Single and Cooperative 2D/3D Image Mosaicing

PhD Thesis

Saad Ali Imran
s.a.imran@cranfield.ac.uk

Cranfield UNIVERSITY

Single and Cooperative 2D/3D Image Mosaicing

by

Saad Ali Imran

A thesis submitted in partial fulfilment for the

degree of

Doctor of Philosophy

in the

Department of Informatics and Systems Engineering

Cranfield University

Date of Submission: February 2014
Supervisor: Dr. Nabil Aouf

Abstract

This thesis investigates robust and fast methods for single and cooperative 2D/3D image mosaicing to enhance field of view of images by joining them together. Image mosaicing is underlined by the process of image registration and a significant portion of the contributions of this work are dedicated to it.

Image features are identified as a solution to the problem of image registration that uses feature-to-feature matching between images to solve for inter-image transformations. We have developed a novel two signature distribution based feature descriptor that combines grey level gradients and a colour histogram. This descriptor is robust to illumination changes and shows better matching accuracy compared to state of the art. Furthermore, we introduce a feature clustering technique that uses colour codes assigned to each feature to group them together. This allows fast and accurate feature matching as the search space is reduced.

Taking into account feature location uncertainty we have introduced a novel information fusion technique to reduce this error by covariance intersection. This reduced error location is consequently fed to an H_∞ filter taking into account system uncertainty for parameter estimation. We show that this technique outperforms costly nonlinear optimisation techniques. We have also developed a novel coupled filtering scheme based on H_∞ filtering that estimates inter-image geometric and photometric transformations simultaneously. This is shown to perform better than standard least square techniques. Furthermore, we have introduced time varying parameter estimation using recursive techniques that facilitate in tracking changing parameters of inter-image transformations, suitable for image mosaicing between moving platforms.

A method for rapid 3D scene reconstruction is developed that uses homographic lines between images for semi-dense pixel matching. Triangular meshes are then used for a complete visualisation of the scene and to fill in the gaps. To tackle cooperative mosaicing scenarios, additional methods are presented that include descriptor compression using principal components and 3D scene merging using the trifocal tensor.

Capabilities of the proposed techniques are illustrated with real world images.

Acknowledgements

I am grateful to the following for contributing to this thesis in one way or another.

To Dr. Nabil Aouf for his guidance, motivation and technical advice. Thank you for believing in me and giving me an opportunity to conduct this work.

To my committee members, especially Professor Mark Richardson for his advice.

To Cranfield Univeristy and the Shrivenham Defence Academy for providing an environment conducive to research. To the EPSRC and BAE Systems for funding my research.

To Peter Kovesi for his MATLAB scripts and to the people of OpenCV for their C/C++ libraries.

To Diego Rodriguez and Lounis Chermak for their support through difficult times in research and otherwise. Without you this experience would have been less fulfilling. A very special thanks to Diego Rodriguez for proof reading the first draft of this thesis.

To my lab fellows for their kindness. A special thank you to Tarek Mouats and Raheem Larabi for making those Champions League nights even more fun.

To friends and relatives who have kept me in their prayers.

To my parents who, although being so far away, have been with me every step of the way. Thank you for your support and immense love. To my sisters and Faiqa Nayyar for their support.

To my dearest wife Affia Wahab. Thank you for your encouraging words and enduring patience. I will make up for all the times you have played second fiddle to my PhD. I promise!

Shuker Alhamdulillah.

Contents

List of Figures

List of Tables

List of Publications

- S. Imran and N. Aouf. A recursive least squares solution for recovering robust planar homographies. *12th Annual Conference on Towards Autonomous Robotic Systems*, pages 36-45, September 2011.
- S. Imran and N. Aouf. Robust L_∞ homography estimation using reduced image feature covariances from an RGB image. *Journal of Electronic imaging*, 21(4), 2012.
- S. Imran and N. Aouf. A robust two signature descriptor with orientation bins and colour codes for enhanced feature matching. *IEEE International Conference on Systems, Man and Cybernetics*, October 2013.
- S. Imran and N. Aouf. Photo-geometric registration via a coupled L_∞ filter. *IEEE International Conference on Robotics and Biomimetics*, December 2013.
- S. Imran and N. Aouf. Bundle adjustment and Kalman filtering for homography estimation. *IEEE International Conference on Robotics and Biomimetics*, December 2013.

Abbreviations

FOV	Field of View
dof	degree of freedom
CCD	Charged Couple Device
SVD	Singular Value Decomposition
RanSaC	Random Sample Consensus
LM	Levenberg Marquardt
CC	Cross Correlation
1SD	1 Signature Descriptor
2SD	2 Signature Descriptor
CSHOT	Combined Signature of Histogram orienTations
OSR	Open Search Region
RSR	Reduced Search Region
RLS	Recursive Least Squares
CI	Covariance Intersection
DLT	Direct Linear Transform
SOCP	Second Order Cone Programming
BA	Bundle Adjustment
IMU	Intertial Measurement Unit
DT	Delaunay Triangulation
PCA	Principal Component Analysis

To my Grandfather

1
Introduction

This thesis explores 2D and 3D image mosaicing in context of single and cooperating autonomous ground vehicles. Mosaics enhance limited views of individual images by joining them together for greater scene information. Applications of 2D and 3D mosaics include surveillance in and outside of military contexts, environmental modelling and a relatively new application called 3D photo tourism. The aim of this work is to develop rapid, robust and automated techniques for accurate 2D and 3D image mosaicing in the presence of feature location uncertainty and in the context of cooperating views and platforms.

The process of mosaicing is underpinned by image registration. This includes model estimation from feature-to-feature mapping. A significant portion of the thesis is dedicated to this problem and novel techniques based on colour and the recursive H_∞ filter are proposed for robust and fast estimation. Rapid scene reconstruction in projective 3D space is also tackled in a cooperative and non-cooperative scenario.

A key point in the thesis is the assumption of a completely uncalibrated camera unless stated otherwise. No prior knowledge of the camera parameters, its motion, optics or photometric characteristics is assumed. The capabilities of the techniques are illustrated with many real world image examples.

Five published and/or accepted publications are derived from this thesis and more are under way exploring latest developments from Chapter 5 and 6.

1.1 Background

Images acquired from a mobile autonomous system represent an efficient way to conduct environmental surveillance. This is true for both military and civilian applications, since they provide an end user with useful information of the environment from which important decisions, like target classification, can be made.

However, the information content of these images is limited and susceptible to noise. Soda straw field of views (FOV), similar to looking through a straw, and disorienting rotations are a common problem. Collectively, these limitations make it a demanding task for a user to correctly identify objects of interest in an image. Over the years, many techniques have been proposed to overcome this problem and enhance image detail [1].

Inherent FOV restrictions are always an underlying issue. A partial solution is to use a fish eye lens, which although capable of capturing larger scenes, is subject to substantial distortions, as shown in Figure 1.1. An alternative solution is to use image mosaicing. Mosaicing is a construction of a larger scene with a number of smaller images. It helps to overcome issues of limited and corrupted information within an image. As an example, consider Figure 1.2. It shows two overlapping images of an outdoor scene containing an aircraft. Separately they give a limited sense of the entire scene. However, by mosaicing the two together, we extend the FOV and get an enlarged view of the environment, shown in Figure 1.3. Now spotting an approaching threat to the aircraft from the left hand side is easier.

Extending the context of surveillance, we can use image mosaics to overcome the issues of limited FOVs and provide the end user with a larger view of the environment making it easier to spot objects of interest. A single platform can achieve this objective with a certain level of efficiency and reliability, but its capabilities are still limited. However, a number of such platforms operating in cooperation can greatly increase mission success and allow for more complex missions to be undertaken.

Historically, image mosaics were used to construct maps as shown in Figure 1.4. Then images were manually aligned to fit together (the seam where the images

Figure 1.1: *View of Canary Wharf, London taken from a fish eye camera [2]. The distortions are obvious, making it difficult for an automated system to detect and classify objects.*

(a) (b)

Figure 1.2: *Overlapping images from different two view points.*

Figure 1.3: *Image mosaic of Figure 1.2(a) and 1.2(b). Notice the extended field of view.*

meet is quite visible). However, thanks to the emergence of modern computing this task can now be automated, though finding accurate transformations between images is still a difficult task. Issues of relating images that include finding accurate correspondences between images, data outliers, occlusion, illumination changes and measurement error of control points make it a complex problem to solve.

1.1.1 Research Statement

In light of this background we set the research theme for this thesis as

to develop robust and fast 2D and 3D image mosaicing techniques to enhance overall image information content in the presence of measurement error of control points and variation in illumination between images. Furthermore, to show applicability of these techniques to cooperating platforms.

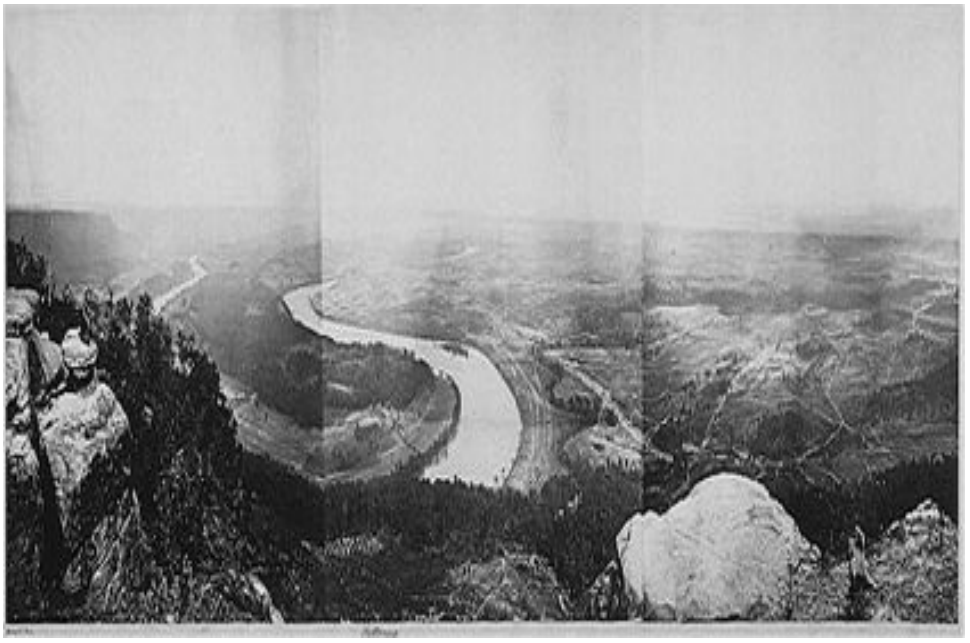


Figure 1.4: Map manually aligned together. Notice the visible seams.

1.2 Related Work

In this section, we include literature relevant to image registration, image mosaics and cooperative surveillance, all major themes of the thesis.

Image Registration Image mosaicing is underlined by the problem of image registration that includes finding the geometric transformation that relates two or more images together. Here, we highlight these models and the techniques to estimate them.

The most general transformation model relating two images captured from a camera under general motion, is an 8 degree of freedom (dof) planar projective transformation, also known as the homography. For special cases, i.e., special camera motions, the mapping between two images is simpler than a general homography and can be estimated more reliably and quickly.

For camera motion parallel to the image plane and rotating about the principal axis, images are related by a 4 dof similarity transform. Such motion is seen in document scanning and satellite imagery [3, 4]. Scenes imaged under telephoto viewing conditions, where perspective effects are negligible, are related by a 6 dof

affine transformation [4]. All these transformations are valid in the correct viewing conditions. Citing reasons for numerical stability, some authors have chosen to use biquadratic transformations, by approximating the homography by a Taylor expansion [5, 6]. This has 12 dof but is unable to correctly model perspective effects [1].

Solving for the transformation model is done in one of two ways, either using a feature based method or a direct method [7]. Feature based methods use point-to-point mapping to estimate the transformation and are more common because of their speed and invariance to geometric and photometric changes [1, 8, 9]. They are, also, used exclusively throughout this thesis.

The challenge in efficient feature based registration is extracting and matching image features between views. Various techniques have been developed over the years as possible solutions. Examples include [10–18]. Chapter 3 gives a detail account on various extractors and matching techniques.

An experimental study on 2D homography estimation is given in [19], where various transformation models are computed. It is concluded, that for a limited number of correspondences a more restricted homography, i.e, similarity of affine transformation can be used. Taking advantage of invariant properties of imaging geometry, [20] have introduced a convex hull based registration technique that uses hull vertices to seek for initial correspondences between images. Although promising, the proposed methodology fails when difference between the scenes is large.

Use of Kalman filters for parameter estimation in vision is proposed in [21–23]. In [21], loop closing for autonomous localisation and mapping is used to drive down global positioning and alignment errors. In [22, 23], it is used for camera calibration and pose estimation.

False matches, commonly known as outliers, adversely affect estimates of the transformation. Techniques based on sample consensus offer an effective solution, as used in [10, 24]. The technique is explained in Chapter 2. Treating outliers using convex hulls of features is done in [25], where coherence matching is used instead of sample consensus to find dominant transformations. This technique, however, suffers with limited overlap of images.

Fundamental to feature based registration is image overlap [26]. Without it, correspondences cannot be established and consequently transformations are not computed. Aligning non-overlapping image sequences is done in [27]. The technique uses temporal alignment via time stamps of images to align two streams of images from two attached cameras. Good results are shown for limited overlap and no-overlapping image streams for limited camera movement. An adaptive correlation window is used in [28] within the area of overlap to overcome issues with small overlap. It uses direct methods computing inter-image transformation based on difference in intensities.

Besides image registration done in the spatial domain from features, there are also techniques based in the frequency domain [29–32]. Though these methods claim to be fast, they require significant overlap between images.

Image Mosaics Image mosaics enhance limited field of views of individual images and are mostly applied to images related by homographies [10, 24, 33]. Using these inter-image transformations, overlapping images of the same 3D scene are warped into the same coordinate frame resulting in information rich views.

An example of wide-surveillance from mosaicing of overlapping images is found in [10]. Using feature based methods, they are able to mosaic large number of images, over 2000 images together. Object recognition and tracking is then applied on top of the final mosaic. Similarly, globally consistent aerial mosaics are built in [13], also for surveillance. Here, images are taken from a downward looking camera onboard an airborne platform. Aerial mosaicing is done in [34] in real time using a bag of words algorithm for image representation. This allows loop closures ensuring global consistency of the mosaics. Mosaicing from wide-baseline cameras is done in [35]. Here, a technique capable of overcoming large parallax is introduced by integrating two robust affine invariant feature detectors. They show their technique to be effective in texture rich environments.

Mosaicing in realtime is undertaken in [36–38]. In [36], a miniature robot called "the scout", fitted with a video camera, is used to construct mosaics in real time to assist disaster missions. It uses feature based registration to build mosaics and transmits the mosaics wirelessly to an end user. In [37], feature tracking

techniques are used to estimate transformations for mosaicing and is part of a wider project including surveillance and tracking. Mosaic based navigation for autonomous underwater vehicles is done in [38]. Here, trajectories are defined on the mosaic for purposes of future navigation. Real time video mosaicing is acheived in [39] based on a simplified SIFT algorithm for feature matching where computation time is reduced by decreasing the number of octaves used for scale space generation. Though the time taken to construct the mosaics is not included.

Optical flow based mosaicing is done in [40] to construct mosaics in real-time. Here, an affine transformation model is used and implemented on a low-cost PC. It is worth noting that an issue with optical flow is its need for small gap between consecutive frames for accurate motion estimation. Though, as the technique proposed in [40] is using an image stream from a video camera, this is not a major problem. Image streams taken under telephoto operations are mosaiced in [41] whilst preserving the 3D scene information. It is used, among other applications, for under car inspection.

