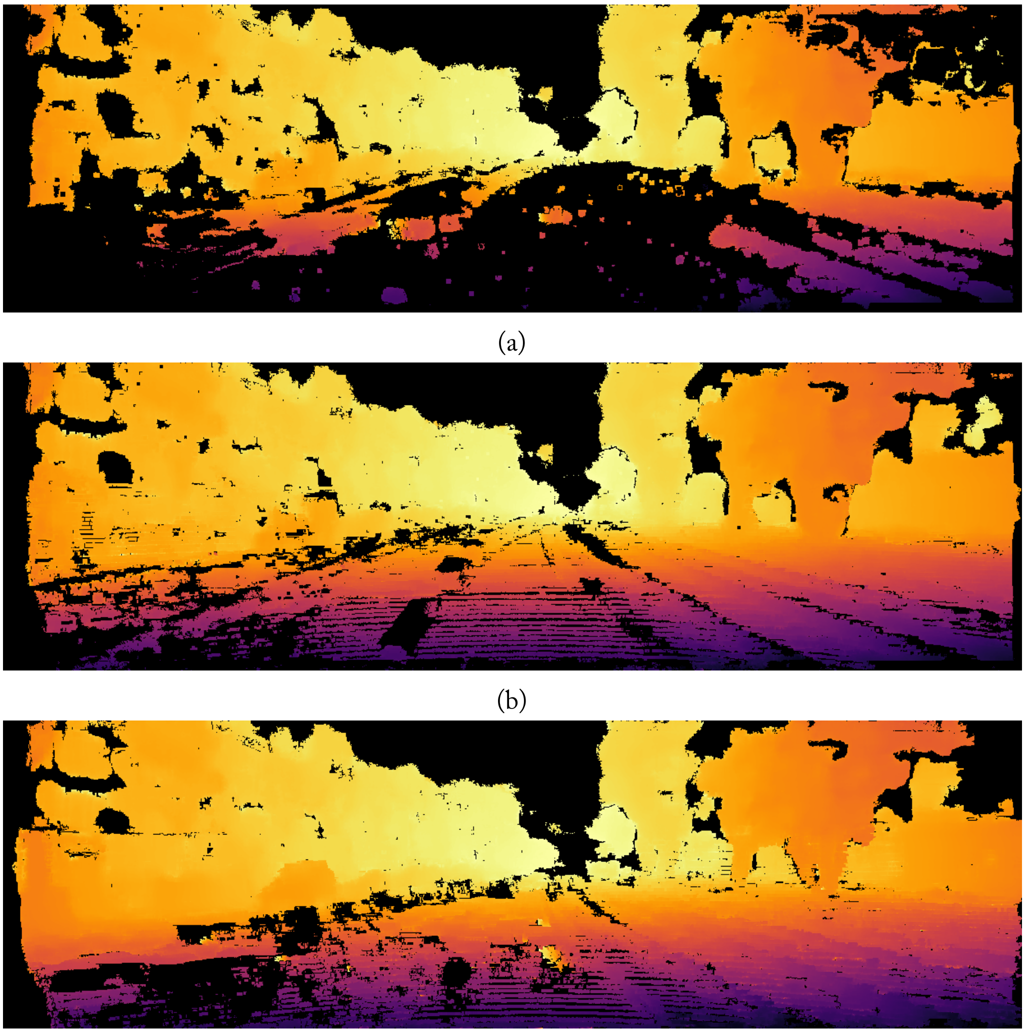


Figure 7.15: Corner scene matched using Gotcha with various random dot patterns. From top to bottom, RLRD, SLRD, RLRD+Ambient, SLRD+Ambient. From left to right (a-e), 2000, 6000, 10000, 20000, 40000 LIDAR points.

7.2.4.1 Outdoor data (KITTI)

As the KITTI data provides a fixed number of LIDAR points per stereo pair, all were used for random pattern generation and disparity map seeding. This data represents both a real-world application of the LIDAR random dot technique and a demonstration of the efficacy of simulated patterns when an invisible laser is used. Also, this demonstrates that the technique may be applied to existing systems and previously acquired data. A comparison between the match result for ambient illumination (see Chapter 6) and SLRD combined with ambient illumination is shown in Figure 7.16.



(a)

(b)

(c)

Figure 7.16: (a) KITTI Disparity map from ambient illumination only, using Gotcha with SIFT seeds. (b) from ambient illumination only, using Gotcha with SIFT and LIDAR seeds. (c) from ambient illumination and SLRD, using Gotcha with LIDAR seeds. A spot diameter of 6 px was used for this result.

7.2.5 Discussion

For all scenes, the addition of random texture significantly improved the match results. The best match results were observed when RLRD patterns were combined with ambient light imagery. This is unsurprising since these were the most realistic representations of the scenes and the most texture was available for matching. SLRD patterns provided considerably better match results than ambient light alone and were largely comparable to the results using RLRD. However as SLRD does not fully capture scene geometry and does not allow for spot distortion due to perspective effects, this may explain the poorer results seen in Corner and Bricks.

In outdoor imagery, the results were largely similar to when ambient imagery is used with LIDAR seed points. In Figure 7.16c there was a clear improvement in the left hand region of the image, although some regions of the road were incorrectly matched.

Similarly to the results in Chapter 6, the number of matched pixels was found to increase asymptotically with the number of LIDAR points. However in some cases, such as Corner, worse results were seen as the number of points increased, even when using real imagery. In this scene, degradation is visible in the disparity map on the left-hand wall. On the other hand the rear wall, which was fronto-parallel to the camera, remained well matched.

In all scenes, the most consistent disparity map improvement was seen when around 10k points were used, regardless of which type of pattern was used and whether or not the ambient image was included. This reflects the fact that above this number of points, there were relatively few untextured regions left in the image. Additionally this suggests there was sufficient texture on the scale of a correlation window (12 px square). For instance, when only 2-3k points were used the disparity maps contained a lot of holes from regions that were not illuminated with the pattern.

In some scenes, such as Bricks (Figure 7.10) and Corner (7.12), SLRD and ambient texture performed relatively poorly. If a scene contains good local ambient texture, as is the case with Bricks, it is possible that adding additional texture on top may degrade matching performance. This is particularly the case with SLRD as the simulated spots do not fully capture the interaction with the laser beam and the local surface of the scene. In this particular case, a solution would be to exclusively paint texture into image regions which are not already sufficiently textured.

As discussed in Section 3.3.2, a prior range information such as LIDAR measurements may be combined with image intensity information to produce more accurate disparity maps than with intensity alone. To date, most popular stereo matching algorithms, both local and global, have been used for data fusion. MRF frameworks in particular show promise (Zhu, Wang, Yang, Davis, and Pan, Zhu et al.). The use of an MRF using LIDAR seed points as a high-confidence *a priori* input with ambient image texture would likely be more accurate[3] than Gotcha with simulated texture, at the expense of computation time.

The naive approach of texturing the entire image has the same issues with redundancy described in Section 6.5. The next section discusses a strategy for intelligently texturing an image using a LIDAR scan by predicting which regions of an image will fail to be matched.

7.3 Match prediction

In this section, a binary classifier is developed to predict which pixels in an image are likely to be stereo matched. The motivation is to reduce redundancy by selectively providing texture in image regions where the matcher is predicted to fail. The classifier was developed with Gotcha in mind, but training could be performed using a different stereo matcher without loss of generality.

In the case of region-growing stereo specifically, if it is possible to determine which regions of the image are unlikely to be matched, then the LIDAR can be scanned selectively in those areas. An appropriate method for selective LIDAR scanning was introduced in Section 6.5.

The performance of Gotcha depends on both the initial seed points and the texture in the image. Intuitively, there are two ways in which pixels are matched. The first trivial case is when a pixel is itself a seed point. The second is that the region growing process is able to expand the disparity map from a seed point to a particular pixel.

The following section describes how the initial distribution of seed points, combined with measures of local image texture, can accurately predict the matching performance of a stereo matcher.

7.3.1 Measures of image texture

Texture features can be classified into first and second order statistics. Higher orders exist, but they are rarely used. First order statistics consider intensities only and can be derived from image histograms. Second order statistics were popularised by Haralick et al. (1973), who introduced the gray level co-occurence matrix (GLCM) from which many features can be computed. Second

[3] Zhu demonstrated a typical 60% accuracy improvement in test imagery compared to stereo alone

order ('Haralick') features are commonly used for texture classification and comparison. For the purposes of this work, first order features were found to be sufficient.

7.3.1.1 Image gradient

Stereo match performance is often well correlated with the presence of edges in an image as edges are by definition formed by regions of high intensity variation. Edge detection methods, such as the widely used Canny detector (Canny, 1986), normally involve computing the image intensity gradient. Approximations of the gradient function are obtained by convolving the image with a kernel. Examples of this include the Roberts cross, Scharr, Sobel and Prewitt operators (Petrou and Petrou, 2010).

A different convolution kernel is required for the horizontal and vertical directions and the resulting gradient images can be combined to form a gradient magnitude image and a gradient direction. An example of the Sobel operator applied to an image is shown in Figure 7.17.

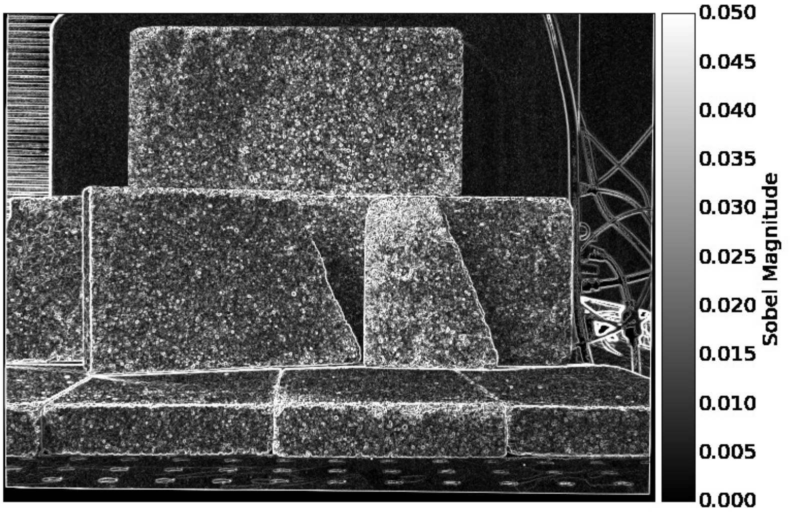


Figure 7.17: Sobel filtering applied to the Bricks scene. The Sobel magnitude is the output of the Sobel operator and is always positive.