It is possible to retrieve 3D scene information from image mosaics for use in inspection and model building as done in [41, 42]. In [42], 3D textured models of the scene are constructed from geometric information extracted from panoramic images.

Live photo-mosaicing has also been attempted recently from a cluster of wireless cameras as in [43]. Here, to overcome issues of limited communication bandwidth, a data aggregation technique is proposed that minimises data sent to the central processing unit. By retaining only required data, live updates of the mosaics is achieved and also the 3D point clouds can be projected onto the mosaics. A detailed communication cost analysis is also undertaken to find the least costly combination of cameras.

Instead of planar mosaics, as in [10, 44], spherical mosaics can also be built. The QuickTime VR software [45] is an example of the use of spherical mosaics to represent the view in every direction. Rendering in any direction on this representation allows a virtual reality application. High-resolution and multi-scale mosaics are built in [46] which are mapped on a cubic surface. Here, a pan-tilt-zoom camera is used with the camera parameters determined online. A Gaussian sphere is used in

[47] to represent the mosaic as it reduces projective distortions caused by a camera rotating around its focal point.

Multi-perspective image mosaics are also used to visualise long scenes. The case for such mosaics is put forward in [48], where it is claimed that multispectral panoramas provide a more complete FOV to encompass a scene. Here, an automatic multi perspective mosaicing tool using pushbroom cameras for image capture is also presented. A less costly multi-perspective panorama builder that is capable of running on mobile phones is introduced in [49]. In it, optical flow is applied for image registration and vertical strip extraction for panorama building. As each strip is captured from a slightly different viewpoint, the panorama exhibits multi-persepective characteristics. Multi-perspective panoramas of 3D models is proposed in [50] for application in computer generated imagery.

In addition to surveillance and virtual reality, applications of image mosaics include autonomous driving and driver assistance, as in [33]. Here, mosaicing in dynamic environments helps provide more information to the guidance and navigation unit of the autonomous vehicle, making it more effective. An application of image mosaics is also found in agricultural engineering. In [44], tree mosaicing and consequently seam processing is performed to determine when to apply pesticides to high-trees. Multispectral mosaicing is another avenue under consideration in [51]. It allows even more information to a user for example, thermal signatures, in addition to just imaging data.

Cooperative Navigation and Surveillance An advantage of cooperative surveillance is that it allows coverage of a wider area more easily. The cyber scout project uses autonomous robots, called "sentries" to scout an area and conduct surveillance and reconnaissance missions [37]. The sentries are fitted with 5 cameras that allow for this in addition to navigation and decision making architecture. Images acquired from the cameras are used to construct mosaics on which motion detection and consequently classification is done. A cooperative multi-sensor approach is proposed in [52] for surveillance. Here, an end user can task the system to monitor or track an object in the area of surveillance. The cooperative system then

sends appropriate commands to sensors in time and space to this end. The imaging system is capable of estimating the 3D positions of the object being tracked and conveying it back to the end user. To tackle issues of estimation error, [53] proposes fusion of location estimates from different views using filtering. It shows good results for a rotating camera platform mounted on a ground vehicle.

1.3 Overview and Principal Contributions

The remaining chapters with their principal contributions are as follows. Seven chapters, including the introduction and two appendices form this thesis. The contributions are included with the individual publications in which they appear. This is to highlight the novelty and applicability of the work. A list of publications is given in Section 1.4 of this chapter.

Chapter 2: Imaging Geometry

Concepts pertaining to digital imaging and multi-view geometry are introduced in this chapter. They include image formation, geometric transformation and mapping between images. These are necessary topics to solve image mosaicing and 3D scene reconstruction.

Chapter 3: Interest Points in Images

In this chapter, we introduce several techniques for fast and robust feature description and matching. Feature matches are used to estimate inter-image transformations from point-to-point mapping. The contributions include

- An analysis comparing correlation based techniques with relaxation methods. Published in conference paper-1: TAROS, 2011.
- Implementing a robust gradient histogram descriptor to a fast image feature detector. Published in conference paper-1: TAROS, 2011.

- Developing a new two signature feature descriptor constituting both intensity gradients and a colour histogram. It shows to outperform state of the art single signature descriptors. Accepted in conference paper-2: SMC, 2013.
- A novel feature coding technique allowing fast and reliable feature matching by bundling similar coloured features together. This assists in rapid image mosaicing. Accepted in conference paper-3: ROBIO, 2013.

Chapter 4: Robust Homography Estimation

Several novel techniques to tackle inter-image homography estimation are developed in this chapter. The homography is a transformation that relates points from one plane to another, or more importantly features from one image to another, a concept helpful in image mosaicing. The contributions include

- A novel recursive least squares technique that takes into account periodic measurements to estimate homography for use on board single mosaicing platforms. Published in conference paper-1: TAROS, 2012.
- A novel information fusion based technique to reduce image feature location uncertainty by applying covariance intersection on RGB images. Published in journal paper-1: JEI, 2013.
- A novel H_∞ filtering technique that takes into account feature location uncertainty to estimate the homography. Published in journal paper-1: JEI, 2012.
- A coupled filter simultaneously estimating geometric and photometric transformations between two images. Accepted in conference paper-3: ROBIO 2013.
- Filter based time varying homography estimation framework taking into account uncertainty for use on board single mosaicing platforms. Submitted to journal paper-2: JEI, 2013.
- Comparing global optimisation using bundle adjustment to the Kalman filter. Accepted in conference paper-4: ROBIO, 2013.

Chapter 5: Rapid 3D Reconstruction

Here, we introduce a couple of new techniques for rapid projective scene reconstruction in the 3D space. The contributions include

- A new inertial measurement unit assisted feature matching technique. It is included in this chapter since it is based in the 3D domain.
- An innovative homographic line semi-dense 3D scene reconstruction algorithm with Delaunay Visualisation to fill in empty spaces.

Chapter 6: Cooperative Mosaicing Methods

In this chapter, we propose methods to assist in cooperative mosaicing. These include

- A feature compression technique based on principal component analysis to allow communication of features between platforms with a low computational cost.
- A novel trifocal tensor based technique to merge 3D reconstructions from three cooperating platforms at a time.

Chapter 7: Conclusions and Future Work

Here, we conclude with main findings and contributions of the thesis and derive areas for future research.

1.4 Publihsed/Submitted Manuscripts

Journals

1. S. Imran and N. Aouf. Robust L_∞ homography estimation using reduced image feature covariances from an RGB image. *Journal of Electronic Imaging JEI*, 21(4), 2012. (published)

2. S. Imran and N. Aouf. L_∞ based estimation technique for time varying homographies with system uncertainty. *Journal of Electronic Imaging JEI*. (submitted)

Conferences

1. S. Imran and N. Aouf. A recursive least squares solution for recovering robust planar homographies. 12^{th} *Annual Conference on Towards Autonomous Robotic Systems TAROS*, 2011. (published)

2. S. Imran and N. Aouf. A robust two signature descriptor with orientation bins and colour codes for enhanced feature matching. *IEEE International Conference on Systems, Man and Cybernetics SMC*, 2013. (Accepted)

3. S. Imran and N. Aouf. Photo-geometric registration via a coupled L_∞ filter. *IEEE International Conference on Robotics and Biomimetics ROBIO*, 2013. (Accepted)

4. S. Imran and N. Aouf. Bundle Adjustment and Kalman filtering for homography estimation. *IEEE International Conference on Robotics and Biomimetics ROBIO*, 2013. (Accepted)

1.5 Software Tools

Listed below are tools used during the research.

- MATLAB [The Mathworks Official Website: http://www.mathworks.com/, N.d.]: MATLAB is a technical computing environment developed by The MathWorks group.
- C/C++: an intermediate-level language since it comprises both high-level and low-level language features. Developments in this study are made on the Xcode IDE.
- OpenCV: an opensource library of programming functions aimed at real-time computer vision applications developed by Intel and supported by Willow Garage.
- OpenGL: a cross-language, multi-platform application programming interface for rendering 2D and 3D graphics, initially released in 1992.

All codes developed during this research are written and tested on a Apple Macbook 2.16 GHz Intel Core 2 Duo with 2.5 GB SDRAM.

This document is written in LaTeX using TeXShop version 2.47 editor/compiler.

2
Imaging Geometry

The purpose of this chapter is to familiarise the reader with basic principles of image formation and multi-view geometry. These concepts form a basis for contributions made in the thesis. We start with image representation in a 2D array, where each index (pixel) has an associated intensity value. These pixels can be regarded as discrete points in space and are therefore subject to geometric transformations. This is followed by an introduction to image formation modelled by the pinhole camera which is a $\mathbb{R}^3 \rightarrow \mathbb{R}^2$ projective mapping. We then explore projective relationships between images of planar surfaces acquired from a moving or rotating camera. Two types transformations are discussed in fair detail. The first, maps points from one image plane to another and is known as the homography $\mathcal{H}$. This transformation is central to constructing image mosaics. The second transformation, maps a point from one image to an epipolar line in a corresponding image and is known as the fundamental matrix F. This matrix encapsulates the 3D structure of the scene and underpins the inverse mapping $\mathbb{R}^2 \rightarrow \mathbb{R}^3$. Furthermore, we show how to extract the camera matrices from F used later on in the thesis for 3D scene reconstruction.

The ideas presented here are by no means all encompassing, though they give a good grounding in image geometry. For a more in depth discussion, refer to [24, 54, 55].

Notation Points are represented as homogenous coordinates, i.e., a point (a, b) is $(a, b, 1)$ in homogenous coordinates. Conversely, a homogenous coordinate (a, b, c) is (ca, cb) in non-homogenous coordinates.

2.1 Image Representation

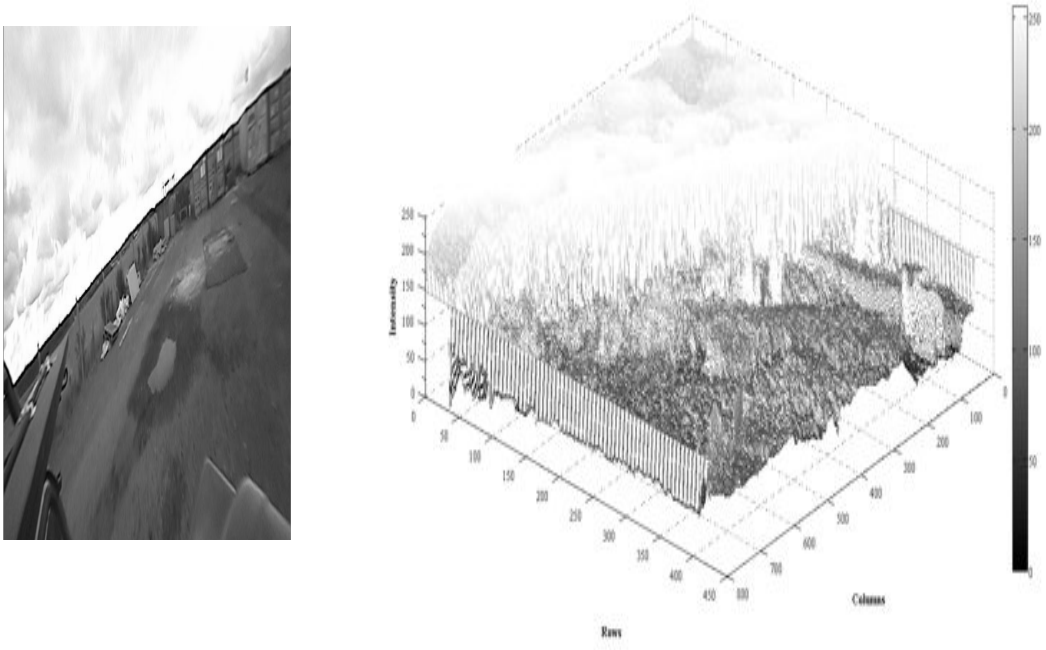
Here, we concern ourselves with discrete image representation. A discrete image of a scene is a 2D array filled with intensity values reflected from real world objects. Mathematically, it is a map $\mathcal{I}$ defined on a domain Ω of a two-dimensional surface, taking positive integer values $\mathbb{Z}$ ranging from $\{0-255\}$ in an 8bit representation. For a camera, Ω is a planar, rectangular region occupied by a photographic medium or in the case of a digital camera by a CCD (Charge Couple Device) sensor. The mapping $\mathcal{I}$ on a discrete Ω is given by

$$\mathcal{I} : \Omega \subset \mathbb{Z}^2 \rightarrow \mathbb{Z}_+;\ (x, y) \rightarrow \mathcal{I}(x, y) \tag{2.1}$$

where (x, y) are image coordinates in 2-space.

This mapping can be represented graphically as in Figure 2.1 for a grey scale image. The variation of intensity over Ω is clearly visible. Brighter pixels take higher values whereas darker pixels take lower ranging values. These values depend predominantly on ambient conditions of the scene, reflectance properties of the materials and the camera.

For colour images, a combination of 3 primary colours in three 2D overlapping arrays is used to define colour for each pixel. This is shown in Figure 2.2. Each array or channel corresponds to a primary colour. For an RGB colour image these are red, green and blue, and for an 8bit representation can take values ranging from $\{0-255\}$. A grey scale image is obtained from an RGB image by weighting the three channels. This though comes at the expense of image sharpness.



(a) (b)

Figure 2.1: *(a) Original image. (b) A 2D grey scale image plotted as a surface map. The rows and columns define* Ω*, whilst the intensity (z-axis) gives the range from* $\{0-255\}$*. N.B. The image is a BAE Systems test image.*

Figure 2.2: *Region of interest showing individual RGB values for each pixel. A combination of these individual RGB values gives colour to an image.*

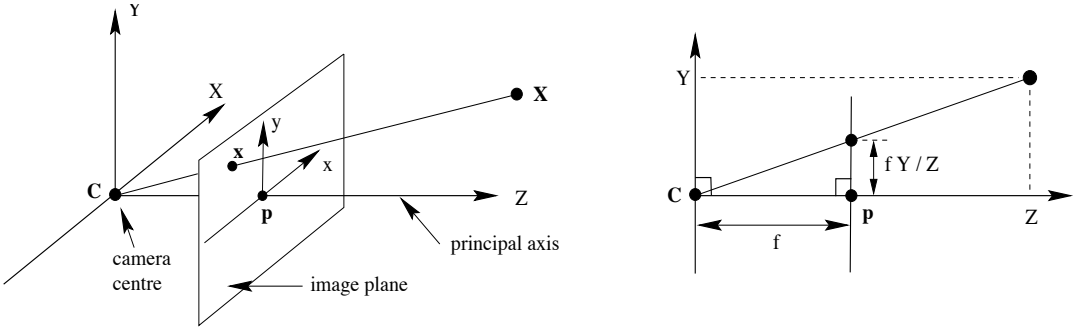


Figure 2.3: **C** *is the camera centre also known as the optical centre and* **p** *is the principal point. Note that the image plane is placed in front of the camera centre. Image reproduced from [24].*

2.2 The Pinhole Camera

A camera is a mapping π between the 3D world and a 2D image

$$\pi : \mathbb{R}^3 \rightarrow \mathbb{R}^2; \mathbf{X} \rightarrow \mathbf{x} \tag{2.2}$$

where $\mathbf{X} = (X, Y, Z)^{\mathrm{T}}$ are the coordinates in 3-space, $\mathbf{x} = (x, y)^{\mathrm{T}}$ are the corresponding coordinates in 2D and T is a transpose. It is composed of a lens or a set of lenses used to implement a controlled change in the direction of propagation of light on to a photographic medium. Models of such cameras can be complex to develop and understand. A less complicated model is based on the thin lens camera characterised by an aperture, a focal length f and the diameter d of the lens. A further idealisation of this leads to the pinhole model where the aperture of the lens is reduced to zero and all rays are forced through a single point, the centre of projection. This same model will be used in the thesis.