7.3.1.2 Image entropy

Entropy is a measure of how much information is contained in a signal (Shannon, 1948). Entropy is a statistical measure of randomness in a system. The first order (histogram) entropy is defined for an image as:

$$E = -\sum(p_i \log(p_i)) \tag{7.4}$$

Where p_i is the number of pixels with intensity value i within the region of interest (that is, p represents the region histogram).

The entropy can also calculated using the grey-level co-occurrence matrix (GLCM) (Haralick et al., 1973). The GLCM C_{ij} is an $N \times N$ matrix where N is the number of grey levels present in the image (i.e. 255 for an 8-bit image). A GLCM is defined for a particular displacement operator, e.g. $d = (x, y)$. Given an image $I(u, v)$, The entry $C[i, j]$ is defined as the number of pixels for which $I(u, v) = i$ and $I(u + x, v + y) = j$. The entropy can then be defined as:

$$E = -\sum_i \sum_j C_{ij} \log C_{ij} \tag{7.5}$$

This definition varies slightly from the first and is slightly more challenging to calculate, as it requires a choice of displacement operator. For texture analysis, the operator is typically chosen to be large enough to cover the type of texture to be identified.

For simplicity and speed, the first order entropy was used. Histogram computation is comparatively fast as it is linear in the number of intensity levels, rather than quadratic. Figure 7.18 shows the 5x5 local entropy for an example image.

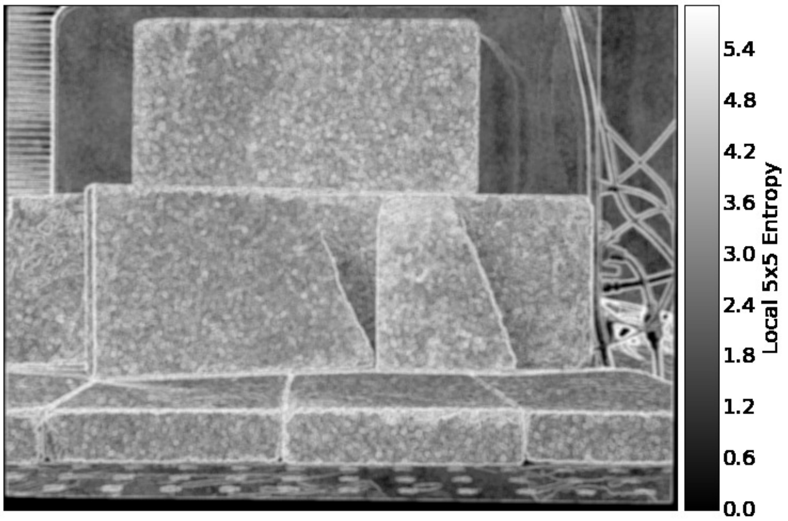


FIGURE 7.18: Local entropy map for Bricks with a 5 x 5 neighbourhood.

Like the gradient operator, higher values of image entropy correlate to well-textured image regions such as the bricks. The computer chassis behind the bricks, with a largely uniform intensity, has a low entropy. Unlike the gradient, no spatial information is preserved by the entropy.

7.3.1.3 SIFT keypoint distance

There is a strong correlation between the distribution of initial Gotcha seed points and the final disparity map. This is expected, since these seed points are typically located in high texture regions. The seed point distribution also encodes coarse information about occlusions, as occluded features are not matched in the initial stage of Gotcha.

Seed point density was not chosen as a texture metric because it requires specifying a region of interest. An alternative metric is the distance to the nearest seed point (DNSP). Given a list of i pixels (x_i, y_i) and a list of j seed point locations (x_{sj}, y_{sj}), the DNSP for each pixel is equal to:

$$DNSP_{(x_i,y_i)} = \arg\min_i(\sqrt{(x_{si} - x)^2 + (y_{si} - y)^2}) \tag{7.6}$$

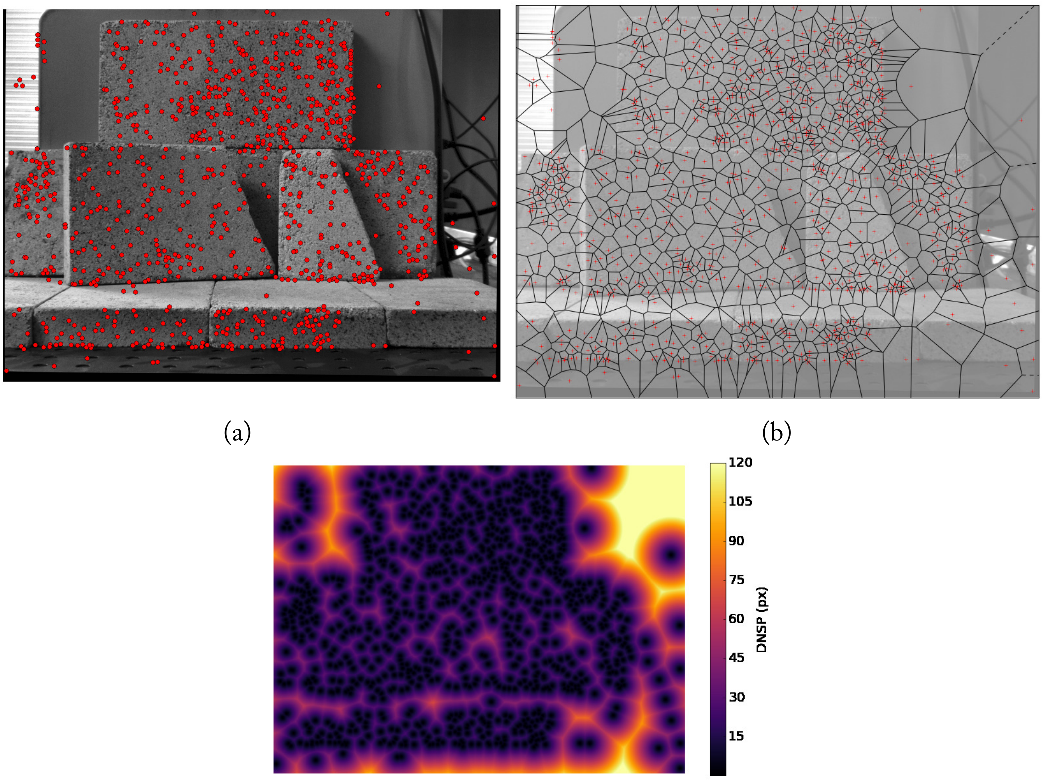
This is a nearest neighbour search and, computationally, is equivalent to constructing a Voronoi tessellation (Aurenhammer, 1991) from the seed points. The DNSP is therefore the distance from each pixel in a Voronoi cell to the seed point corresponding to that cell. Figure 7.19 shows an example of a left stereo image, matched SIFT seed points and the resulting DNSP.

7.3.1.4 Benchmark results

Stereo pairs from the Middlebury dataset (Section 3.2.3) were used to provide training data for the classifier. For each stereo pair, the image entropy, gradient magnitude (Sobel) and per-pixel DNSP was calculated. Each stereo pair was matched using Gotcha with the same (default) settings and the disparity maps were converted into binary representations where a '1' pixel indicates a match was found.

Comparisons between pairs of texture metrics, showing how many pixels are marked as matched or unmatched are shown in Figure 7.20.

Comparing the performance of each parameter individually, entropy performs well with separated maxima for matched and unmatched pixels. Using the Sobel gradient magnitude alone is not sufficient, but when combined with for example entropy Figures 7.20a and 7.20b the two distribu-


(a)

(b)

(c)

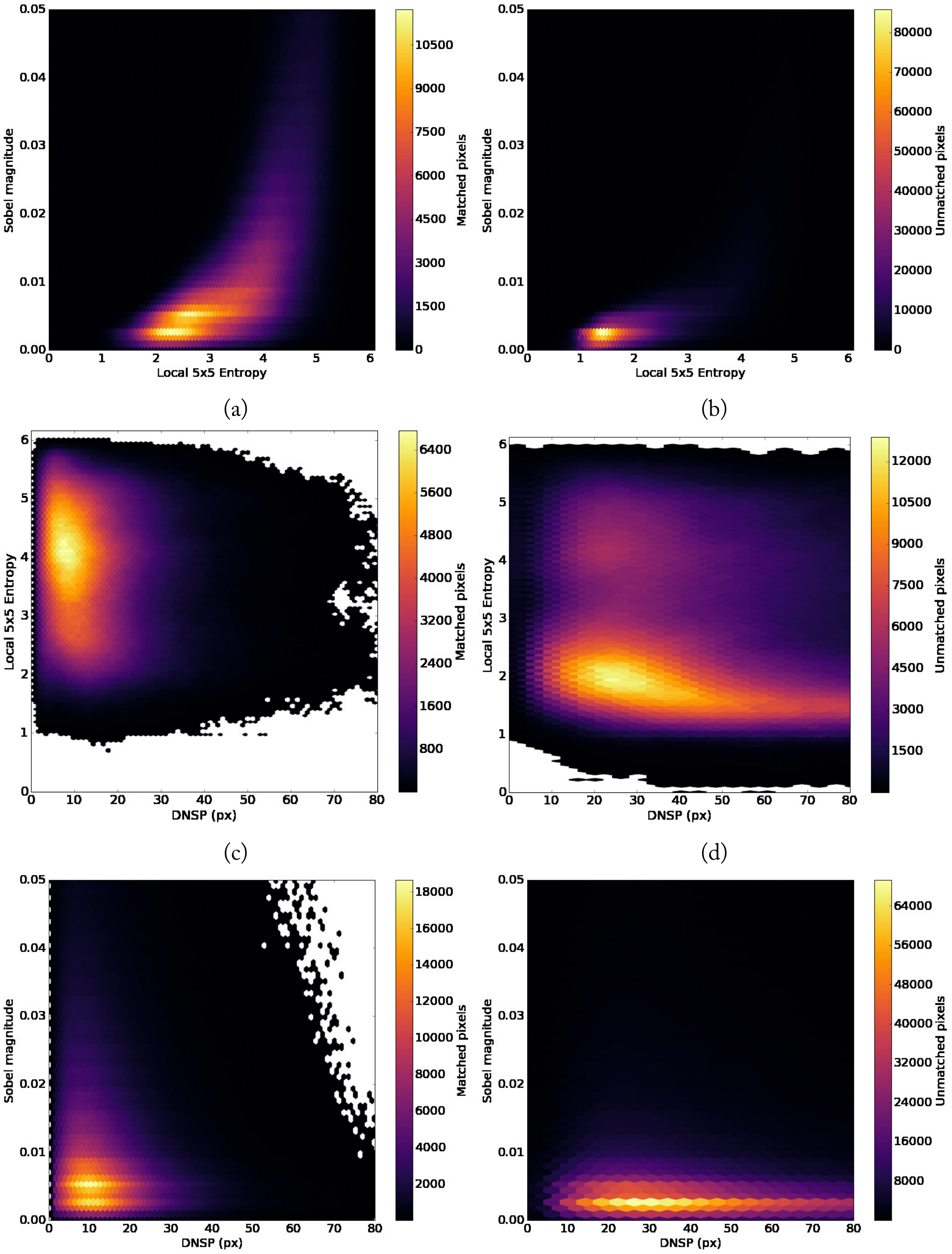
FIGURE 7.19: (a) Matched SIFT keypoints (red dots) in the Bricks scene (b) Voronoi cells produced from the set of matched keypoints. Larger cells imply a further distance to the next closest seed. (c) Per-pixel DNSP generated using matched keypoints.

tions are more distinct. The results suggest that DNSP is a good metric for determining whether a pixel will be matched (DNSP $<\sim$ 30 px), but is best combined with entropy.

Empirically, matched points tend to have a DNSP of under 35-40 px, an entropy greater than 2 and a gradient magnitude less than 0.02. The local entropy window size did not affect the results significantly.

7.3.2 Prediction

Predicting whether a pixel will be matched or not, given a number of textural cues, can be posed as a binary classification problem. There are many techniques for binary classification. Among the more popular are neural networks, k-nearest neighbours, support vector machines (SVMs), decision trees and random forests. The performance of a particular classifier can be characterised using a confusion matrix, defining the following properties:



(a) (b)

(c) (d)

(e) (f)

Figure 7.20: Number of pixels matched and unmatched (out of 14M pixels) for various combinations of local 5 x 5 entropy, gradient magnitude and DNSP. (a) entropy vs gradient, matched, (b) entropy vs gradient, unmatched, (c) DNSP vs entropy, matched, (d) DNSP vs entropy, unmatched, (e) DNSP vs gradient, matched, (f) DNSP vs gradient, unmatched.

- True positive (TP)
- True negative (TN)
- False positive (FP)
- False negative (FN)

A positive result was defined as a pixel being matched. Using these figures, further metrics can be calculated such as the precision:

$$P = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} \tag{7.7}$$

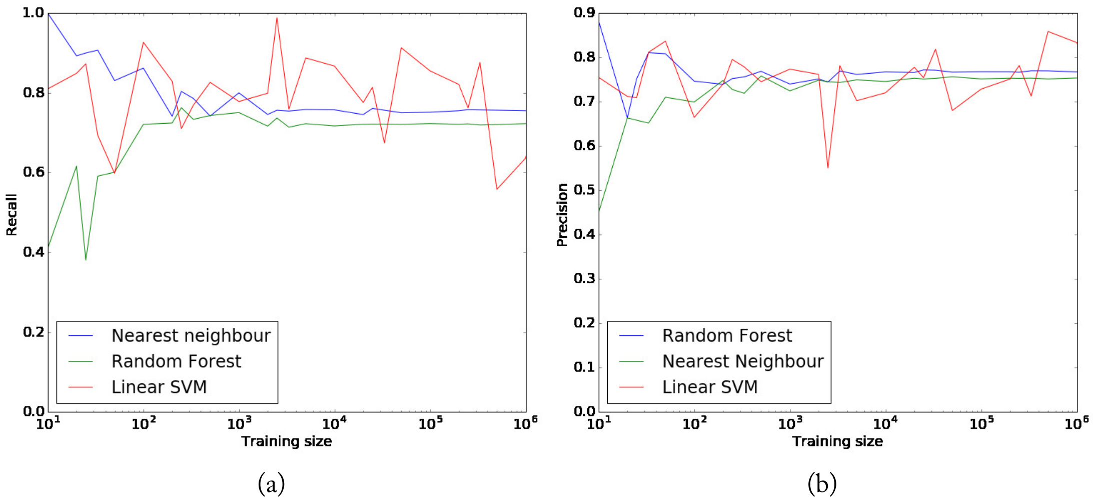
and recall:

$$R = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \tag{7.8}$$

False negative results, pixels which are incorrectly marked as unmatched, might cause the LIDAR to be scanned where it does not need to be. On the other hand if there are many false positive classifications, large areas of the image may remain unmatched. Thus both high precision and recall scores are desirable, but a high precision is more important.

Three binary classifiers were evaluated: linear support vector machine (SVM), random forest (RF) and nearest neighbour (NN). The performance of each classifier with respect to varying the size of the training data is shown in Figure 7.21. Training was performed 10 times for each sample size. Training points were randomly sampled and then tested on 1 M randomly sampled points held back from the training set.

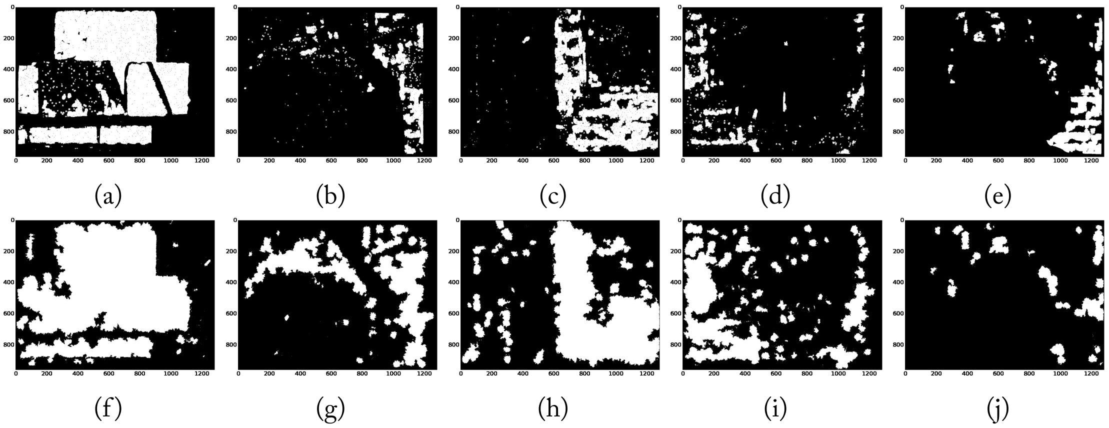
The performance of the NN and RF classifiers appeared to be stable above 10k training points. The SVM performed similarly, but there was a stronger dependence on the set of training points used. Training sizes orders of magnitude larger than this did not yield significantly better classification performance. Given the smooth shaped clusters shown in Figure 7.20, this result should be expected. Using 10k training points, the classification time per pixel was 14.3 μs for RF, 34.3 μs for NN and 45.2 μs for SVM. Classification is well-suited to parallelisation, since each pixel is labelled independently.



(a) (b)

Figure 7.21: Precision and recall curves for RF, NN and SVM classifiers as a function of the number of training points.

When applied to test images, small holes in the predicted map were filled. Next, binary erosion were performed to remove any remaining isolated points in the predicted map. Representative predicted matching maps for the 5 scenes in Section 6.4 are shown in Figure 7.22, classified using a RF trained on 50k points.



(a) (b) (c) (d) (e)

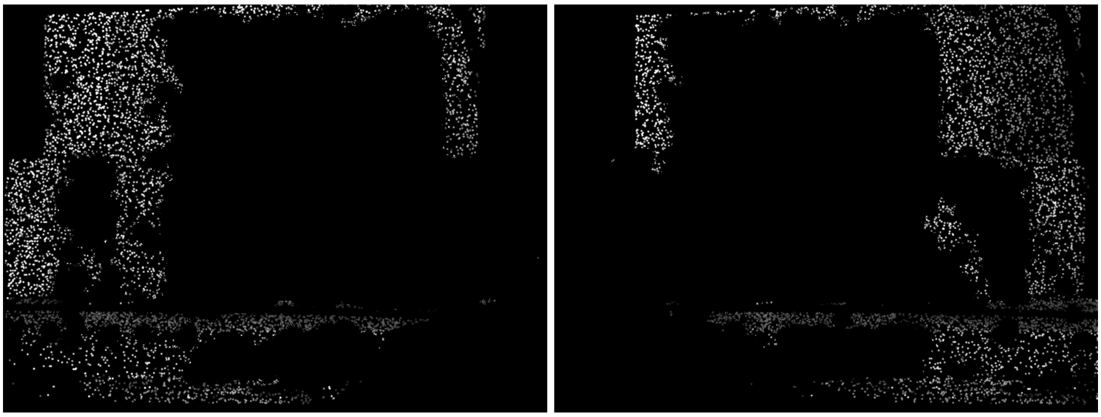
(f) (g) (h) (i) (j)

Figure 7.22: Binarised Gotcha disparity maps for Bricks, Chair, Corner, PotPlant and Biba scenes (a-e). Disparity maps predicted using a random forest classifier trained on 50k points (f-j). White areas denote matched regions.

The classifier tended to be overly optimistic with regard to how much of the image was matched, but unmatched regions were generally well identified.

7.3.3 Intelligent image texturing

Using the classifier, it was possible to selectively texture images in predicted unmatched regions. In order to best exploit existing image texture, stereo pairs were matched without any additional illumination. In parallel with this, the classification result was used to produce a masked random dot pattern in featureless regions only. The masked pattern for Bricks is shown in Figure 7.23.



(a) (b)

FIGURE 7.23: Masked LIDAR random dot pattern for Bricks. The pattern is generated only in image regions that are classified as featureless.

The disparity maps from the stereo images with ambient illumination were combined with the disparity maps from the selective texture projection images. If any holes in the disparity map remained, they were filled using the iterative scanning technique described in Section 6.5.2. An example workflow for Bricks is shown in Figure 7.24.

In this case, the initial number of LIDAR points used was 12000 which was reduced to 3957 after image classification. After gap filling, 7983 LIDAR points were used in total. In comparison, using a progressive LIDAR scan with occlusion detection (Section 6.5.3) required 21417 points. This is a reduction in LIDAR points of over 60%. This technique is most effective when there is significant texture available in the scene. Otherwise, texturing the entire image is an appropriate strategy for robust matching.