Refer to Figure 2.3, let the centre of projection, also known as the camera centre **C**, be the origin of a Euclidean coordinate system and consider the plane Z = f, which is called the image plane (focal plane). Now imagine a point **X** in space which under the pinhole camera is mapped to a point **x** on the image plane. It is where the line joining **X** to the centre of projection intersects the image plane. Via similar triangles we can deduce the perspective relationship that governs this mapping

$$x = f\frac{X}{Z}, \quad y = f\frac{Y}{Z} \tag{2.3}$$

where x and y represent coordinates on the image plane. The focal length is positive for a frontal model where the image plane is placed ahead of $\mathbf{C}$. In Figure 2.3, the line extending from $\mathbf{C}$ perpendicular to the image plane is the principal axis and the point of intersection is known as the principal point $\mathbf{p}$.

If the world and image points are represented in homogenous coordinates then the projection can be represented in compact form as

$$\mathbf{x} = KR\left[I \mid -\tilde{\mathbf{C}}\right]\mathbf{X}. \tag{2.4}$$

Here K is the 3×3 intrinsic parameter matrix of the camera and constitutes f, the principal offset which shifts the origin in the image plane to the upper left hand point and axial scaling factors to convert Euclidean coordinates into pixels. R and $\tilde{\mathbf{C}}$ are known as the camera's extrinsic parameters and they reconcile the camera coordinate frame with the world coordinate frame in which the world points are represented, shown in Figure 2.4. R is a 3×3 rotation matrix and $\tilde{\mathbf{C}}$ represents homogenous coordinates of the camera centre in the world frame.

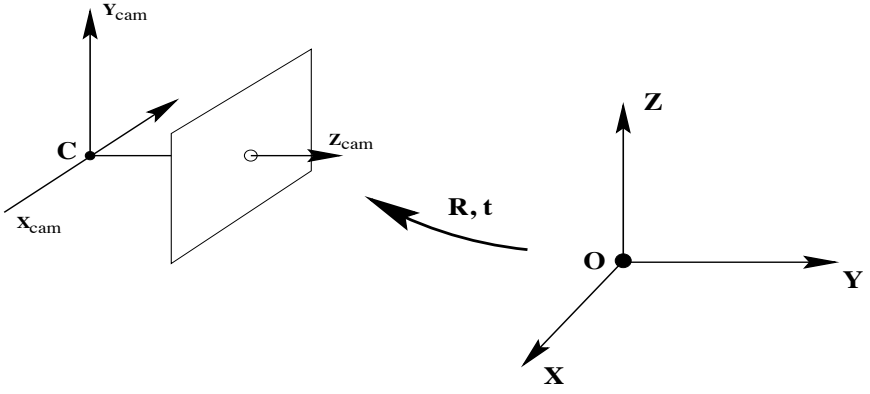


Figure 2.4: *Euclidean transformation from world to camera coordinate. Image reproduced from [24].*

Equation (2.4) can be written compactly as $\mathbf{x} = \mathrm{P}\mathbf{X}$, where P is known as the camera calibration matrix. It is a 3×4 matrix with 9 degrees of freedom: 3 for K, 3 for R and 3 for $\tilde{\mathbf{C}}$.

To conclude, in certain unusual cases the intrinsic parameter matrix is augmented with a parameter S that skews the pixel elements. This is the skew parameter and is set to zero for most normal cameras, as is the case here. It is also important to remember that real cameras will deviate from the standard pin hole model depending on the quality of the camera. This will lead to distortion in the projection, causing errors. We, in this work, do not show how to correct these errors but propose how to manage them efficiently.

2.3 Projective Transformations in 2-Space

Images of planar surfaces under camera motion or a rotating camera are related via projective transformations, Figure 2.5. Here we discuss two such transformations. The first, maps points from one image to another called the homography. This transformation is essential to constructing image mosaics as pixels from image can be joined to another. The second transformation, captures intrinsic projective geometry between two views and is known as the fundamental matrix. This transformation maps pixels from one view to lines in another and leads to projective 3D scene reconstruction from two or more views.

2.3.1 The Homography $\mathcal{H}$

The following definition reproduced from [24] gives a 2D point-to-point mapping in the perspective plane $\mathbb{P}^2$ for homogenous coordinates

Definition 2.1. $\mathbb{P}^2 \rightarrow \mathbb{P}^2$ is a projectivity if and only if there exists a non-singular 3×3 matrix $\mathcal{H}$ such that for any point in $\mathbb{P}^2$ represented by a vector $\mathbf{x}$ it is true that $\mathbf{x}^{'} = \mathcal{H}\mathbf{x}$.

Here $\mathbf{x}^{'} = \begin{pmatrix} x^{'} & y^{'} & z^{'} \end{pmatrix}^{\mathrm{T}}$ and $\mathbf{x} = (x \ y \ z)^{\mathrm{T}}$. Following on from this, we may define a homographic transformation on homogenous vectors as

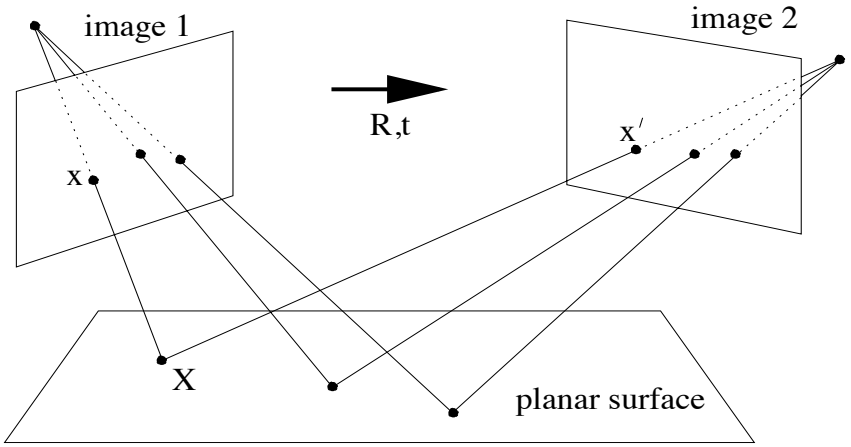


Figure 2.5: *Projective transformation exists between two views of a scene. Reproduced from [24].*

$$\begin{pmatrix} x' \\ y' \\ z' \end{pmatrix} = \begin{bmatrix} r_1 & r_2 & t_x \\ r_3 & r_4 & t_y \\ u & v & s \end{bmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} \tag{2.5}$$

where r_i are parameters of $\mathcal{H}$ which deal with rotation, t_x and t_y are translational parameters, s is the scaling parameter and u and v relate to projectivity. The transformation matrix can be called homogenous, since it is scale independent. Consequently, the matrix has 8 degrees of freedom. Figure 2.6 gives a projective transformation along with its invariant properties. See how properties of parallelism do not hold. Other transformations include affine, Euclidean and similarity transforms. In this study we solely deal with the projective transformation $\mathcal{H}$.

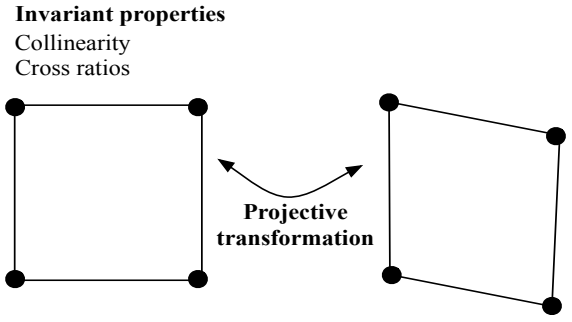


Figure 2.6: *Projective transformation of a quadrilateral. Notice how parallel lines are no longer parallel. Also, given are $\mathcal{H}$'s invariant properties.*

Estimating $\mathcal{H}$

Here, a basic algorithm to estimate $\mathcal{H}$ is outlined. If two images of the same scene are related through a projective mapping, there exist common features between them. Based on this, if corresponding features are identified between two views then through point-to-point mapping we can estimate a transformation.

We use image feature based methods for estimation as they are more robust to geometric and photometric deformations, not to mention faster compared to direct or area based techniques [1]. This is discussed in detail in Chapter 3.

We present here a linear method for estimating $\mathcal{H}$. Equation (2.5) can be rearranged with non-homogenous coordinates as

$$\begin{aligned} x^{'}\left(ux+vy+sz\right)-r_1x-r_2x-t_xz=0 \\ y^{'}\left(ux+vy+sz\right)-r_3x-r_4x-t_yz=0. \end{aligned} \tag{2.6}$$

Each correspondence produces two equations in $\mathcal{H}$. The above equation can be arranged into a design matrix $C\mathbf{h}=0$, where $\mathbf{h}$ is a vector of parameters of $\mathcal{H}$.

As already mentioned, the homography has 8 degrees of freedom, as it is defined up to scale. We therefore only require a minimum of 8 equations to obtain a solution for the transformation. This, therefore, implies that we need a minimum of 4 feature correspondences since each match produces two equations. Usually, an overdetermined solution is estimated as extracted features are prone to uncertainty and measurement error. This way the best transformation, the dominant homography, is determined. Also, care should be taken that features used for estimation are independent, that they are not collinear, since this could lead to a degenerate solution.

The parameters of $\mathcal{H}$ are then estimated using SVD (Singular Value Decomposition), which correspond to the unit vector with the smallest singular value of C subject to $||\mathbf{h}||=1$.

Outlier rejection - dealing with false matches False correspondences between images greatly affect the accuracy of the estimated transformation. For this

purpose an iterative method to reject outliers known as RanSaC (Random Sample and Consensus) algorithm is applied. This is a model estimator and its procedure is given below in Algorithm 2.1.

Algorithm 2.1: RanSaC algorithm for model estimation

objective: Reject false matches and estimate the dominant transformation, $\mathcal{H}$

output: Dominant $\mathcal{H}$ with a set of consistent feature matches

algorithm:

while *not stop and* $i < i_{max}$ **do**

- i:= i + 1;
- Make an initial estimate of $\mathcal{H}$ from 4 random feature correspondences from the sample;
- Then determine the set of matches that fall within an error threshold t of the model;
- **if** *If the consensus set is big enough* **then**
  - re-estimate $\mathcal{H}$ using this set;
  - stop:= true;

It is to be remembered that RanSaC will return the dominant transformation consistent with the majority of the sample set.

Non linear optimisation - refining $\mathcal{H}$ For an even more accurate estimation of $\mathcal{H}$, more often than not non-linear optimisation is performed. For this a cost function is required. The error used to quantify the quality of $\mathcal{H}$ used exclusively through out this thesis is the symmetric transfer error (reprojection error) given as

$$\sum_k d\left(\mathbf{x}_k, \mathcal{H}^{-1}\mathbf{x}_k^{'}\right)^2 + d\left(\mathbf{x}_k^{'}, \mathcal{H}\mathbf{x}_k\right)^2 \tag{2.7}$$

where d is the Euclidean distance and k is the number of feature correspondences.

The optimisation technique applied is the widely used Levenberg-Marquardt (LM) algorithm which interpolates between gradient descent and Gauss-Newton optimisation and is robust. LM will, however, only find local minimas and is sensitive to initial conditions. Please refer to Appendix A for a description of the algorithm.

Below is a general algorithm for estimating $\mathcal{H}$ with feature correspondences.

Algorithm 2.2: Estimating $\mathcal{H}$ between overlapping images using image features

objective: Estimate an accurate homography $\mathcal{H}$ between two views.

algorithm:

Extract common features between images;

for *i:= 1: no. of features in one image* **do**

  Find corresponding feature in second image;

Perform outlier removal using RanSaC;

Estimate the dominant $\mathcal{H}$ that maps one image to the other. If necessary, non-linear optimisation can be performed to better the estimate minimising the following cost function

$$\sum_k d\left(\mathbf{x}_k, \mathcal{H}^{-1}\mathbf{x}_k^{'}\right)^2 + d\left(\mathbf{x}_k^{'}, \mathcal{H}\mathbf{x}_k\right)^2 \tag{2.8}$$

2.3.2 The Fundamental Matrix F

This transformation captures a scene's intrinsic projective geometry and with slight algebraic manipulation can lead to projective scene reconstruction in 3-space. It, as opposed to $\mathcal{H}$ which maps points to points, maps points from one view to lines in another.

Consider Figure 2.7, here a point $\mathbf{X}$ in the 3D world is imaged in two views as $\mathbf{x}$ and $\mathbf{x}^{'}$. Taking one of these image points $\mathbf{x}$ and back-projecting it into 3 -space from $\mathbf{C}$, a ray is formed passing through $\mathbf{X}$. This ray is imaged in the second view as a line $\mathbf{l}^{'}$, and as $\mathbf{X}$ lies on this ray, its image in the second view must lie on $\mathbf{l}^{'}$. These lines are known as epipolar lines since they intersect the epipole. The epipoles themselves are an intersection of the line joining the camera centres with the image plane, Figure 2.7. An advantage of this epipolar geometry is that it helps in establishing correspondences, as the search space is constrained to a line. This is especially handy when looking for stereo correspondences.

The mapping of a point to an epipolar line is given through F as

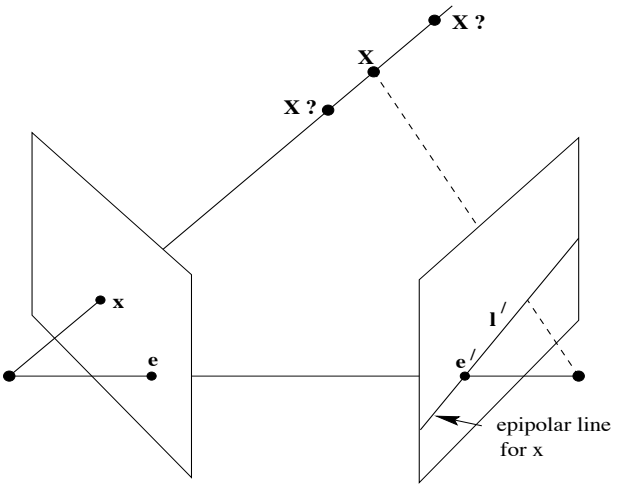


Figure 2.7: *Showing the epipolar geometry, i.e, the epipoles, the epipolar line* $\mathbf{l}'$ *and camera centres. A 3D world point maps to two image planes at point* $\mathbf{x}$ *and* $\mathbf{x}'$*. Considering* $\mathbf{x}$ *and back-projecting to the 3D point forms a ray which is imaged onto the second view as* $\mathbf{l}'$*. The projection of the world point in 3-space in the second view will be on* $\mathbf{l}'$*. The question marks represent the ill defined inverse mapping from 2-space to 3-space without triangulation.*

$$\mathbf{l}' = \mathrm{F}\mathbf{x} \tag{2.9}$$

where F is a 3×3 matrix with rank 2. Any scene viewed by two non-coincident cameras will form a unique F satisfying

$$\mathbf{x}'\mathrm{F}\mathbf{x} = 0. \tag{2.10}$$

Estimating F

Equation (2.10) defines F for all correspondences in two views of the same scene. Expanding it for a pair of correspondences $\mathbf{x}'$ and $\mathbf{x}$ with homogenous coordinates gives

$$x'xf_{11} + x'yf_{12} + x'f_{13} + y'xf_{21} + y'yf_{22} + y'f_{23} + xf_{31} + yf_{32} + f_{33} = 0 \tag{2.11}$$

where f_* are parameters of F and z' and z are equal to 1. Each correspondence therefore gives one linear equation in the parameters of F.