7.4 Conclusions

In this chapter, two methods for texturing an image using a LIDAR were described. It was shown that texture projection using a LIDAR enables robust and dense disparity map generation. Ran-

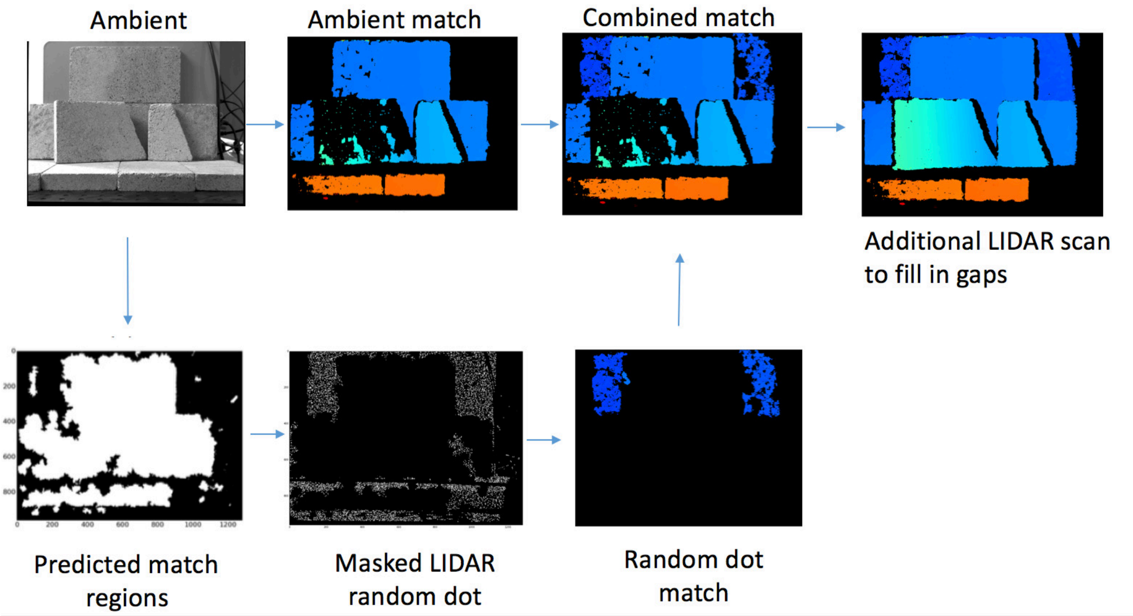


Figure 7.24: Workflow for intelligent image texturing using the Bricks scene as an example. The final unmatched regions on the left and right edges are parts of the image which are occluded.

dom dot patterns were created by combining images acquired of a visible LIDAR spot or by simulating the images of the spot given a system cross-calibration. Both methods were found to be effective, particularly when the pattern is combined with ambient light imagery.

A binary classifier was developed that was capable of predicting which parts of a stereo pair would be unmatched using Gotcha. The classifier was used to generate masked random dot patterns exclusively in featureless image regions. Matching this pattern and ambient light imagery allowed for exploitation of natural image texture, supplemented with artificial texture where necessary.

Results from indoor and outdoor scenes were presented, demonstrating that LIDAR texture generation is versatile and can be applied to pre-existing systems without modification. It was found that a relatively sparse scan is sufficient, with an asymptotic improvement in the number of matched pixels as the number of LIDAR points increased. For the test imagery, relatively sparse scans containing less than 20 k points were suitable. Selectively texturing the image reduced the number of LIDAR points required, though this relies on there being some texture in the ambient imagery.

This further reduction in the number of LIDAR points required is commercially relevant. Dense, fast-scanning, commercial systems produce vast quantities of data per scene and reducing the number of points necessary for analysis may enable real-time applications. Bespoke LIDAR scanning systems using galvanometers or gimbals in combination with an accurate 1D LIDAR are quite slow. However these systems are more accurate and are significantly cheaper to construct compared to COTS 3D scanning LIDAR systems. The bottleneck is typically the LIDAR units which are available with sampling rates of up to around 100 Hz. Scans requiring tens or hundreds of thousands of points are therefore impractical for many real-world applications. Using the techniques in this chapter, less than 20 k LIDAR points were required for dense matching in all the test scenes. Such a scan would take around 5-6 minutes using a 100 Hz system which is approaching the limit of acceptability for static industrial inspection tasks.

Conclusions 8

From a survey of previous studies of data fusion of stereo matching with additional range information for the purpose of disparity map improvement (Section 3.3.2), it is clear that most previous research focused on ToFCs. Comparatively little research had been performed into using LIDAR. Most solutions aimed to solve the issues of stereo matching in featureless regions by either directly filling in unmatched regions, *a posteriori*, or by constraining the stereo matching process based on additional range information *a priori*.

This work explored data fusion of stereo with LIDAR in detail, considering the complementarity of the two techniques. Stereo matching can provide dense reconstruction in well textured regions and LIDAR can provide sparse, but accurate and robust measurements elsewhere. Two approaches were taken: in Chapter 6, LIDAR measurements were used, *a priori*, to generate robust correspondences to seed a region growing stereo matcher (Gotcha, Section 6.2). In Chapter 7, LIDAR was used to generate artificial texture in featureless regions. In both cases significant improvements in the number of matched pixels were observed compared to matching images with ambient illumination alone.

Imaging the LIDAR spot directly (Section 4.3.2) provides robust correspondences between the two views and this fact has been repeatedly exploited for the purposes of fast and accurate camera calibration (Section 5.4), tiepoint generation for stereo region growing (Section 6.3) and texture projection (Section 7.2). LIDAR spot imaging is not limited to visible light, but visible systems allow the scene to be imaged in colour, whereas a NIR system would require a removable bandpass filter. In Chapters 6 and 7 it was shown that provided a cross-calibration exists between the stereo and LIDAR systems, the LIDAR spot need not be directly visible.

Since stereo matching excels when the input images contain strong texture, it was proposed that the LIDAR should be scanned where stereo matching was expected to fail. By progressively scanning the LIDAR at increasingly high angular resolution in unmatched regions and/or predicting which parts of an image would be unmatched prior to starting the scan, the number of LIDAR scan points required for a particular scene required to densely reconstruct scenes (Sections 6.5 and 7.3) was reduced. Indoor and outdoor stereo matching benchmark datasets were used for testing as well as several novel challenging indoor scenes. Stereo image matching was performed using Gotcha and SGM for comparison. A key observation was that many scenes which were apparently featureless, to the extent that stereo matching failed, still contained sufficient local texture to enable matching guided via LIDAR-derived correspondences.

System design choices are dictated by project requirements and there are wide variety of camera arrangements and LIDAR systems to choose from. With this in mind, the methods presented for disparity map improvement are largely agnostic to the hardware used. The LIDAR and scanning system used for this work were chosen with accuracy as a priority, rather than acquisition speed. The ability to selectively point the LIDAR, combined with efficient scanning patterns, significantly reduced scanning times, depending on the scene. The calibration method in Chapter 5 requires a visible LIDAR system, but places no specific requirements on the cameras used.

The scanning system described in Chapter 4 is not suitable for real-time operation, for example onboard a moving vehicle. It is nevertheless a cheaper and more accurate alternative to COTS scanning LIDAR systems which currently cost tens of thousands of pounds. If the system acquisition speed was improved to more than 1k pt/s, resulting in scan times of seconds to minutes, it would be usable for tasks like industrial inspection of static objects. On the other hand, LIDAR-seeded stereo matching and texture projection were shown to be possible using existing sensor platforms such as the KITTI robotic car (using a Velodyne LIDAR), without modification.

Overall this research has demonstrated that integration of scanning LIDAR measurements into stereo camera systems is beneficial: providing accurate ground truth, intrinsic and extrinsic self-calibration and improved disparity maps in featureless or low-texture scenes. As systems integrating multiple depth sensors are becoming increasingly common, the results from this work reinforce the advantages of merging active range data with passive stereo imaging.

8.1 Thesis Contributions

In summary, this thesis provides three main contributions to the field:

1. (Chapter 5) A method of calibrating either a single camera or a stereo camera system using a visible-beam scanning LIDAR. The LIDAR is used to generate robust calibration points and both intrinsic and extrinsic calibration is possible without an explicit calibration target.

2. (Chapter 6) A method for integrating a scanning LIDAR into a region-growing stereo matching algorithm (Gotcha). By using LIDAR measurements as seed correspondences, it is possible to both improve matching in featureless regions while reducing the number of LIDAR points needed to densely reconstruct the scene.

3. (Chapter 7) A method for using a scanning LIDAR to generate texture in featureless image regions. Alongside this, a classifier was developed which was capable of identifying which parts of an image were likely to be unmatched. Classification results were then used to guide the LIDAR scan initially to avoid redundant measurements.

8.2 Recommendations for further work

Testing was performed in a laboratory environment, with the exception of the use of Middlebury and KITTI benchmark data. While effort was taken to develop scenes that were particularly challenging for stereo matching, it would be beneficial to acquire data from a wider variety of environment types. Due to the acquisition speed of the system, outdoor testing was only performed on benchmark data and given the suitability of LIDAR for long range, outdoor measurements, such scenes would be useful. In addition to this, the data fusion algorithms should be tested using different LIDAR and camera systems.

The acquisition speed of the system could be improved in a number of ways. A galvanometer scanner would allow a much more compact system with improved scanning speeds. Similarly the LIDAR used was chosen for precision and accuracy, and is slow at only 20 Hz. Faster 1D LIDAR systems are available, for example from Acuity (Portland, USA) [1]. A major bottleneck for direct LIDAR spot imaging is the camera frame rate. High end machine vision cameras offer 150 fps at full resolution, or higher if only part of the sensor is read out. Exploiting this to only read out the sensor region where the laser spot is predicted to fall would improve performance up to several

[1] `http://www.acuitylaser.com`, accessed 20/8/2016

hundred fps. This could be achieved using the mapping between LIDAR spot location and scan angle described in Section 6.5.1.

The use of a visible LIDAR system provided several advantages over a NIR system, namely that the laser spot was easily identified in colour imagery which is useful for texturing and calibration. On the other hand, as shown in the PotPlant scene (Figure 6.5), vegetation was not reconstructed well. Further testing is suggested, using a wider variety of plants with differing leaf sizes and geometries. Alternatively a NIR LIDAR system could be used, to take advantage of better leaf reflectance.

A 'black box' camera calibration system that requires no significant human interaction or operator expertise would be well suited for commercial development. So far, the calibration routine was only tested using high quality, low-distortion machine vision lenses. The method should also be applicable to fisheye lenses, with an appropriate choice of camera model. These lenses pose a greater challenge for calibration due their more complicated distortion characteristics and very wide field of view. Since calibration points should ideally be distributed across the field of view, this would require a scanner with a suitably large angular scan range.

Stereo matching prediction was only used to direct where the LIDAR should be scanned, but this kind of classification could also be applied to other fields such as planetary mapping. Match prediction could also be used as a diagnostic tool to optimise scene illumination, for example by identifying image regions that do not contain sufficient texture. The classifier was only trained using Gotcha and a small number of textural metrics, which are unlikely to be optimal. Other texture metrics could be explored and the classifier applied to different stereo matching algorithms.