Equation (2.11) can be assembled in a design matrix $C\mathbf{f} = 0$ for k correspondences. A minimum of 8 such correspondences are required to solve for $\mathbf{f}$ up to scale. We can solve for $\mathbf{f}$ using SVD, the smallest singular value of C. This also applies for an overdetermined solution.

As with $\mathcal{H}$, a dominant F can be estimated using the RanSaC algorithm, getting rid of false correspondences and giving a robust estimate.

An optimised solution for F As in the case for $\mathcal{H}$, the estimate for F can also be optimised. The difference exists in the cost function over which the solution is minimised. The cost function to minimise is

$$\sum_k d\left(\mathbf{x}_k, \hat{\mathbf{x}}_k\right)^2 + d\left(\mathbf{x}_k^{'}, \hat{\mathbf{x}}_k^{'}\right)^2 \tag{2.12}$$

where d is Euclidean distance and $\hat{\mathbf{x}}_k$ and $\hat{\mathbf{x}}_k^{'}$ are estimated true correspondences that satisfy $\hat{\mathbf{x}}_k^{'}\mathrm{F}\hat{\mathbf{x}}_k = 0$. They are determined first, by triangulating the point $\hat{\mathbf{X}}$ in 3D and then using the expression $\hat{\mathbf{x}} = \mathrm{P}\hat{\mathbf{X}}$ and $\hat{\mathbf{x}}^{'} = \mathrm{P}^{'}\hat{\mathbf{X}}$ to determine the points. The cost function is minimised over parameters of P, $\mathrm{P}^{'}$ and $\mathbf{X}$ leading to an optimised F.

We define how to extract the camera matrices next and the triangulation method is left for Chapter 5. The general algorithm for estimating F is same as Algorithm 2.2, but with a different cost function.

Retrieving The Camera Matrices from F

One of the most important properties of F is that it can be used to determine camera matrices of two views up to a projective transformation. They are fundamental in the inverse mapping $\mathbb{R}^2 \rightarrow \mathbb{R}^3$ leading to 3D scene reconstruction. Here, we outline how to extract these matrices from F. Using them for 3D scene reconstruction is tackled later on in Chapter 5.

The following definition is due to [24]. It describes how to obtain camera matrices P and $\mathrm{P}^{'}$ from F. The camera matrices are a 3×4 matrix each.

Definition 2.2. The camera matrix corresponding to a fundamental matrix F may be chosen as $\mathrm{P} = [I|0]$ and $\mathrm{P}^{'} = \left[\left[\mathbf{e}^{'}\right]_{\times}\mathrm{F}|\mathbf{e}^{'}\right]$.

Here I is the identity matrix, $\mathbf{e}^{'}$ is the epipole in the second view and $\left[\mathbf{e}^{'}\right]_{\times}$ is a skew-symmetric matrix. $\mathbf{e}^{'}$ is the right null-vector of F and can be got from an SVD of F. Since $\mathbf{e}^{'}$ is a 3-vector, $\left[\mathbf{e}^{'}\right]_{\times}$ is given by,

$$[e^{'}]_{\times} = \begin{bmatrix} 0 & -e_3^{'} & e_2^{'} \\ e_3^{'} & 0 & -e_1^{'} \\ -e_2^{'} & e_1^{'} & 0 \end{bmatrix}. \tag{2.13}$$

The above definition puts the world origin at the first camera. It is worth noting that a pair of camera matrices uniquely determine F but the converse is not true. The camera matrices obtained from F are up to a projective transformation and, without further information and manipulation, do not produce a metric reconstruction.

Additional information There exists a specialisation of F known as the essential matrix E. If the camera's internal parameters K are known before hand, we can get E from F as

$$\mathrm{E} = K^{'}\mathrm{F}K. \tag{2.14}$$

Using E the camera matrices may be retrieved up to a scale factor, meaning that we can reconstruct a scene in 3D without projective ambiguity and get the relative pose of the cameras [24].

The work done in 3D reconstruction in this thesis uses F as we are assuming no prior knowledge of the camera's internal properties.

2.4 Chapter Conclusion

In this chapter, basic principles of image formation and display are introduced. The pinhole camera is described as a model of choice for image formation. We talk in fair detail about multi-view geometry, in particular point-to-point and point-to-line mapping between two views. The first is defined by the homography $\mathcal{H}$ and is used to construct image mosaics. The second is defined by the fundamental matrix F and captures the epipolar geometry describing the geometry of the scene. This transformation is used, among other things, for inverse mapping from 2-space to 3-space. We show how to extract the camera matrices from F to be used in 3D reconstruction. Finally, we provide general algorithms to estimate both $\mathcal{H}$ and F.

3
Interest Points in Images

In the previous chapter, we alluded to image features and how they are used to estimate projective transformations, i.e., the homography and the fundamental matrix. Here, we indulge further and explain how features are extracted, described and matched between corresponding views.

Image features are known by many names, e.g., control points, interest points, corners and edges. Their uses range from model estimation to classification and object recognition in images [8, 56, 57]. Because of their abundant use in image processing and machine vision, a lot of work has gone into developing reliable and efficient methods for feature extraction and matching (feature description is included in the matching phase). In this chapter, we introduce techniques that enhance these methods, specifically feature matching. An investigation into two types of feature descriptors is conducted, namely raw intensity and distribution vectors. A novel two signature descriptor that combines intensity gradients and colour is introduced showing promising results. Furthermore, an innovative feature clustering technique based on colour codes is given that reduces search space for feature matches, therefore increasing speed and reducing the rate of false matches.

The first task when using image features is to define what type of information is extracted from the images, known as the feature space. This can be raw pixel values or other distinct properties, such as edges, corners, line intersections and closed-boundary regions [8]. In our work, the feature space consists of corners and edges. As opposed to extracting salient features from images, area-based techniques also exist that match large patches of images. This technique is best suited to images

with poor image content, i.e., medical imaging [7, 56, 58]. Furthermore, area-based methods have low invariance to geometric and photometric changes. This is in contrast to feature based methods which are comparatively robust. To be discussed shortly.

A rich survey on feature detectors is given in [56]. It includes detectors based in the intensity and the frequency domain. The latter is difficult to implement and requires significant overlap between images for accurate model estimation [30, 31]. Detectors originating in the intensity domain include the Harris feature detector. It is one of the earliest and quickest detectors available. It uses local gradient information to define an edge or a corner [59]. A performance comparison on grey-level intensity based detectors is given in [60]. Here a less complex version of the Harris feature detector is proposed and several performance criteria, such as stability, localisation and complexity are introduced. Recently, scale invariant feature detectors have been proposed [57, 61–63]. These detectors cope well with changes in scale and other geometric transformations, such as affine rotations between views. Recent still, fast detection techniques have become available which use machine learning to define checking rules for the detector. These, although fast, are susceptible to image noise [64].

To deal with a high number of detected features, several techniques have been proposed to prune this number and use a reliable subset for estimation. A few techniques are discussed in [56]. In [20], a procedure in which points belonging to a convex hull of the whole set are used for matching. This reduces the number of false matches leading to an accurate estimation.

After detection of features comes feature description. This is necessary for establishing correspondence between images and accurate matching leads to effective performance of higher level tasks, e.g., model estimation. Over the years, much effort has been spent to describe local regions around a feature with enough distinctiveness and robustness for accurate feature matching and a variety of techniques exist with varying complexities. A less demanding method is to use local raw intensity values in the vicinity of the feature as a descriptor. Although, at times effective and appreciated for its simplicity, its lack of robustness makes it weak [56]. Local gradient information defined over grey level images can also be used

for simple description and matched using cross-correlation(CC). Some techniques describe features over processed images, such as moment images to deal with illumination [65].

Another effective technique, though complex to apply, is to describe features in the frequency domain. Exploiting the Fourier representation of an image, phase correlation can be applied to identify translations between two images [29]. More recently, matching at a sub-pixel level is achieved using phase correlation, as shown in [66]. Phase based local description is also studied and applied in [67] and provides invariance to changes in illumination. For a further insight into these techniques refer to [56, 68].

Yet another form of feature representation is distribution based, where local information is compressed into appropriate bins. In its most simplest form, this is a histogram of intensity values. Such methods reduce the dimensionality of the local representation and add robustness, though at the cost of descriptive power. In [69], local description is achieved using non-parametric methods which make it robust to illumination changes. Intensity relations between neighbouring pixels are encoded by binary strings and a distribution of all possible combinations is represented by histograms. Although suitable for local representation, it requires a large dimension [68]. Besides intensity values, other types of local information can also be used for local distribution. For example, gradients structured into orientation bins is applied in [57] for local representation and has proven to be very robust to scale and small geometric error. A performance evaluation of local descriptors presented in [68] reveals how this technique outperforms all the other contestants.

Feature Based Method vs. Direct Method for Estimation

Before diving into the subject of feature detection and matching, we compare the feature based estimation method to the direct area based method. This is to justify our approach for using image features.

Direct methods, seek a transformation, e.g., homography, between two images which maximises a similarity measure when one of them is warped and compared to the other.

As previously mentioned, an important motivation for using features is their invariance to geometric and photometric transformations. The cornerness measures for most feature extractors are inherently immune to changes in illumination up to a certain extent [57, 59, 61]. This is not the case for direct methods. For such, techniques steps are required to achieve photometric invariance. An example is to use a photometric model to account for changes in illumination between images [1]. This, though, requires extra parameters to be included in the registration process. Other techniques include pre-filtering the images to remove low frequency content prior to registration and/or using an invariant similarity score such as normalised cross-correlation. If these measures are inadequate, corresponding pixels in two images may exhibit large differences in the matching score even with exact geometric registration. This can lead to a biased estimate of the geometric transformation because of large residuals. Therefore, direct methods are arguably at a disadvantage here.

Another appealing aspect of using images features is the ease in which robustness to outliers can be achieved. Occlusions and specularities in a scene lead to inaccurate registration. In such cases, parts of the image will map correctly under a transformation, whilst others will not. Consequently, different features matches will be consistent with different motion models. In feature based methods, robustness to these outliers can easily be achieved using RanSaC (Chapter 2) which outputs the dominant transformation. In the case of direct methods using an outlier rejection algorithm, such as RanSaC, is costly. Other techniques based on non-convex M-estimators are available but are also computationally expensive [1, 70].

A drawback of the feature based approach is its comparatively limited applicability. Image features are suitable for a vast variety of everyday images, but for more specialist applications they can prove inadequate. For example, scenes with little edge information, such as lakes, deserts or medical imaging [7, 56, 58]. Indeed, even for such cases, there may still be an appropriate choice of feature type but

this requires significant scene understanding. Direct methods, on the other hand, are more universal because they do not distinguish between discontinuous and smoothly varying regions of the image.

3.1 Feature Detection

Our focus in this chapter is on feature matching which includes feature description and matching. For this reason we use a simple but reliable feature detector, i.e., Harris features. These features are robust to noise and are invariant to rotations [59, 60, 71].

Harris Features

This feature extractor is a gradient based detector. Changes in pixel intensities over a set region in an image greater than a threshold t define a feature. Moravec was one of the first to use such a technique [72]. Harris and Stephen [59] later improved upon it and introduced the Harris feature extractor. Their technique defines a second moment matrix A that describes local gradient distribution. The eigenvalues λ of A either define a corner, an edge or a flat region (features are taken as corners and edges). In order to avoid the computationally expensive task of determining λ every time, a cornerness measure $\mathbf{C}_m$ is introduced

$$\mathbf{C}_m = |A| + k \times tr\,(A)\,, \tag{3.1}$$

where tr is the trace and A is a second moment matrix given by

$$A = \sum_l \sum_m w\,(l,m) \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_x^2 \end{bmatrix}. \tag{3.2}$$

Here I_a is the derivative in the direction a. These derivates are usually weighted over a Gaussian window w of size $l \times m$. The value for k is chosen heuristically, a typical value of 0.04 is used and t is set for $\mathbf{C}_m$ below which all features are discarded.

Figure 3.1 gives an example of Harris features extracted from an image for different t. It is obvious that increasing the threshold results in a reduced number of features.

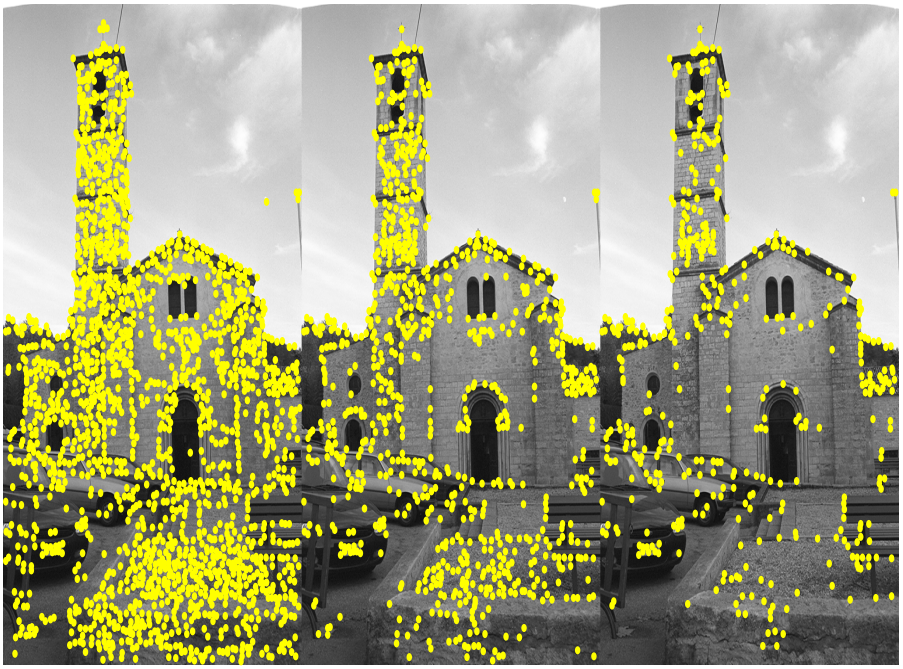


Figure 3.1: *Harris feature extracted from an image with different thresholding* t *values (values tested sare 30, 40, 50). Selecting a reasonable for the* t *is an empirical task. Non-maximum suppression is also applied with a dilation radius of 3 pixels. The image is taken from [73].*

After extracting initial features, non-maximum suppression is applied to get features whose gradients are locally maximal in a defined radius. This allows us to reduce local congestion of features and helps in managing features more efficiently in terms of space and goodness of features. Here, we employ a dilation mask of a 3 pixel radius for non-maximum suppression.

3.2 Feature Matching

Here, techniques for feature matching are presented. This includes feature description since this is a precursory step in matching. We have already mentioned that the most simplest form of feature description is the use of raw intensity values.

This, though effective and liked for its simplicity, is not robust enough to geometric and photometric changes [68].

We begin by concentrating on this simple sort of descriptor, since it does have its merits. It is simple, easy to implement and fast. This descriptor is matched using two different types of cross-correlation techniques called maximal correlation and supported correlation. Next, we introduce distribution based techniques and compare them to simple raw intensity vectors. Two types of distribution based descriptors are included. The first is a sole gradient based descriptor loaded into orientation bins. The second is a novel two signature distributor that integrates grey level gradients with colour. It combines the robustness of distribution based techniques with the descriptive power of colour extracted from a CIExy colour chart. We finish off by contributing a novel feature clustering technique that allows extra fast feature matching. This technique is based on colour coding and shows promising results.