In general the methods described in this thesis could be used to reduce the number of LIDAR points required in data fusion applications. As real-time scanning LIDAR systems generate vast quantities of data (potentially millions of 3D points every second), a reduction in the number of necessary points is beneficial. For instance, an autonomous vehicle controller might choose to discard every other LIDAR point if it was determined that it would not negatively impact scene reconstruction.

Ultimately a more extensive analysis of the trade-off between acquisition time (many measurements) and accuracy (fewer, slower, measurements) should be performed. To avoid the significant capital outlay for a scanning LIDAR system, this could be performed by acquiring a high accuracy scan with a slower, lower cost, system and simulating a faster system by degrading the data artificially.

8.3 Open problems

The goal of the 'ideal' imaging system described in Section 1.3 has not yet been realised. Although there have been significant advances in stereo matching algorithms over the past 20 years, solving the correspondence problem is still challenging and will likely remain so for the near future. Stereo vision systems will continue to be relevant as colour imagery provides important contextual information that is lacking from pure range sensors.

COTS 3D scanning LIDAR systems are still not affordable for many applications and are bulky. It is reasonable to assume that LIDAR systems will become progressively miniaturised using, for example Microelectromechanical Systems (MEMS) technology. Flash LIDAR (Section 2.4.4) is not yet available as a COTS product, but will likely benefit from economy of scale once the technology matures. Due to physical and geometric limitations, ranging methods like ToFCs and laser triangulation remain suited to specific applications such as indoor or close-range operation.

In light of this, data fusion is an attractive and proven solution to deal with the deficiencies of different sensors. Large emerging markets such as autonomous vehicles and portable electronics are likely to be the main driver behind these new technological developments.

Bibliography

Abdel-Aziz, Y. I. and H. M. Karara (2015). Direct Linear Transformation from Comparator Coordinates into Object Space Coordinates in Close-Range Photogrammetry. In *Photogrammetric Engineering & Remote Sensing*. doi:10.14358/PERS.81.2.103.

Adams, M. D. (1993). Amplitude modulated optical range data analysis in mobile robotics. In *Pattern Recognition (ICPR), 21st International Conference on*, pp. 8–13. doi:10.1109/ROBOT.1993.292116.

Ahmadabadian, A. H., S. Robson, J. Boehm, M. Shortis, K. Wenzel, and D. Fritsch (2013). A comparison of dense matching algorithms for scaled surface reconstruction using stereo camera rigs. *ISPRS Journal of Photogrammetry and Remote Sensing 78*, 157–167. doi:10.1016/j.isprsjprs.2013.01.015.

Amann, M.-C., T. M. Bosch, M. Lescure, R. A. Myllylae, and M. Rioux (2001). Laser ranging: a critical review of usual techniques for distance measurement. *Optical Engineering 40*(1), 10–19. doi:10.1117/1.1330700.

Aurenhammer, F. (1991). Voronoi diagrams—a survey of a fundamental geometric data structure. *ACM Computing Surveys (CSUR) 23*(3), 345–405. doi:10.1145/116873.116880.

Badino, H., D. Huber, and T. Kanade (2011). Integrating LIDAR into Stereo for Fast and Improved Disparity Computation. In *3D Imaging, Modeling, Processing, Visualization and Transmission (3DIMPVT), 2011 International Conference on*, pp. 405–412. IEEE. doi:10.1109/3DIMPVT.2011.58.

Baltsavias, E. P. (1999). Airborne laser scanning: basic relations and formulas. *ISPRS Journal of Photogrammetry and Remote Sensing 54*(2–3), 199–214. doi:10.1016/S0924-2716(99)00015-5.

Baribeau, R. and M. Rioux (1991). Influence of speckle on laser range finders. *Applied Optics 30*(20), 2873. doi:10.1364/AO.30.002873.

Beder, C., B. Bartczak, and R. Koch (2007). A Combined Approach for Estimating Patchlets from PMD Depth Images and Stereo Intensity Images. In F. Hamprecht, C. Schnörr, and B. Jähne (Eds.), *Lecture Notes in Computer Science*, pp. 11–20. Springer Berlin Heidelberg. doi:10.1007/978-3-540-74936-3_2.

Beraldin, J. A. (2004). Integration of laser scanning and close-range photogrammetry -The last decade and beyond. In *XXth International Society for Photogrammetry and Remote Sensing ISPRS Congress*, Istanbul, pp. 972–983. XXth International Society for Photogrammetry and Remote Sensing (ISPRS) Congress. Commission VII.

Beraldin, J. A. (2009). Basic theory on surface measurement uncertainty of 3D imaging systems. In J. A. Beraldin, G. S. Cheok, M. McCarthy, and U. Neuschaefer-Rube (Eds.), *IS&T/SPIE Electronic Imaging*, pp. 723902. SPIE. doi:10.1117/12.804700.

Berkovic, G. and E. Shafir (2012). Optical methods for distance and displacement measurements. *Advances in Optics and Photonics 4*(4), 441. doi:10.1364/AOP.4.000441.

Besl, P. J. (1988). Active, optical range imaging sensors. *Machine Vision and Applications 1*(2), 127–152. doi:10.1007/BF01212277.

Biemond, J., R. L. Lagendijk, and R. M. Mersereau (1990). Iterative methods for image deblurring. *Proceedings of the IEEE 78*(5), 856–883. doi:10.1109/5.53403.

Birchfield, S. and C. Tomasi (1998). Depth discontinuities by pixel-to-pixel stereo. In *Computer Vision, 1998. Sixth International Conference on*, pp. 1073–1080. doi:10.1109/ICCV.1998.710850.

Blais, F. (2004). Review of 20 years of range sensor development. *Journal of Electronic Imaging 13*(1), 231–243. doi:10.1117/1.1631921.

Boehler, W., M. B. Vicent, and A. Marbs (2003). Investigating Laser Scanner Accuracy (White Paper, FH Mainz University).

Boyle, W. S. and G. E. Smith (2013). Charge Coupled Semiconductor Devices. *Bell System Technical Journal 49*(4), 587–593. doi:10.1002/j.1538-7305.1970.tb01790.x.

Bradski, G. (2000). The opencv library. *Doctor Dobbs Journal 25*(11), 120–126.

Brown, M. Z., D. Burschka, and G. D. Hager (2003). Advances in computational stereo. *Pattern Analysis and Machine Intelligence, IEEE Transactions on 25*(8), 993–1008. doi:10.1109/TPAMI.2003.1217603.

Campbell, J. R., D. L. Hlavka, E. J. Welton, C. J. Flynn, D. D. Turner, J. D. Spinhirne, V. S. Scott, and I. H. Hwang (2013). Full-Time, Eye-Safe Cloud and Aerosol Lidar Observation at Atmospheric Radiation Measurement Program Sites: Instruments and Data Processing. *Journal of Atmospheric and Oceanic Technology 19*(4), 431–442. doi:doi:10.1175/1520-0426(2002)019<0431:FTESCA>2.0.CO;2.

Canny, J. (1986). A Computational Approach to Edge Detection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on PAMI-8*(6), 679–698. doi:10.1109/TPAMI.1986.4767851.

Chow, J. C. K. and D. D. Lichti. Photogrammetric Bundle Adjustment With Self-Calibration of the PrimeSense 3D Camera Technology: Microsoft Kinect. *IEEE Access 1*, 465–474. doi:10.1109/ACCESS.2013.2271860.

Collis, R. T. H. and M. G. H. Ligda (1964). Laser Radar Echoes from the Clear Atmosphere. *Nature 203*(4944), 508–508. doi:10.1038/203508a0.

Currie, D., S. Dell'Agnello, and G. Delle Monache (2011). A Lunar Laser Ranging Retroreflector Array for the 21st Century. *Acta Astronautica 68*(7-8), 667–680. doi:10.1016/j.actaastro.2010.09.001.

Day, T. and J.-P. Muller (1989). Digital elevation model production by stereo-matching spot image-pairs: a comparison of algorithms. *Image and Vision Computing* 7(2), 95–101. doi:10.1016/0262-8856(89)90002-4.

de Brujin, N. G. (1975). Acknowledgement of priority to C. Flye Sainte-Marie on the counting of circular arrangements of 2^n zeros and ones that show each n-letter word exactly once.

Deacon, A. T., A. G. Anthony, S. N. Bhatia, and J.-P. Muller (1991). Evaluation of a CCD-based facial measurement system. *Medical informatics = Medecine et informatique 16*(2), 213–228. doi:10.3109/14639239109012128.

Diebel, J. and S. Thrun (2006). An application of markov random fields to range sensing. *Advances in neural information processing systems*.

Duane, C. B. (1971). Close-Range Camera Calibration. *Photogrammetric engineering*, 855–866.

Ernst, I. and H. Hirschmuller (2008). Mutual Information Based Semi-Global Stereo Matching on the GPU. *Lecture Notes in Computer Science 5358*, 228–239.

Faig, W. (1975). Calibration of Close-Range Photogrammetric Systems: Mathematical Formulation. *Photogrammetric Engineering and Remote Sensing 41*(12).

Fankhauser, P., M. Bloesch, D. Rodriguez, R. Kaestner, M. Hutter, and R. Siegwart (2015). Kinect v2 for mobile robot navigation: Evaluation and modeling. In *2015 International Conference on Advanced Robotics (ICAR)*, pp. 388–394. IEEE. doi:10.1109/ICAR.2015.7251485.

Fischer, J., G. Arbeiter, and A. Verl (2011). Combination of Time-of-Flight depth and stereo using semiglobal optimization. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pp. 3548–3553. IEEE. doi:10.1109/ICRA.2011.5979999.

Fischler, M. A. and R. C. Bolles (1981). Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM 24*(6), 381–395. doi:10.1145/358669.358692.

Foix, S., G. Alenya, and C. Torras (2011). Lock-in Time-of-Flight (ToF) Cameras: A Survey. *IEEE Sensors Journal 11*(9), 1917–1926. doi:10.1109/JSEN.2010.2101060.

Fossum, E. R. (1995). CMOS image sensors: electronic camera on a chip. In *International Electron Devices Meeting*, pp. 17–25. IEEE. doi:10.1109/IEDM.1995.497174.

Fraser, C. S. (1997). Digital camera self-calibration. *ISPRS Journal of Photogrammetry and Remote Sensing 52*(4), 149–159. doi:10.1016/S0924-2716(97)00005-1.

Gallup, D., J. M. Frahm, P. Mordohai, and M. Pollefeys (2008). Variable baseline/resolution stereo. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on IS -*, pp. 1–8. IEEE. doi:10.1109/CVPR.2008.4587671.

Ganapathi, V., C. Plagemann, D. Koller, and S. Thrun (2010). Real time motion capture using a single time-of-flight camera. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on IS -*, pp. 755–762. IEEE. doi:10.1109/cvpr.2010.5540141.

Geiger, A., P. Lenz, C. Stiller, and R. Urtasun (2013). Vision meets robotics: The KITTI dataset. *The International Journal of Robotics Research 32*(11), 1231–1237. doi:10.1177/0278364913491297.

Gong, X., Y. Lin, and J. Liu (2013). 3D LIDAR-Camera Extrinsic Calibration Using an Arbitrary Trihedron. *Sensors 13*(2), 1902–1918. doi:10.3390/s130201902.

Grimson, W. E. L. (1985). Computational Experiments with a Feature Based Stereo Algorithm. *Pattern Analysis and Machine Intelligence, IEEE Transactions on PAMI-7*(1), 17–34. doi:10.1109/TPAMI.1985.4767615.

Gruen, A. (1985). Adaptive least squares correlation: a powerful image matching technique. *South African Journal of Photogrammetry 14*, 175–187. doi:10.1.1.93.6891.

Gruen, A. (2012). Development and Status of Image Matching in Photogrammetry. *The Photogrammetric Record 27*(137), 36–57. doi:10.1111/j.1477-9730.2011.00671.x.

Gudmundsson, S. A., H. Aanaes, and R. Larsen (2008). Fusion of stereo vision and Time-Of-Flight imaging for improved 3D estimation. *International Journal of Intelligent Systems Technologies and Applications 5*(3/4), 425. doi:10.1504/IJISTA.2008.021305.

Gurram, P., S. Lach, E. Saber, H. Rhody, and J. Kerekes. 3D Scene Reconstruction through a Fusion of Passive Video and Lidar Imagery. In *2007 36th IEEE Applied Imagery Pattern Recognition Workshop (AIPR 2007)*, pp. 133–138. IEEE. doi:10.1109/AIPR.2007.17.

Habib, A. F., A. M. Pullivelli, and M. F. Morgan (2005). Quantitative measures for the evaluation of camera stability. *Optical Engineering 44*(3), 033605–033605–8. doi:10.1117/1.1872840.

Hahne, U. and M. Alexa (2008). Combining Time-Of-Flight depth and stereo images without accurate extrinsic calibration. *International Journal of Intelligent Systems Technologies and Applications 5*(3/4), 325. doi:10.1504/IJISTA.2008.021295.

Hall, D. L. and J. Llinas (1997). An introduction to multisensor data fusion. *Proceedings of the IEEE 85*(1), 6–23. doi:10.1109/5.554205.

Halmos, M. J., M. D. Jack, J. F. Asbrock, C. Anderson, S. L. Bailey, G. Chapman, E. Gordon, P. E. Herning, M. H. Kalisher, L. F. Klaras, K. Kosai, V. Liquori, M. Pines, V. Randall, R. Reeder, J. P. Rosbeck, S. Sen, P. A. Trotta, P. Wetzel, A. T. Hunter, J. E. Jensen, T. J. DeLyon, C. W. Trussell, J. A. Hutchinson, and R. S. Balcerak (2001). 3D flash ladar at Raytheon. In G. W. Kamerman (Ed.), *Aerospace/Defense Sensing, Simulation, and Controls*, pp. 84–97. SPIE. doi:10.1117/12.440096.

Han, J., L. Shao, D. Xu, and J. Shotton (2013). Enhanced Computer Vision With Microsoft Kinect Sensor: A Review. *Cybernetics, IEEE Transactions on 43*(5), 1318–1334. doi:10.1109/TCYB.2013.2265378.

Hancock, S., M. Disney, J.-P. Muller, P. Lewis, and M. Foster (2011). A threshold insensitive method for locating the forest canopy top with waveform lidar. *Remote Sensing of Environment 115*(12), 3286–3297. doi:10.1016/j.rse.2011.07.012.

Hancock, S., P. Lewis, M. Foster, M. Disney, and J.-P. Muller (2012). Measuring forests with dual wavelength lidar: A simulation study over topography. *Agricultural and Forest Meteorology 161*, 123–133. doi:10.1016/j.agrformet.2012.03.014.

Haralick, R. M., K. Shanmugam, and I. Dinstein (1973). Textural Features for Image Classification. *Systems, Man and Cybernetics, IEEE Transactions on 3*(6), 610–621. doi:10.1109/TSMC.1973.4309314.

Hartley, A. and R. Zisserman (2003). Multiple View Geometry in Computer Vision. Cambridge University Press.

Hartley, R. I. (1997). In defense of the eight-point algorithm. *Pattern Analysis and Machine Intelligence, IEEE Transactions on 19*(6), 580–593. doi:10.1109/34.601246.

Hatze, H. (1988). High-precision three-dimensional photogrammetric calibration and object space reconstruction using a modified DLT-approach. *Journal of biomechanics 21*(7), 533–538. doi:10.1016/0021-9290(88)90216-3.

Hebert, M. (2000). Active and passive range sensing for robotics. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pp. 102–110 vol.1. IEEE. doi:10.1109/ROBOT.2000.844046.

Heikkila, J. (2000). Geometric camera calibration using circular control points. *Pattern Analysis and Machine Intelligence, IEEE Transactions on 22*(10), 1066–1077. doi:10.1109/34.879788.

Heikkila, J., O. C. V. Silven, and P. R. . P. . I. C. S. C. on (1997). A four-step camera calibration procedure with implicit image correction. In *Computer Vision and Pattern Recognition, 1997. Proceedings., 1997 IEEE Computer Society Conference on*. doi:10.1109/CVPR.1997.609468.

Hess, M., S. Robson, and A. Hosseininaveh Ahmadabadian (2014). A contest of sensors in close range 3D imaging: performance evaluation with a new metric test object. *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences XL-5*, 277–284. doi:10.5194/isprsarchives-XL-5-277-2014.

Hirschmuller, H. (2008). Stereo Processing by Semiglobal Matching and Mutual Information. *Pattern Analysis and Machine Intelligence, IEEE Transactions on 30*(2), 328–341. doi:10.1109/TPAMI.2007.1166.

Hirschmuller, H. and D. Scharstein (2007). Evaluation of Cost Functions for Stereo Matching. In *Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on*, pp. 1–8. IEEE. doi:10.1109/CVPR.2007.383248.

Jelalian, A. (1990). *Laser Radar Systems* (1st ed.). Norwood: Artech House.

Jin, M. and T. Maruyama (2012). A fast and high quality stereo matching algorithm on FPGA. In *Field Programmable Logic and Applications (FPL), 2012 22nd International Conference on*, pp. 507–510. doi:10.1109/FPL.2012.6339266.

Juberts, M. and A. J. Barbera (2004). Status report on next-generation LADAR for driving unmanned ground vehicles. *Optics East*, 1–12. doi:10.1117/12.580235.

Julesz, B. (1960). Binocular Depth Perception of Computer-Generated Patterns. *Bell Labs Technical Journal 39*(5), 1125–1162. doi:10.1002/j.1538-7305.1960.tb03954.x.

Kalarot, R. and J. Morris (2010). Comparison of FPGA and GPU implementations of real-time stereo vision. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPR Workshops)*, pp. 9–15. IEEE. doi:10.1109/CVPRW.2010.5543743.

Kanade, T. and M. Okutomi (1994). A stereo matching algorithm with an adaptive window: theory and experiment. *Pattern Analysis and Machine Intelligence, IEEE Transactions on 16*(9), 920–932. doi:10.1109/34.310690.

Khoshelham, K. and S. O. Elberink (2012). Accuracy and Resolution of Kinect Depth Data for Indoor Mapping Applications. *Sensors 12*(12), 1437–1454. doi:10.3390/s120201437.

Knuth, D. E. (1974). Postscript about NP-hard problems. *SIGACT News 6*(2), 15–16. doi:10.1145/1008304.1008305.

Kollorz, E., J. Penne, J. Hornegger, and A. Barke (2008). Gesture recognition with a Time-Of-Flight camera. *International Journal of Intelligent Systems Technologies and Applications*. doi:10.1504/ijista.2008.5.issue-3-4;pageGroup:string:Publication.

Kolmogorov, V. and R. Zabih (2001). Computing visual correspondence with occlusions using graph cuts. In *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on IS - SN -*, pp. 508–515 vol.2. IEEE Comput. Soc. doi:10.1109/ICCV.2001.937668.

Kolmogorov, V. and R. Zabih (2004). What energy functions can be minimized via graph cuts? *Pattern Analysis and Machine Intelligence, IEEE Transactions on 26*(2), 147–159. doi:10.1109/TPAMI.2004.1262177.

Konolige, K. (2010). Projected texture stereo. In *Robotics and Automation ICRA, IEEE International Conference on*, pp. 148–155. IEEE. doi:10.1109/ROBOT.2010.5509796.

Krotkov, E. (1988). Focusing. *International Journal of Computer Vision 1*(3), 223–237–237. doi:10.1007/BF00127822.

Krüger, L. and C. Wöhler (2011). Accurate chequerboard corner localisation for camera calibration. *Pattern Recognition Letters 32*(10), 1428–1435. doi:10.1016/j.patrec.2011.04.002.

Kuhnert, K. and M. Stommel (2006). Fusion of Stereo-Camera and PMD-Camera Data for Real-Time Suited Precise 3D Environment Reconstruction. In *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on IS - SN - VO -*, pp. 4780–4785. IEEE.

Kwak, K., D. F. Huber, H. Badino, and T. Kanade (2011). Extrinsic calibration of a single line scanning lidar and a camera. *IROS*, 3283–3289. doi:10.1109/IROS.2011.6094490.

Landstrom, A. and M. J. Thurley (2012). Morphology-Based Crack Detection for Steel Slabs. *Selected Topics in Signal Processing, IEEE Journal of 6*(7), 866–875. doi:10.1109/JSTSP.2012.2212416.

Lange, R. and P. Seitz (2001). Solid-state time-of-flight range camera. *IEEE Journal of Quantum Electronics 37*(3), 390–397. doi:10.1109/3.910448.

Lange, R., P. Seitz, A. Biber, and R. Schwarte (1999). Time-of-flight range imaging with a custom solid state image sensor. In H. J. Tiziani and P. K. Rastogi (Eds.), *Industrial Lasers and Inspection (EUROPTO Series)*, pp. 180–191. SPIE. doi:10.1117/12.360988.