3.2.1 Intensity Vectors and Correlation Matching

Here, we compare two cross correlation matching techniques to match image features. They are maximal cross correlation and supported cross correlation. We show how to extract intensity vectors and introduce the cross correlation score S.

Once features are extracted, an intensity vector can be obtained by placing a window of a predefined size over each pixel. The bigger the window size the more distinctive the descriptor. This, though, comes at a computational cost. These windows are compared between images to establish correspondences.

Consider Figure 3.2. Here, a feature $\mathbf{x}$ in one image is matched to a corresponding feature $\mathbf{x}'$ in another image from a list of potential matches. A window of predefined size $((2n+1)\times(2m+1))$ is placed over $\mathbf{x}$, we call this a correlation window and an intensity vector is extracted. This vector is matched to similar vectors extracted from the second image to establish correspondence. A straightforward correlation based matching algorithm will iterate through a list of potential features to find a match. A search window d_{max} can be defined over the second image to reduce the computational effort of cycling through all candidates. The size

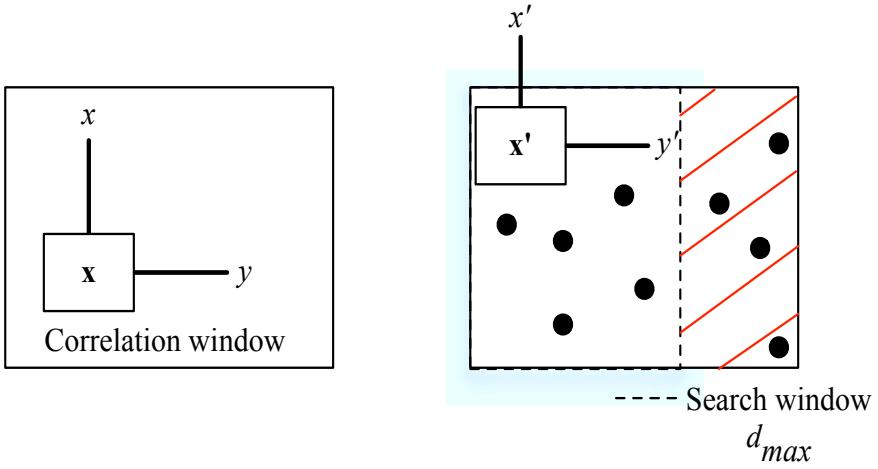


Figure 3.2: *Showing how features (black dots) are matched using simple intensity vectors. An image vector (patch) from one image is matched to a list of potential matches in another. A search window d_{max} (hashed lines) can be defined to reduce the search space. In the figure, the red cross hatched area is the blocked space for looking for feature correspondences.*

of the d_{max} reflects prior knowledge of the amount of distortion between the two images, meaning how much one image has transformed from the other.

The normalised cross-correlation score between two intensity vectors is given by

$$\mathbf{S}\left(P_{\mathbf{x}}, P_{\mathbf{x}'}\right) = \frac{\sum_{i=-n}^{n}\sum_{j=-m}^{m}\left(\left[\mathcal{I}_1\left(x+i, y+j\right)-\bar{\mathcal{I}}_1\right]\left[\mathcal{I}_2\left(x^{'}+i, y^{'}+j\right)-\bar{\mathcal{I}}_2\right]\right)}{\sqrt{\sigma^2\left(\mathcal{I}_1\right)\sigma^2\left(\mathcal{I}_2\right)}}, \quad (3.3)$$

where $\mathcal{I}$ is an image, $\bar{\mathcal{I}}_c$ $(c = 1, 2)$ is the average of the window at point (x, y) and $\sigma\left(\mathcal{I}_c\right)$ is the standard deviation. Cross-correlation score (matching score) $\mathbf{S}$ ranges from -1 for a pair of intensity vectors which are completely unalike to +1 where they are identical. Note that the above measure is symmetrical, so we can switch between the two images and the correlation score will be the same.

With the help of Equation (3.3) a matching score between each $\mathbf{x}$ in one image and all $\mathbf{x}^{'}$ in another image is obtained. This way, for each feature an array of potential matches is determined based on $\mathbf{S}$. Selecting the best correspondences from this array of potential matches is where distinction between the maximal cross-correlation and supported cross-correlation method for feature matching exists.

Maximal Cross-Correlation

From the array of potential matches for every $\mathbf{x}$ in the first image the feature yielding the highest $\mathbf{S}$ is chosen from the second image. If the match is consistent in the opposite direction then it is taken as a conclusive correspondence between two images. This means that, if a correspondence is to be established between two features they both should have the highest matching scores with each other in both directions. This is exactly what is shown in Figure 3.3. Here, a feature in the right hand image has two potential matches in the left hand image but only one match yields maximal score in both directions. It is therefore a conclusive match.

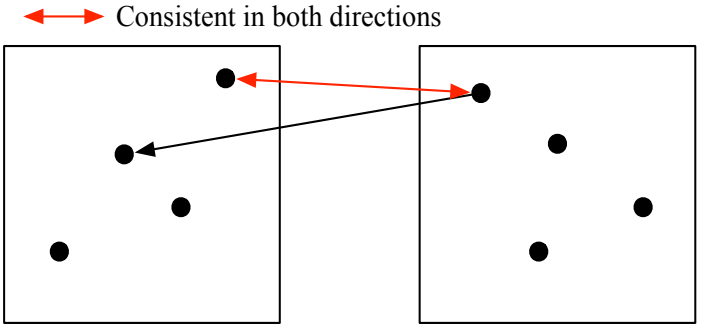


Figure 3.3: *Maximal cross-correlation technique for establishing feature correspondences. A correspondence is only conclusive if it yields maximal cross-correlation score* $\mathbf{S}$ *in both directions, as shown by the red two headed arrow. The black dots represent features.*

Supported Cross-Correlation

In contrast to the above method, the supported cross-correlation technique only considers potential matches having $\mathbf{S}$ above a certain threshold. Furthermore, to disambiguate the matches, it employs a relaxation technique [74].

After applying the correlation process previously defined, a feature $\mathbf{x}$ in the first image can be paired with a number of features in the second image called potential matches. Now if $\left(\mathbf{x}, \mathbf{x}^{'}\right)$ is a good match it is expected that the neighbouring points $\left(n_{1i} \in N\left(\mathbf{x}\right), n_{2j} \in N\left(\mathbf{x}^{'}\right)\right)$ of the two features will also produce good matches, as seen in Figure 3.4. On the other hand, if the candidate match is a bad match then neighbouring points will exhibit bad matches. Here, N is the neighbourhood of a feature.

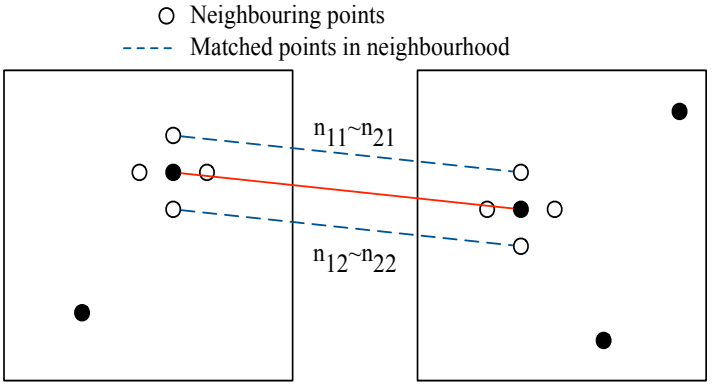


Figure 3.4: *Conclusive feature correspondences will produce good matches within the neighbourhood of the features. Black dots are features.*

The technique formally defines a measure of support for each candidate match referred to as the "strength" of the match. It is the sum of the maximum individual scores for each n_{1i} with $N\left(\mathbf{x}^{'}\right)$ given by the following expression

$$S_M\left(\mathbf{x},\mathbf{x}^{'}\right)=c_{\mathbf{x}\mathbf{x}'}\sum_{n_{1i}\in N(\mathbf{x})}\left[max_{n_{2j}\in N'(\mathbf{x}')}\frac{c_{ij}\delta\left(\mathbf{x},\mathbf{x}^{'};n_{1i},n_{2j}\right)}{1+dist\left(\mathbf{x},\mathbf{x}';n_{1i},n_{2j}\right)}\right]. \tag{3.4}$$

Here $c_{\mathbf{x}\mathbf{x}'}$ and c_{ij} are the correlation scores (Equation (3.3)) between match $\left(\mathbf{x},\mathbf{x}^{'}\right)$ and (n_{1i},n_{2j}), respectively, $dist\left(\mathbf{x},\mathbf{x}^{'};n_{1i},n_{2j}\right)$ is the average distance between each feature and its neighbour

$$dist\left(\mathbf{x},\mathbf{x}^{'};n_{1i},n_{2j}\right)=\left[d\left(\mathbf{x},n_{1i}\right)+d\left(\mathbf{x}^{'},n_{2j}\right)\right]/2$$

and

$$\delta\left(\mathbf{x},\mathbf{x}^{'};n_{1i},n_{2j}\right)=\begin{cases}e^{\frac{-r}{\epsilon_r}} & if\ (n_{1i},n_{2j})\ is\ a\ candidate\ match\ and\ r<\epsilon_r\\ 0 & otherwise\end{cases}$$

where r is the relative difference given by

$$r=\frac{\left|d\left(\mathbf{x},n_{1i}\right)+d\left(\mathbf{x}^{'},n_{2j}\right)\right|}{dist\left(\mathbf{x},\mathbf{x}';n_{1i},n_{2j}\right)}$$

and ϵ_r is the relative difference threshold. Remarking on S_M, the neighbourhood matches whose relative positions are similar to the candidate correspondence $\left(\mathbf{x}, \mathbf{x}^{'}\right)$ are considered. Their contributions are a multiple of the exponential of the negative relative error r and therefore when they are similar in position to the candidate match their contribution is large. In the instance that a neighbouring point n_{1i} has a several matches in $N\left(\mathbf{x}^{'}\right)$ the max operator selects the one with the smallest distance difference. Lastly, the contribution of each neighbouring point is weighted by its distance to the candidate match. This allows points closer to the candidate match to have more contribution in S_M.

The measure for S_M is not symmetric, it will not be the same if the roles of the two images are reversed. This happens when several points in $N\left(\mathbf{x}\right)$ score maximal values with a point in $N\left(\mathbf{x}^{'}\right)$. To overcome this problem, before computing the summation, if more than one neighbouring point scores a maximal value with the same point in $N\left(\mathbf{x}^{'}\right)$, the one with the highest value is taken. This ensures symmetry, for if the roles of the two images are reversed the same pairings will be counted.

In order to finalise the matches "a winner take all" strategy is adopted in which a candidate match $\left(\mathbf{x}, \mathbf{x}^{'}\right)$ is taken as a conclusive match if either $\mathbf{x}$ or $\mathbf{x}^{'}$ has no higher S_M with any other possible matches. This is illustrated clearly in Figure 3.5. Consider the first row, here $\mathbf{x}_1$ has two candidate matches $\left(\mathbf{x}_1, \mathbf{x}_1^{'}\right)$ and $\left(\mathbf{x}_1, \mathbf{x}_3^{'}\right)$ and the one chosen is one with the highest S_M, i.e., $\left(\mathbf{x}_1, \mathbf{x}_1^{'}\right)$.

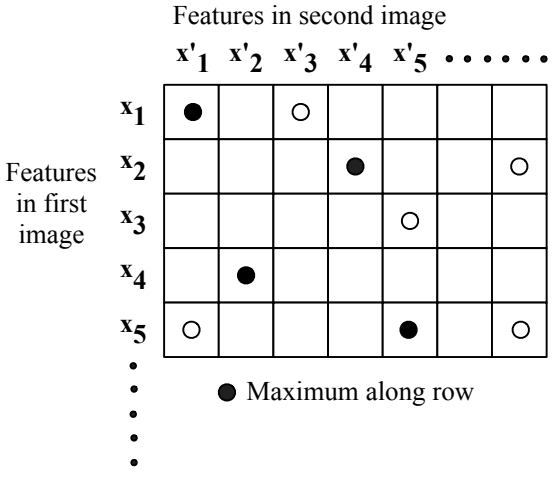


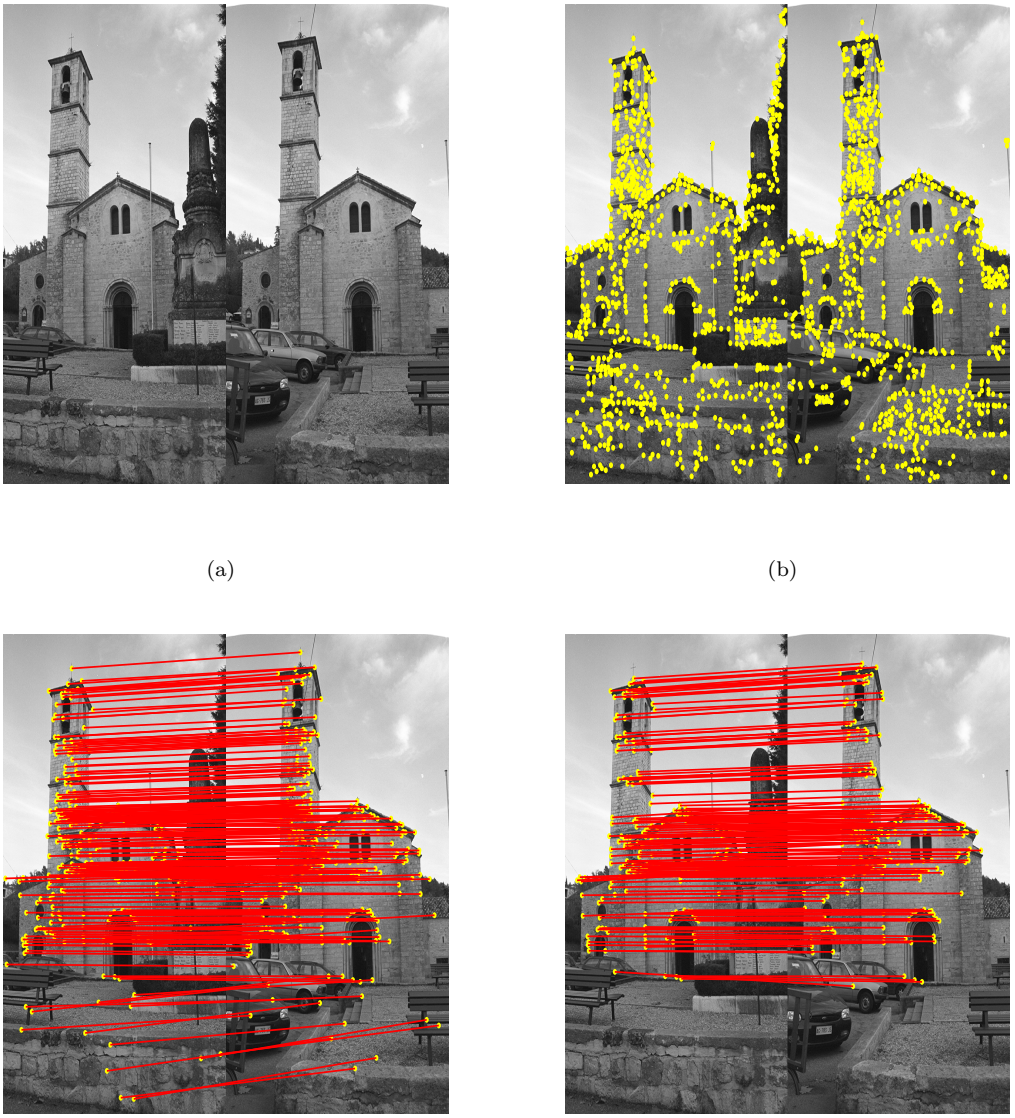
Figure 3.5: *Illustration of the "winner take all" approach.*

Experimental analysis Now we include a few examples showing how these two matching techniques fair. In order to compare matching techniques, we introduce a measure for comparison. This measure is the precision, which is the ratio of correct matches to all matches [68]. If a technique has matched all features correctly, it will have precision of 1. Correct matches are identified via RanSaC (Chapter 2).