Laurin, D. G., J. A. Beraldin, F. Blais, and M. Rioux (1999). UCL Single Sign-on. *Proc. SPIE 3707*, 278–289.

Lazaros, N., G. C. Sirakoulis, and A. Gasteratos (2008). Review of Stereo Vision Algorithms: From Software to Hardware. *International Journal of Optomechatronics 2*(4), 435–462. doi:10.1080/15599610802438680.

Lazebnik, S., C. Schmid, and J. Ponce (2006). Beyond Bags of Features: Spatial Pyramid Matching for Recognizing Natural Scene Categories. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pp. 2169–2178. IEEE. doi:10.1109/CVPR.2006.68.

Lee, D. H., K. M. Lee, and S. U. Lee (2008). Fusion of Lidar and Imagery for Reliable Building Extraction. *Photogrammetric Engineering & Remote Sensing 74*(2), 215–225. doi:10.14358/pers.74.2.215.

Lee, J., Y.-J. Kim, K. Lee, S. Lee, and S.-W. Kim (2010). Time-of-flight measurement with femtosecond light pulses. *Nature Photonics 4*(10), 716–720. doi:10.1038/nphoton.2010.175.

Lhuillier, M. (1998). Efficient Dense Matching for Textured Scenes Using Region Growing. In *Procedings of the British Machine Vision Conference 1998*, pp. 70.1–70.10. British Machine Vision Association. doi:10.5244/c.12.70.

Liang, C.-K., C.-C. Cheng, Y.-C. Lai, L.-G. Chen, and H. H. Chen (2011). Hardware-Efficient Belief Propagation. *Circuits and Systems for Video Technology, IEEE Transactions on 21*(5), 525–537. doi:10.1109/TCSVT.2011.2125570.

Lim, J. (2009). Optimized projection pattern supplementing stereo systems. In *Robotics and Automation ICRA, IEEE International Conference on*, pp. 2823–2829. doi:10.1109/ROBOT.2009.5152786.

Lindner, M., I. Schiller, A. Kolb, and R. Koch (2010). Time-of-Flight sensor calibration for accurate range sensing. *Computer Vision and Image Understanding 114*(12), 1318–1328. doi:10.1016/j.cviu.2009.11.002.

Longuet-Higgins, H. C. (1981). A computer algorithm for reconstructing a scene from two projections. *Nature 293*(5828), 133–135. doi:10.1038/293133a0.

Loop, C. and Z. Zhang (1999). Computing rectifying homographies for stereo vision. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pp. 1–131 Vol. 1. IEEE Comput. Soc. doi:10.1109/CVPR.1999.786928.

Lowe, D. G. (2004). Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision 60*(2), 91–110. doi:10.1023/B:VISI.0000029664.99615.94.

Malamas, E. N., E. G. M. Petrakis, M. Zervakis, L. Petit, and J.-D. Legat (2003). A survey on industrial vision systems, applications and tools. *IMAVIS 21*(2), 171–188. doi:10.1016/S0262-8856(02)00152-X.

Marquardt, D. W. (1963). An Algorithm for Least-Squares Estimation of Nonlinear Parameters. *Journal of the Society for Industrial and Applied Mathematics 11*(2), 431–441. doi:10.2307/2098941.

Marr, D. and T. Poggio (1979). A Computational Theory of Human Stereo Vision. *Proceedings of the Royal Society of London B: Biological Sciences 204*(1156), 301–328. doi:10.1098/rspb.1979.0029.

Marr, D., T. Poggio, E. C. Hildreth, and W. E. L. Grimson (1991). A computational theory of human stereo vision. In *From the Retina to the Neocortex*, pp. 263–295. Boston, MA: Birkhäuser Boston. doi:10.1007/978-1-4684-6775-8_11.

Massaro, R. D., J. E. Anderson, J. D. Nelson, and J. D. Edwards (2014). A Comparative Study between Frequency-Modulated Continous Wave LADAR and Linear LiDAR. *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences XL-1*, 233–239. doi:10.5194/isprsarchives-XL-1-233-2014.

McCarthy, T., S. Fotheringham, and M. Charlton (2007). Integration of LIDAR and stereoscopic imagery for route corridor surveying. *Mobile Mapping Technology 37*, 1125–1130.

Mirzaei, F. M., D. G. Kottas, and S. I. Roumeliotis (2012). 3D LIDAR-camera intrinsic and extrinsic calibration: Identifiability and analytical least-squares-based initialization. *The International Journal of Robotics Research 31*(4), 452–467. doi:10.1177/0278364911435689.

Molinier, T., D. Fofi, J. Salvi, Y. Fougerolle, and P. Gorria (2008). Projector view synthesis and virtual texturing. *2nd International Topical Meeting on Optical Sensing and Artificial Vision*.

Mroz, F. and T. P. Breckon (2012). An empirical comparison of real-time dense stereo approaches for use in the automotive environment. *EURASIP Journal on Image and Video Processing 2012*(1), 1–19. doi:10.1186/1687-5281-2012-13.

Muller, J.-P. and A. Anthony (1987). Synergistic ranging systems for remote inspection of industrial objects. *Proceedings 2nd Industrial and Engineering Survey Conference.*, 270–291.

Muller, J.-P., A. Anthony, A. T. Brown, A. T. Deacon, S. A. Kennedy, P. M. Montgomery, G. W. Robertson, and D. M. Watson (1988). Real-Time Stereo Matching Using Transputer Arrays for Close-Range Applications. *IAPR Workshop on CV - Special Hardware and Industrial Applications*, 45–49.

Murray, D. and J. J. Little (2005). Patchlets: Representing Stereo Vision Data with Surface Elements. *Application of Computer Vision, 2005. WACV/MOTIONS '05 Volume 1. Seventh IEEE Workshops on 1*, 192–199. doi:10.1109/ACVMOT.2005.90.

Naroditsky, O., A. Patterson, and K. Daniilidis (2011). Automatic alignment of a camera with a line scan LIDAR system. *ICRA*, 3429–3434. doi:10.1109/ICRA.2011.5980513.

Nelson, R., W. Krabill, and G. MacLean (1984). Determining forest canopy characteristics using airborne laser data. *Remote Sensing of Environment 15*(3), 201–212. doi:10.1016/0034-4257(84)90031-2.

Newman, P., G. Sibley, M. Smith, M. Cummins, A. Harrison, C. Mei, I. Posner, R. Shade, D. Schroeter, L. Murphy, W. Churchill, D. Cole, and I. Reid (2009). Navigating, Recognizing and Describing Urban Spaces With Vision and Lasers. *The International Journal of Robotics Research 28*(11-12), 1406–1433. doi:10.1177/0278364909341483.

Newman, T. S. and A. K. Jain (1995). A Survey of Automated Visual Inspection. *Computer Vision and Image Understanding 61*(2), 231–262. doi:10.1006/cviu.1995.1017.

Niclass, C. and E. Charbon (2005). A single photon detector array with 64-64 resolution and millimetric depth accuracy for 3D imaging. In *ISSCC. 2005 IEEE International Digest of Technical Papers. Solid-State Circuits Conference, 2005.*, pp. 364–366. IEEE. doi:10.1109/ISSCC.2005.1494020.

Niclass, C., M. Soga, H. Matsubara, and S. Kato (2011). A 100m-Range 10-Frame/s 340x96-Pixel Time-of-Flight Depth Sensor in 0.18μm CMOS . In *ESSCIRC (ESSCIRC), 2011 Proceedings of the IS -*, pp. 107–110. IEEE. doi:10.1109/ISSCC.2013.6487827.

Ohta, Y. and T. Kanade (1985). Stereo by Intra- and Inter-Scanline Search Using Dynamic Programming. *IEEE Transactions on Pattern Analysis and Machine Intelligence PAMI-7*(2), 139–154. doi:10.1109/TPAMI.1985.4767639.

Ojala, T., M. Pietikäinen, and D. Harwood (1996). A comparative study of texture measures with classification based on featured distributions. *Pattern Recognition 29*(1), 51–59. doi:10.1016/0031-3203(95)00067-4.

O'Shea, D. C., T. J. Suleski, A. D. Kathman, and D. W. Prather (2003). *Diffractive Optics Design, Fabrication, and Test* . Bellingham, USA: SPIE Press. doi:10.1117/3.527861.

Otto, G. P. and T. Chau (1989). 'Region-growing' algorithm for matching of terrain images. *Image and Vision Computing 7*(2), 83–94. doi:10.1016/0262-8856(89)90001-2.

Park, Y., S. Yun, C. Won, K. Cho, K. Um, and S. Sim (2014). Calibration between Color Camera and 3D LIDAR Instruments with a Polygonal Planar Board. *Sensors 14*(3), 5333–5353. doi:10.3390/s140305333.

Pereira do Carmo, J. (2011). Imaging LIDAR technology developments at the European Space Agency. In *Proceedings of the SPIE*, pp. 800129. European Space Agency, Netherlands. doi:10.1117/12.891945.

Petrou, M. and C. Petrou (2010). *Image Processing The Fundamentals.* Chichester, UK: John Wiley & Sons.

Pierrottet, D., F. Amzajerdian, L. Petway, B. Barnes, G. Lockard, and M. Rubio (2008). Linear FMCW Laser Radar for Precision Range and Vector Velocity Measurements. *MRS Proceedings 1076*, 1076–K04–06. doi:10.1557/PROC-1076-K04-06.

Potmesil, M. and I. Chakravarty (1982). Synthetic Image Generation with a Lens and Aperture Camera Model. *ACM Transactions on Graphics (TOG) 1*(2), 85–108. doi:10.1145/357299.357300.

Press, W. H. (2007). *Numerical Recipes 3rd Edition*. The Art of Scientific Computing. Cambridge University Press.

Quan, L. and Z. Lan (1999). Linear N-point camera pose determination. *Pattern Analysis and Machine Intelligence, IEEE Transactions on 21*(8), 774–780. doi:10.1109/34.784291.

Romero, L., A. Núñez, S. Bravo, and L. Gamboa (2004). Fusing a Laser Range Finder and a Stereo Vision System to Detect Obstacles in 3D. In J. González, C. Lemaître, and C. Reyes (Eds.), *LNAI*, pp. 555–561. Berlin Heidelberg: Springer Berlin Heidelberg. doi:10.1007/978-3-540-30498-2_55.

Sach, L. T., K. Atsuta, K. Hamamoto, and S. Kondo (2009). A Robust Stereo Matching Method for Low Texture Stereo Images. In *Computing and Communication Technologies, 2009. RIVF '09. International Conference on IS - SN - VO -*, pp. 1–8. IEEE. doi:10.1109/RIVF.2009.5174612.

Scharstein, D., H. Hirschmüller, Y. Kitajima, G. Krathwohl, N. Nešić, X. Wang, and P. Westling (2014). High-Resolution Stereo Datasets with Subpixel-Accurate Ground Truth. In X. Jiang, J. Hornegger, and R. Koch (Eds.), *Lecture Notes in Computer Science*, pp. 31–42–42. Cham: Springer International Publishing. doi:10.1007/978-3-319-11752-2_3.

Scharstein, D. and C. Pal (2007). Learning Conditional Random Fields for Stereo. In *Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on*, pp. 1–8. IEEE. doi:10.1109/CVPR.2007.383191.

Scharstein, D. and R. Szeliski (2002). A Taxonomy and Evaluation of Dense Two-Frame Stereo Correspondence Algorithms. *International Journal of Computer Vision 47*(1/3), 7–42. doi:10.1023/A:1014573219977.

Scharstein, D. and R. Szeliski (2003). High-accuracy stereo depth maps using structured light. In *Pattern Recognition (ICPR), 21st International Conference on*, pp. I–195–I–202. IEEE. doi:10.1109/CVPR.2003.1211354.

Scholdstrom, R. (1969). The AGA model 8 laser geodimeter. *Australian Surveyor 22*(6), 436–442. doi:10.1080/00050326.1969.10440155.

Schwarte, R., Z. Xu, H.-G. Heinol, J. Olk, and B. Buxbaum (1997). New optical four-quadrant phase detector integrated into a photogate array for small and precise 3D cameras. In R. N. Ellson and J. H. Nurre (Eds.), *Electronic Imaging '97*, pp. 119–128. SPIE. doi:10.1117/12.269749.

Shan, J. and C. K. Toth (2008). *Topographic laser ranging and scanning: principles and processing*. Boca Raton: CRC press.

Shannon, C. E. (1948). A mathematical theory of communication. *The Bell System Technical Journal 27*(3), 379–423. doi:10.1002/j.1538-7305.1948.tb01338.x.

Shin, D. and J.-P. Muller (2012). Progressively weighted affine adaptive correlation matching for quasi-dense 3D reconstruction. *Pattern Recognition 45*(10), 3795–3809. doi:10.1016/j.patcog.2012.03.023.

Song, Y., C. Glasbey, G. A. M. Heijden, G. Polder, and J. A. Dieleman (2011). Combining Stereo and Time-of-Flight Images with Application to Automatic Plant Phenotyping. In A. Heyden and F. Kahl (Eds.), *Lecture Notes in Computer Science*, pp. 467–478–478. Berlin, Heidelberg: Springer Berlin Heidelberg. doi:10.1007/978-3-642-21227-7_44.

Srinivasan, V. and R. Lumia (1989). A pseudo-interferometric laser range finder for robot applications. *IEEE Transactions on Robotics and Automation 5*(1), 98–105.

Steinberg, A. N. and C. L. Bowman (2004). Rethinking the JDL data fusion levels. *NSSDF JHAPL 38*, 39.

Stettner, R. (2010). Compact 3D flash lidar video cameras and applications. In M. D. Turner and G. W. Kamerman (Eds.), *SPIE Defense, Security, and Sensing*, pp. 768405–768405–8. SPIE. doi:10.1117/12.851831.

Sun, J., N.-N. Zheng, and H.-Y. Shum (2003). Stereo matching using belief propagation. *IEEE Transactions on Pattern Analysis and Machine Intelligence 25*(7), 787–800. doi:10.1109/TPAMI.2003.1206509.

Sun, W. and J. R. Cooperstock (2005). Requirements for Camera Calibration: Must Accuracy Come with a High Price? In *Application of Computer Vision, 2005. WACV/MOTIONS '05 Volume 1. Seventh IEEE Workshops on IS - SN -*, pp. 356–361. IEEE. doi:10.1109/ACVMOT.2005.102.

Sutherland, I. E. (1974). Three-dimensional data input by tablet. *Proceedings of the IEEE 62*(4), 453–461. doi:10.1109/PROC.1974.9449.

Tamura, H., S. Mori, and T. Yamawaki (1978). Textural Features Corresponding to Visual Perception. *Systems, Man and Cybernetics, IEEE Transactions on 8*(6), 460–473. doi:10.1109/TSMC.1978.4309999.

Thornhill, G. D., D. A. Rothery, J. B. Murray, A. C. Cook, T. Day, J.-P. Muller, and J. C. Iliffe (1993). Topography of Apollinaris Patera and Ma'adim Vallis: Automated extraction of digital elevation models. *Journal of Geophysical Research: Biogeosciences 98*(E12), 23581–23587. doi:10.1029/93JE03153.

Thrun, S. (2006). Winning the DARPA Grand Challenge: A Robot Race through the Mojave Desert. In *Automated Software Engineering, 2006. ASE '06. 21st IEEE/ACM International Conference on*. IEEE Computer Society. doi:10.1109/ASE.2006.74.

Toutin, T. (2004). Comparison of stereo-extracted DTM from different high-resolution sensors: SPOT-5, EROS-a, IKONOS-II, and QuickBird. *IEEE Transactions on Geoscience and Remote Sensing 42*(10), 2121–2129. doi:10.1109/TGRS.2004.834641.

Tsai, R. (1987). A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses. *IEEE Journal on Robotics and Automation 3*(4), 323–344. doi:10.1109/JRA.1987.1087109.

Vasconcelos, F., J. P. Barreto, and U. Nunes (2012). A Minimal Solution for the Extrinsic Calibration of a Camera and a Laser-Rangefinder. *Pattern Analysis and Machine Intelligence, IEEE Transactions on 34*(11), 2097–2107. doi:10.1109/TPAMI.2012.18.

Veitch-Michaelis, J., J.-P. Muller, J. Storey, D. Walton, and M. Foster (2015). Data Fusion of Lidar Into a Region Growing Stereo Algorithm. *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences XL-4-W5*(1), 107–112. doi:10.5194/isprsarchives-XL-4-W5-107-2015.

Veitch-Michaelis, J., J.-P. Muller, D. Walton, J. Storey, M. Foster, and B. Crutchley (2016). Enhancement of Stereo Imagery by Artificial Texture Projection Generated Using a Lidar. *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences XLI-B5*, 599–606. doi:10.5194/isprs-archives-XLI-B5-599-2016.

Veksler, O. (2005). Stereo correspondence by dynamic programming on a tree. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, pp. 384–390 vol. 2. doi:10.1109/CVPR.2005.334.

Wald, L. (1999). Some terms of reference in data fusion. *IEEE Transactions on Geoscience and Remote Sensing 37*(3), 1190–1193. doi:10.1109/36.763269.

Wang, L., M. Liao, M. Gong, R. Yang, and D. Nister (2006). High-Quality Real-Time Stereo Using Adaptive Cost Aggregation and Dynamic Programming. In *3D Data Processing, Visualization, and Transmission, Third International Symposium on IS - SN - VO -*, pp. 798–805. IEEE. doi:10.1109/3DPVT.2006.75.

Wehr, A. and U. Lohr (1999). Airborne laser scanning—an introduction and overview. *ISPRS Journal of Photogrammetry and Remote Sensing 54*(2-3), 68–82. doi:10.1016/S0924-2716(99)00011-8.

Wheatstone, C. (1838). Contributions to the Physiology of Vision. Part the First. On Some Remarkable, and Hitherto Unobserved, Phenomena of Binocular Vision. *Philosophical Transactions of the Royal Society of London 128*(0), 371–394. doi:10.1098/rstl.1838.0019.

Woodfill, J. I., G. Gordon, and R. Buck (2004). Tyzx DeepSea High Speed Stereo Vision System. In *Computer Vision and Pattern Recognition Workshop, 2004. CVPRW '04. Conference on IS - SN - VO -*, pp. 41–41. IEEE. doi:10.1109/CVPR.2004.469.

Yang, H., X. Liu, and I. Patras (2012). A simple and effective extrinsic calibration method of a camera and a single line scanning lidar. In *Pattern Recognition (ICPR), 21st International Conference on*, Tsukuba, pp. 1439–1442. IEEE.

Zabih, R. and J. Woodfill (1994). Non-parametric local transforms for computing visual correspondence. In *Computer Vision—ECCV'94*, pp. 151–158. Berlin/Heidelberg: Springer-Verlag. doi:10.1007/BFb0028345.

Zbontar, J. and Y. LeCun (2016). Stereo matching by training a convolutional neural network to compare image patches. *Journal of Machine Learning Research*.

Zhang, C. and Z. Zhang (2011). Calibration between depth and color sensors for commodity depth cameras. In *Multimedia and Expo (ICME), 2011 IEEE International Conference on IS -*, pp. 1–6. doi:10.1109/ICME.2011.6012191.

Zhang, Q. and R. Pless (2004). Extrinsic calibration of a camera and laser range finder (improves camera calibration). In *Intelligent Robots and Systems, 2004. (IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on IS - SN -*, pp. 2301–2306 vol.3. IEEE. doi:10.1109/IROS.2004.1389752.

Zhang, S., C. Wang, and S. C. Chan (2013). A New High Resolution Depth Map Estimation System Using Stereo Vision and Kinect Depth Sensing. *Journal of Signal Processing Systems 79*(1), 19–31. doi:10.1007/s11265-013-0821-8.

Zhang, Z. (2000). A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence 22*(11), 1330–1334. doi:10.1109/34.888718.

Zhang, Z. (2004). Camera calibration with one-dimensional objects. *Pattern Analysis and Machine Intelligence, IEEE Transactions on 26*(7), 892–899. doi:10.1109/TPAMI.2004.21.

Zhu, J., L. Wang, R. Yang, J. E. Davis, and Z. Pan. Reliability Fusion of Time-of-Flight Depth and Stereo Geometry for High Quality Depth Maps. *IEEE Transactions on Pattern Analysis and Machine Intelligence 33*(7), 1400–1414. doi:10.1109/TPAMI.2010.172.

Zhu, J., L. Wang, R. Yang, J. E. Davis, and Z. Pan (2011). Reliability Fusion of Time-of-Flight Depth and Stereo Geometry for High Quality Depth Maps. *IEEE Transactions on Pattern Analysis and Machine Intelligence 33*(7), 1400–1414. doi:10.1109/TPAMI.2010.172.

Zitnick, C. L. and T. Kanade (2000). A cooperative algorithm for stereo matching and occlusion detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence 22*(7), 675–684. doi: 10.1109/34.865184.