Figure 3.6(a) gives a couple of overlapping images. Individual features from these images are extracted and matched using the two cross-correlation techniques just described. Figure 3.6(b) shows the features extracted using the Harris feature detector. Figure 3.6(c) and 3.6(d) shows feature correspondences established using maximal cross-correlation and supported cross-correlation, respectively. The maximal cross-correlation technique yields a precision value of 0.59 and the supported cross-correlation technique gives a value of 0.73.

Figure 3.7 is an example of a couple of overlapping images with changing brightness. For this example, the maximal cross-correlation technique yields a precision value of 0.94, whereas the supported cross-correlation technique yields a value of 0.92. Both techniques have over 90% precision for this image set.

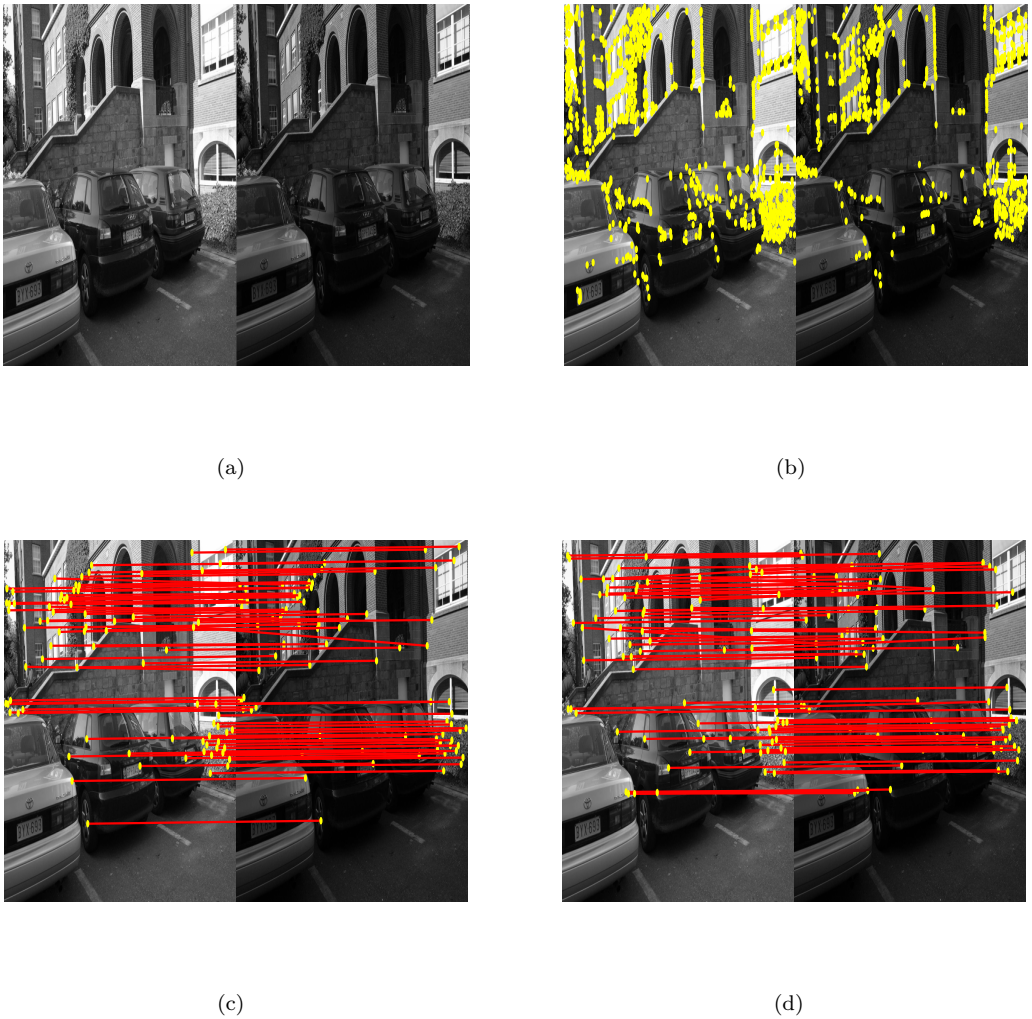
Figure 3.8 is an example of a couple overlapping images where the camera has moved significantly from one image to another. For this image set, the maximal cross-correlation technique yields a precision value of 0.39, whereas the supported cross-correlation technique yields a precision value of 0.64.



(a) (b)

(c) (d)

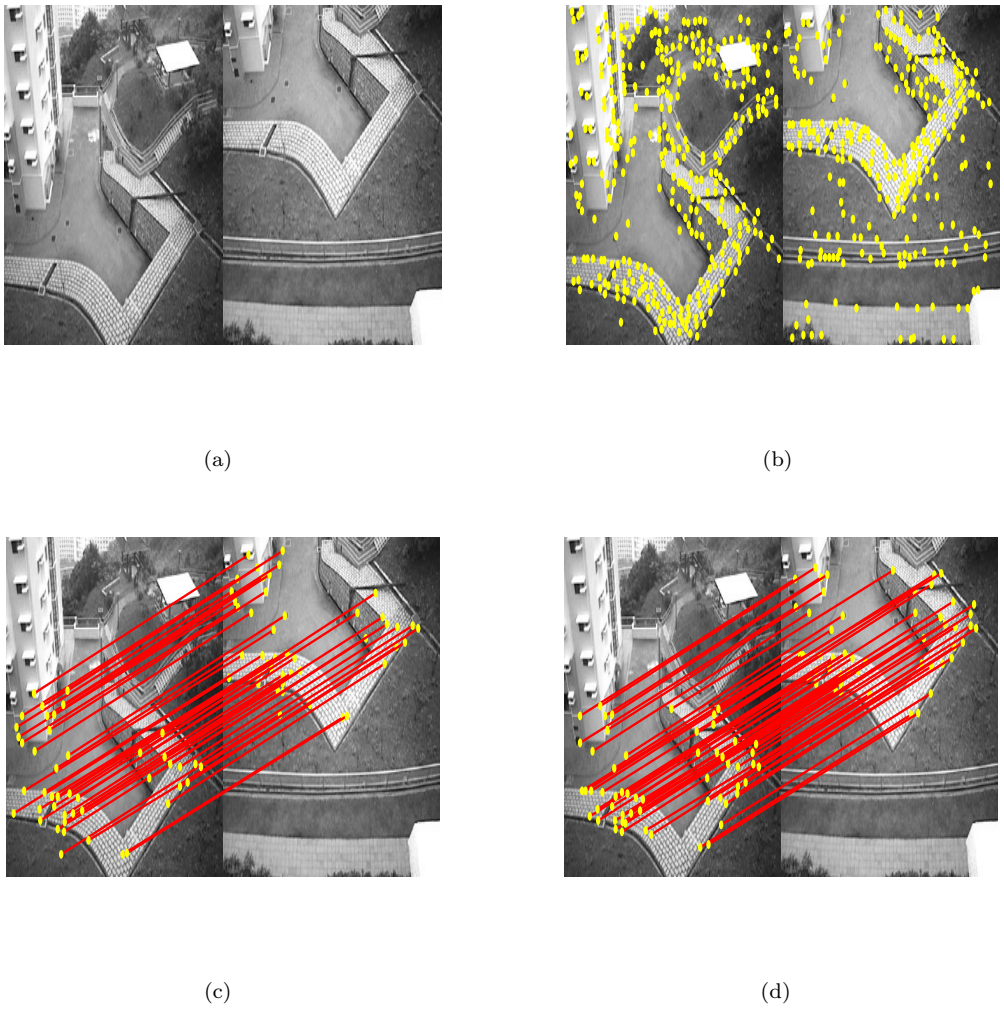
Figure 3.6: *Comparing matching characteristics of maximal cross-correlation to supported cross-correlation. (a) A couple of overlapping images. (b) Harris features with a detection threshold of 30. (c) Feature correspondences from maximal cross-correlation. Note the false correspondences at the bottom of the figure. These are present even after outlier removal with RanSaC. (d) Feature correspondences from supported cross-correlation. The number of false correspondences after RanSaC are greatly reduced.*



(a) (b)

(c) (d)

Figure 3.7: *Comparing the matching characteristics of maximal cross-correlation to supported cross-correlation for images with varying brightness. (a) A couple of overlapping images. (b) Haris features with a detection threshold of 30. (c) Feature correspondences from maximal cross-correlation. A reduced number of correspondences is shown for clarity. (d) Feature correspondences established from cross-correlation. A reduced number of correspondences is shown for clarity.*



(a) (b)

(c) (d)

Figure 3.8: *Comparing the matching characteristics of maximal cross-correlation to supported cross-correlation for images with varying viewpoints. (a) A couple of overlapping images. (b) Harris features with a detection threshold of 30. (c) Feature correspondences from maximal cross-correlation. (d) Feature correspondences from supported cross-correlation.*

3.2.2 Distribution Based Descriptors

Here, we introduce distribution based local feature descriptors. For such type of descriptors, local information is compressed into appropriately defined bins. As mentioned in the introduction of this chapter, these types of methods reduce the dimensionality of the representation and add robustness but at the cost of descriptive power.

In this work we present two distribution based descriptors. The first, is a single signature descriptor using only intensity gradients. The second, is a novel two signature descriptor that uses colour in combination with intensity gradients to increase matching performance.

Below we describe each descriptor in detail, including experimental evaluation.

Single Signature Descriptor - 1SD

We dub this descriptor "1SD" for single signature descriptor. It has been adapted from [57] to work with a fast feature extractor such as the Harris feature detector. In its original form as the SIFT, this descriptor is adapted to multi-scale description, which is an intensive procedure since scale-space operations are required. In our method, fast and robust feature matching is done thanks to Harris features coupled with robust distribution based SIFT like description.

Consider a feature $\mathbf{x}$. A window of pre-defined size $((2n+1) \times (2m+1))$ is placed over it and local intensity information, i.e., gradients in x and y are obtained. This information is already available from the feature extraction step performed for Harris features. From this local gradient information, the magnitude and orientation are determined using

$$\begin{aligned} \mathbf{x}_{mag} &= \sqrt{I_x^2 + I_y^2} \\ \mathbf{x}_{ori} &= \text{atan2}\left(I_x, I_y\right) \end{aligned} \tag{3.5}$$

where atan2 is the arctangent function.

After these quantities are determined, they are used in combination to describe $\mathbf{x}$. The local window defined over $\mathbf{x}$ is sub-divided into v sub-regions. Gradient values of each sub-region are sorted into an 8-bin $\left(\frac{360^\circ}{8}\right)$ histogram depending on the gradient direction. The magnitude is weighted by a Gaussian window centred on $\mathbf{x}$. This is shown in Figure 3.9. When all sub-regions are sorted into their corresponding orientation bins, each bin from each sub-region is concatenated to form a vector, a distribution descriptor.

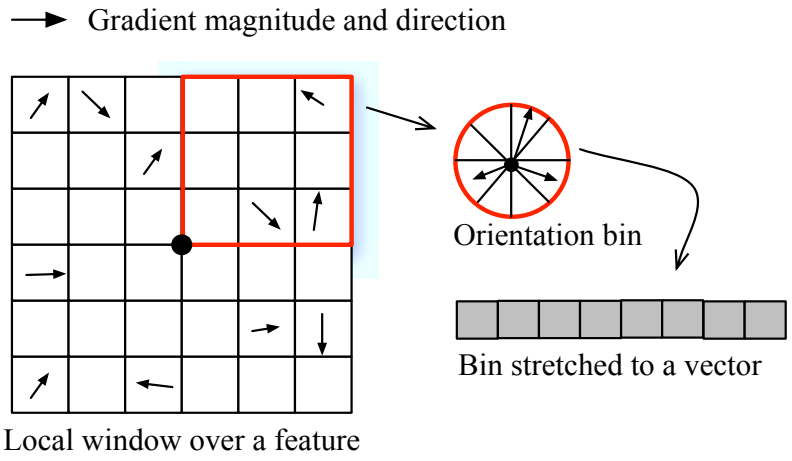


Figure 3.9: *A $(2n+1)\times(2n+1)$ window is centred on a feature* $\mathbf{x}$ *from which local gradient orientations (arrow heads) are sorted into a 8-bin orientation histogram for each sub-region of the window equivalent to Gaussian weighted magnitudes. The descriptor 1SD is constructed by concatenating bins and forming a vector.*

Following is a general algorithm for constructing 1SD for a feature.

Algorithm 3.1: The general algorithm for constructing 1SD

objective: Construct robust one signature descriptor 1SD;

Input: A feature $\mathbf{x}$ and image;

algorithm:

Centre a $(2n+1)\times(2n+1)$ window w on feature $\mathbf{x}$;

Evaluate the gradient magnitudes and orientations within w;

Further sub-divide w into $(2n+1)$ sub-regions of equal size;

for *each sub-region* **do**

  Fill a 8-bin histogram where bin one is from 0-44° the second from 45-89° degrees, and so on. The amount added to the bins is equivalent to the Gaussian weighted magnitude of each gradient.

Concatenate each bin to form a vector of histograms, called 1SD.

Two Signature Descriptor - 2SD

With feature description, we have to compromise between robustness and distinctiveness [75]. Distribution based descriptors group similar data together reducing dimensionality of the original data and add robustness to changes like brightness. This, though, comes at the cost of distinctive power. Yet relying on descriptive

power alone can lead to diminished performance, especially when scene illumination is varied. Furthermore, high descriptive power implies high dimensionality of the descriptor, i.e., large patches over features.

Motivated by this, we present a novel two signature descriptor that combines robustness of a histogram with the descriptive power of colour, the latter somewhat ignored in the literature within this context. We dub this descriptor "2SD". The technique is motivated by the CSHOT (Combined Signature of Histogram and OrienTations) descriptor given in [76] but differs in the way colour is used. As CSHOT uses local RGB values to supplement the histogram descriptor, its length is increased significantly and is sensitive to geometric and illumination changes.

The 2SD descriptor supplements gradient histogram (similar to 1SD) with a colour signature. The descriptive power of colour is beyond doubt, humans use it to identify and relate identical objects with great success. This ability has been somewhat ignored by the research community, and when considered, only raw RGB values or its variants, i.e,. CIELab or HSY are used for description [76, 77]. These characteristics, although powerful when used in abundance, are sensitive to noise. Furthermore, a potentially more powerful ability is to match colour with colour, e.g., red to red, green to green. This ability is introduced into 2SD using a CIExy chromaticity chart.

Use of colour spaces is studied in [78] as a solution to image segmentation and uses K-Means clustering for this. A comparison of local grey-level descriptors to colour invariant descriptors is given in [79]. Here local descriptors constructed using photometric invariants are tested against local grey-level invariants, i.e., gradients and give decent performance, though are more demanding computationally.

First, we introduce the CIExy chromaticity colour space. This space defines a pixel's chromaticity regardless of illumination (by chromaticity we mean quality of colour). Figure 3.10 gives the CIExy colour chart. The outline defines the spectral locus and the RGB gamut is given by the triangle with the three primary colours at the vertices. All 16 million colour combinations visible to the human eye are contained within it [54].

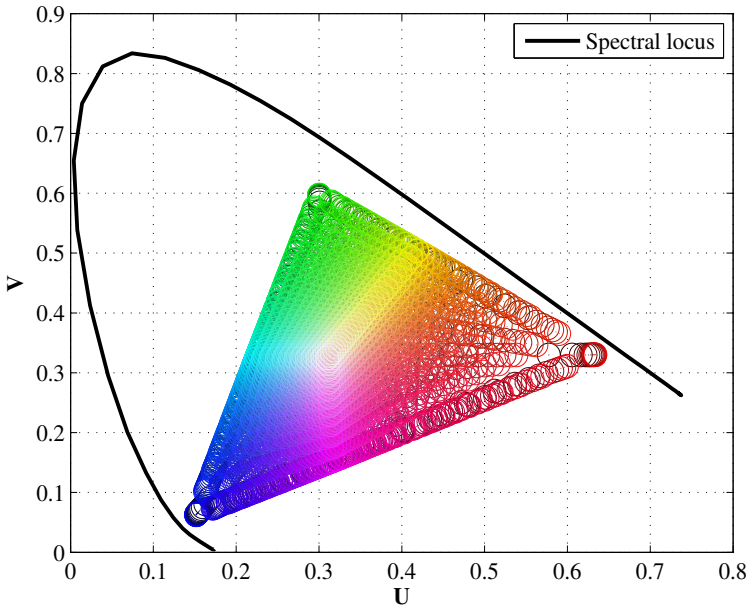


Figure 3.10: *CIExy colour space. The solid outline is the spectral locus. All 16 million colour combinations visible to the human eye are contained within it.*

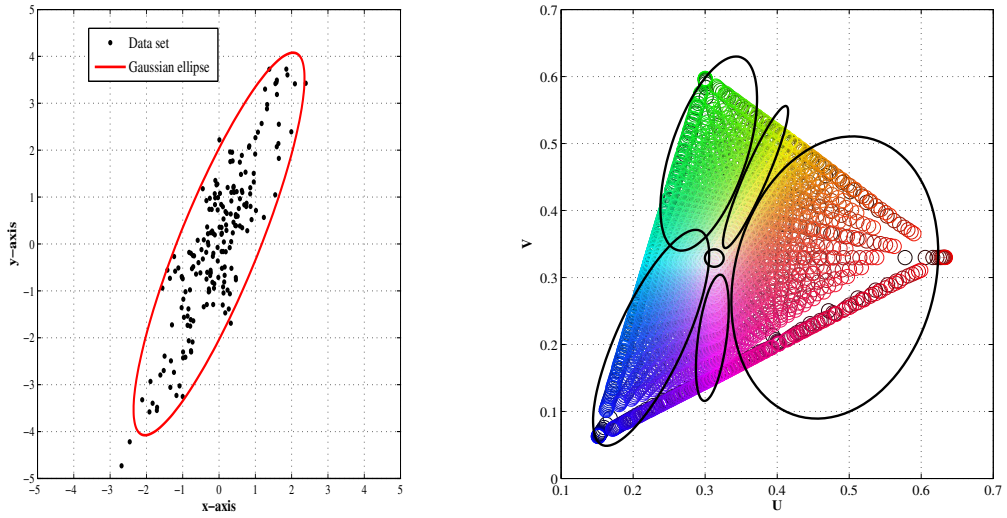
A pixel from an RGB image is transformed to this space via a transformation followed by a normalisation. The transformation is given by Equation (3.6)

$$\begin{bmatrix} L \\ M \\ N \end{bmatrix} = \mathbf{Q} \begin{bmatrix} R \\ G \\ B \end{bmatrix}, \tag{3.6}$$

where R, G and B are the pixel's RGB values, L,M and N are known as the tristimulus values analogous to cones in a human's visual system and $\mathbf{Q}$ is a known transformation given by

$$\mathbf{Q} = \begin{bmatrix} 0.4124 & 0.3576 & 0.1805 \\ 0.2126 & 0.7152 & 0.0722 \\ 0.0193 & 0.1192 & 0.9505 \end{bmatrix}. \tag{3.7}$$

The normalisation is given by Equation (3.8). The normalised coordinates U and V are two coordinates needed to localise a point on the chart.



(a) (b)

Figure 3.11: *(a) 2D Gaussian cluster for a set of random data. (b) CIExy space clustered using Gaussian models. Six clusters are formed.*

$$\begin{aligned} \mathrm{U} &= L/\left(L+M+N\right) \\ \mathrm{V} &= M/\left(L+M+N\right). \end{aligned} \tag{3.8}$$

Observing the colour space in Figure 3.10, it can be divided into several clusters based on colour. In this way, each pixel converted to the CIExy colour space can be colour coded and grouped together depending on the number of divisions of the space regardless of shade. Here, we divide the space into six clusters based on colour, i.e., red, yellow, green, blue, violet and grey (or white).

Assigning pixels to the appropriate cluster is an issue. Due to the asymmetrical nature of the partitions (see Figure 3.10), clustering based on simple Euclidean distances is insufficient. We, therefore, model our clusters using 2D Gaussian distributions. Figure 3.11(a) gives an example 2D Gaussian distribution for a random 2D data set. This is a specific model used to define the distribution of a data set according to its mean and covariance.

Using 2D Gaussian distributions the CIExy is clustered, as shown in Figure 3.11(b). The six clusters are clearly visible. In order to assign pixels to correct bin, the

probability of it belonging to each cluster is determined using Equation (3.9). The pixel is assigned to the cluster with the highest probability. This way, regardless of the shade of the colour, it will be sorted into the correct bin with a high probability. In Equation (3.9) μ is the cluster mean, ϕ is the covariance and $c\,(U, V)$ is the data point on the colour space. Note that the Gaussian models are tuned manually. Table 3.1 gives the model parameters.

$$\rho = \frac{1}{2\pi|\phi|^{1/2}} exp\left(-1/2\,(c-\mu)^T\,|\phi|^{-1}\,(c-\mu)\right). \tag{3.9}$$

Cluster	μ	ϕ
Red	$[0.48, 0.3]$	$\begin{bmatrix} 0.0200 & 0.0030 \\ 0.0001 & 0.0450 \end{bmatrix}$
Yellow	$[0.37, 0.45]$	$\begin{bmatrix} 0.0003 & 0.0050 \\ 0 & 0.0130 \end{bmatrix}$
Green	$[0.305, 0.485]$	$\begin{bmatrix} 0.045 & 0.0055 \\ 0.0055 & 0.0210 \end{bmatrix}$
Blue	$[0.225, 0.21]$	$\begin{bmatrix} 0.0065 & 0.0100 \\ 0.0100 & 0.0260 \end{bmatrix}$
Violet	$[0.31, 0.21]$	$\begin{bmatrix} 0.0004 & 0.0010 \\ 0 & 0.0090 \end{bmatrix}$
Grey	$[0.3127, 0.329]$	$\begin{bmatrix} 0.0017 & 0 \\ 0 & 0.0017 \end{bmatrix}$

Table 3.1: *Manually tuned modelling parameters for Gaussian clusters used in 2SD. Refer to Figure 3.11(b).*

Now that a way to colour code the pixels is available, the second signature can be defined. A window of size 9×9 is centred on a feature and each pixel within is colour coded as just described. Based on these codes, each is summed into one of the 6 appropriate colour bins weighted by a Gaussian window centred on the feature. This emphasises more the central pixels. In such a way, a 6 bin histogram is made that can be augmented to the gradient histogram described earlier and a vector of two distinct signatures is obtained, Figure 3.12. This descriptor is a

134 element descriptor, 128 for the gradient distribution histogram and 6 for the colour codes.

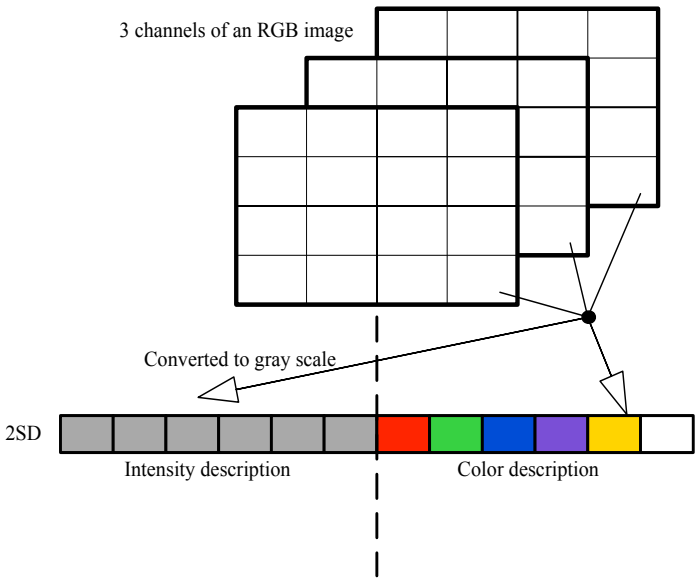


Figure 3.12: *The two signature descriptor (2SD). It is a 134 element vector, 128 for the gradient distribution histogram and 6 for the colour.*

The computational complexity of adding this second signature to 1SD is limited since the Gaussian model covariances and means are predefined.

To show how well our colour coding technique copes with illumination change, we include Figure 3.13. An image is illuminated and also dulled down. Then colour coding is applied to all pixels of the image and the image is segmented. It can be seen that the coding technique performs well even under varying illumination. Majority of the pixels share the same code across the images. This proves, visuallay, that the 2SD descriptor is robust as well as adding extra distinctive power.

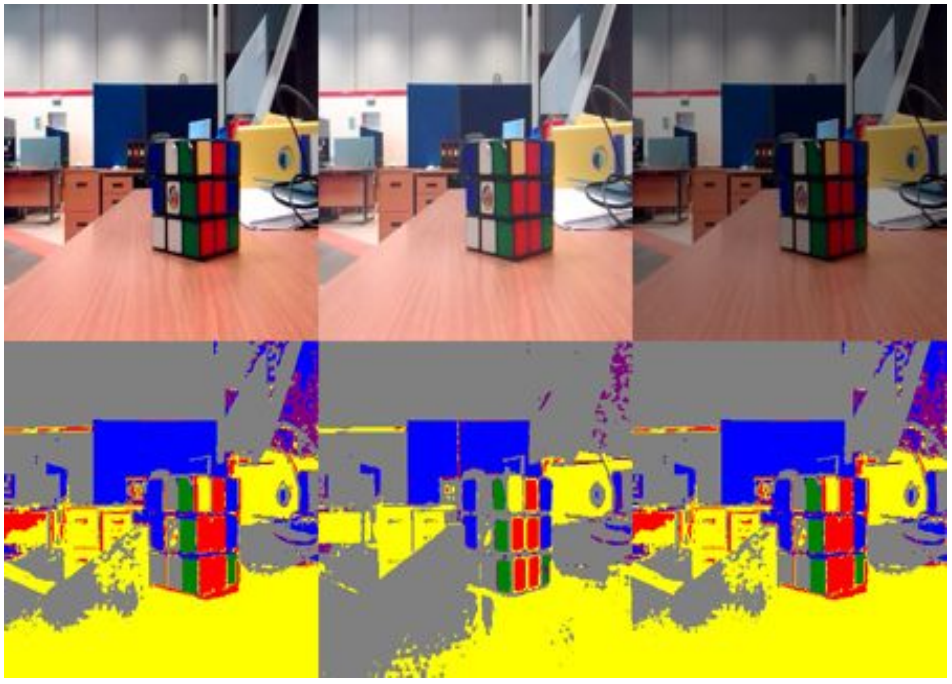


Figure 3.13: *Image segmented using CIExy colour coding for varying illumination. Majority of the pixels are assigned same colour codes.*

Following is a general algorithm for constructing 2SD for a feature.

Algorithm 3.2: The general algorithm for constructing 2SD

objective: Construct robust one signature descriptor 2SD;

Input:A feature **x** and image;

algorithm:

Follow steps to create 1SD;

Centre a $(2n+1) \times (2n+1)$ w on the feature **x** and RGB values;

for *each pixel in window* **do**

Transform RGB values into CIExy points using

$$\mathbf{Q} = \begin{bmatrix} 0.4124 & 0.3576 & 0.1805 \\ 0.2126 & 0.7152 & 0.0722 \\ 0.0193 & 0.1192 & 0.9505 \end{bmatrix}$$

and

$$\mathrm{U} = L/\left(L+M+N\right)$$
$$\mathrm{V} = M/\left(L+M+N\right);$$

Evaluate pdfs for each pixel in w for belonging to what cluster and the maximum is assigned ;

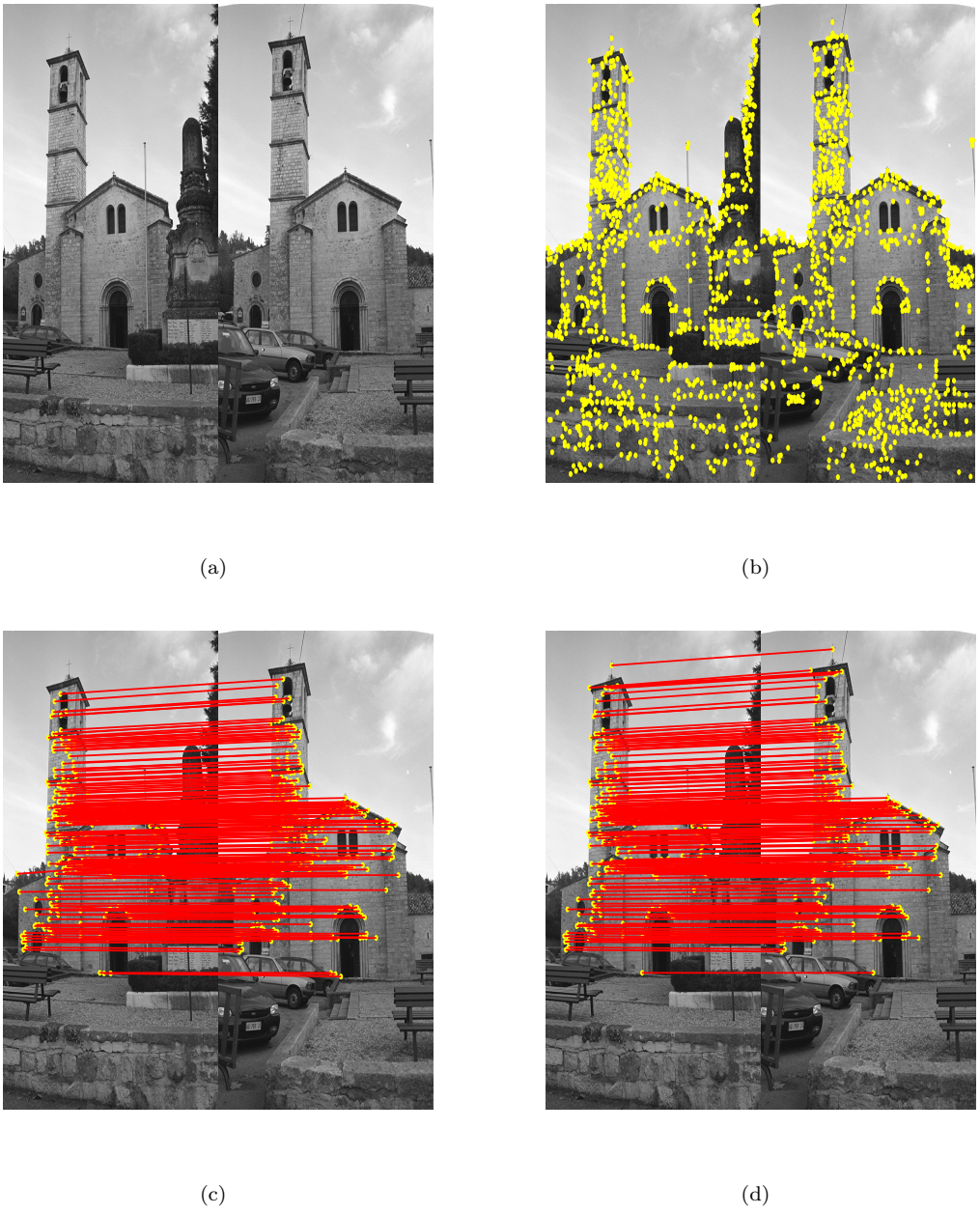
A value corresponding to a Gaussian window centred on **x** is added to assigned cluster. Do this for all pixels;

The 6 bin histogram is added at the end of 1SD;

Experimental analysis Now we include experimental results comparing performance of 1SD to 2SD. We use the Euclidean distance as a measure of similarity between descriptors. Please note that the images tested are colour images but are shown in grey scale to emphasise the features and feature matches.

The same example images used to compare cross-correlation techniques in Section 3.2.1 are used here. This is done so that later on all matching techniques can be fairly compared.

Figure 3.14 gives a set of overlapping images matched using 1SD and 2SD. This is the same example set used in Figure 3.6. Again Harris features are used. We match features between two images using 1SD and 2SD descriptors. Figure 3.14(c)



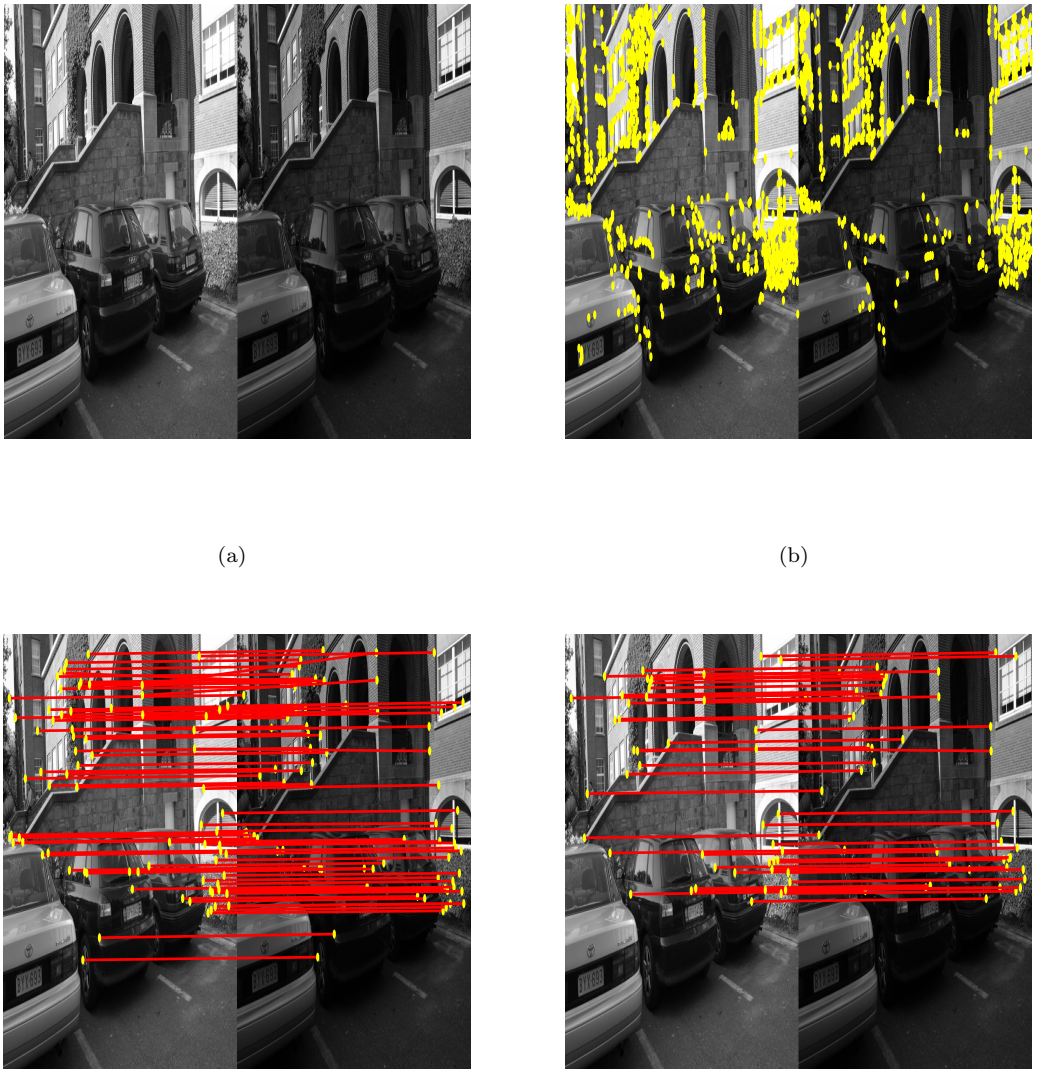
(a) (b)

(c) (d)

Figure 3.14: *Example comparing the matching characteristics of 1SD to 2SD. (a) A couple of overlapping images. (b) Harris features with a detection threshold of 30. (c) Feature correspondences from 1SD. (d) Feature correspondences from 2SD.*

and 3.14(d) show matches for each technique, respectively. As expected, both techniques yield high precision values, more than the two cross-correlation techniques [68]. Our 2SD descriptor outperforms the standalone 1SD descriptor by more than 10%. Values for precision are included in Table 3.2.

Figure 3.15 gives a set of overlapping images with varying brightness for which



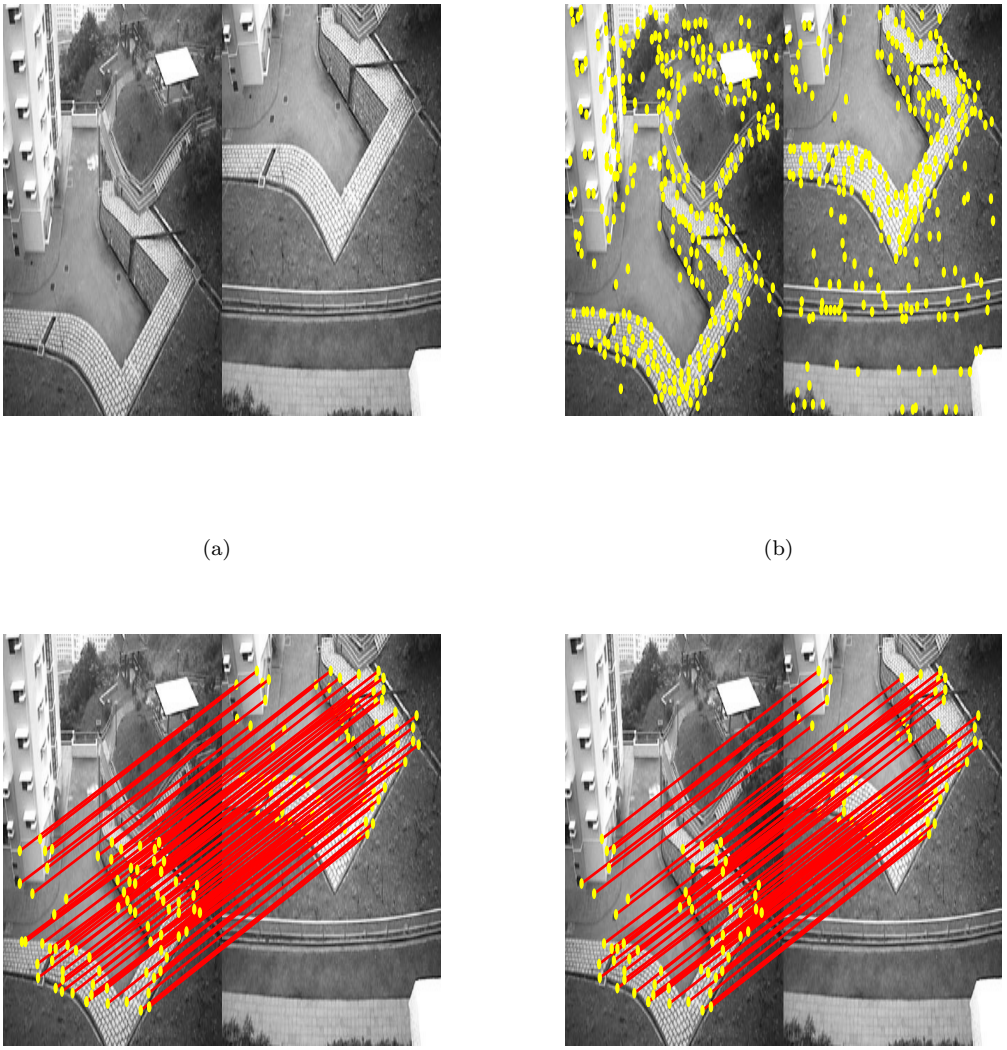
(a) (b)

(c) (d)

Figure 3.15: *Example comparing the matching characteristics of 1SD to 2SD for images with varying brightness. (a) A couple of overlapping images. (b) Harris features with a detection threshold of 30. (c) Feature correspondences from 1SD. (d) Feature correspondences from 2SD.*

feature are matched using 1SD and 2SD. This is the same example set used in Figure 3.7. For this example, both techniques yield high precision values. However, 2SD performs slightly better, by a margin of almost 1%.

Figure 3.16 gives another set of overlapping images taken from a camera separated by a distance for which features are matched using 1SD and 2SD. This is the same



(a) (b)

(c) (d)

Figure 3.16: *Example comparing 1SD to 2SD for outdoor images. (a) A couple of overlapping images. (b) Harris features with a detection threshold of 30. (c) Feature correspondences from 1SD. (d) Feature correspondences from 2SD.*

image set used in Figure 3.8. For this example too, 2SD outperforms 1SD by a percentage of more than 6%.

3.2.3 Discussion

Here, we discuss the four feature matching techniques previously tested. Just as a reminder, these include intensity vectors matched using maximal cross-correlation, intensity vectors matched using supported cross-correlation, 1SD matched using $||L||_2$ and 2SD matched using $||L||_2$. We have tested these techniques on three image sets each containing two images each. These include images taken outdoor in real environments, with a moving camera and changing illumination. Below we include a table giving precision values of each technique for all tested images, Table 3.2.

Methodology	Image set no.1	Image set no. 2	Image set no. 3
Maximal CC	0.59	0.94	0.39
Supported CC	0.65	0.92	0.64
1SD	0.74	0.94	0.66
2SD	0.82	0.95	0.70

Table 3.2: Precision values for four feature matching techniques. They include maximal CC, supported CC, 1SD and 2SD.

From studying Table 3.2, it can be seen that for all images sets the two signature descriptor (2SD) performs better than all other techniques. For image set no. 1, 2SD is on average 25% more precise. For the second image set, where there is a change in brightness, all techniques fair well, yielding a precision value of over 90%. The 2SD is better still. This implies that all techniques tested perform well with varying illumination, with 2SD performing the best. Also, this justifies that our colour coding technique using CIExy colour space is capable of coping with changes in illumination. For image set no. 3, again 2SD comes out on top by almost 30%, followed by 1SD in second place.

The results therefore show that, firstly, distribution based techniques perform better compared to raw intensity vectors, and secondly, the novel two signature descriptor adds to the capability of the intensity histogram. This is because an extra layer of distinction is added to the descriptor.

Figure 3.17: *Images are taken from the Oxford data set [73]. For each data set images are matched row by row, e.g., for the first data set (first two columns) images of the first row are matched, i.e., 1-2 then the next row 3-4.*

Additional Examples

For completeness, we include some additional examples comparing 1SD to 2SD for feature matching. Figure 3.17 gives the images on which the comparison is made. These include images taken with a moving camera, changing illumination and blur. Table 3.3 includes the precision values from the two techniques. Values in boldface are for 2SD. The results reinforce the conclusion that our 2SD descriptor outperforms all competitors for different viewing conditions.

	Test set 1	Test set 2	Test set 3
1-2	**0.92**/0.84	**0.82**/0.79	**0.51**/0.49
3-4	**0.96**/0.93	**0.87**/0.66	**0.69**/0.66

Table 3.3: *Precision values comparing 1SD to 2SD for image sets given in Figure 3.17. Values for 2SD are given in boldface.*

3.3 Matching via Reduced Search Region

In this section, we introduce a novel colour based technique to reduce the search region for feature matching. In general, when a feature $\mathbf{x}$ is to be matched to another feature $\mathbf{x}^{'}$ from a list of potential matches it is compared to each and every feature. This is time consuming and inefficient, especially when dealing with a large number of images.

Our technique is motivated by work done in [34]. Here, the bag of words algorithm used in image recognition is applied to identify regions that have been processed earlier to aid loop closure. Although using clustering for recognition works well, using it for feature matching can be ineffective unless we have a lot of similar features in an image or have a database of information to search from [80]. Every feature can be unique and can lead to a number of cluster centres. The algorithm is also applied in [81] to the problem of SLAM and loop closures. Good results are shown in real time.

We propose a colour coding based scheme that groups same colour features together. Using the CIExy colour coding technique developed previously, we can colour code each feature. This coding technique can be used to reduce the search space for features as only candidates with similar colours are considered for matching, shown in Figure 3.18. This greatly reduces the time required to establish correspondences. An added bonus of using such a technique is that it decreases the potential for false matches.

This technique differs from the bag of words algorithm in that we have a predefined cluster number from which feature matches are sought, which makes it faster. Indeed, our technique can be used as in [81] for applications in SLAM and loop closing.

Experimental analysis First, we show how using our colour clustering technique the rate of false matches is reduced. The benefits in terms of time reduction is seen when features from a number of images are matched together, as will be shown. For just two images, the clustering technique takes more time on account

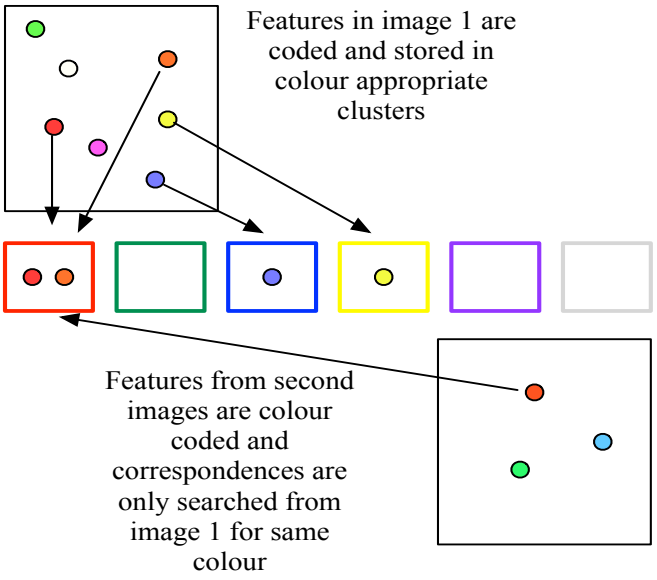


Figure 3.18: *Features are colour coded and clustered into appropriate bins. Then, features with similar colours can be matched increasing speed and reliability.*

of the extra step of checking the colour of the features. Please note that colour images are tested and are only shown in grey scale to emphasise features and feature matches.

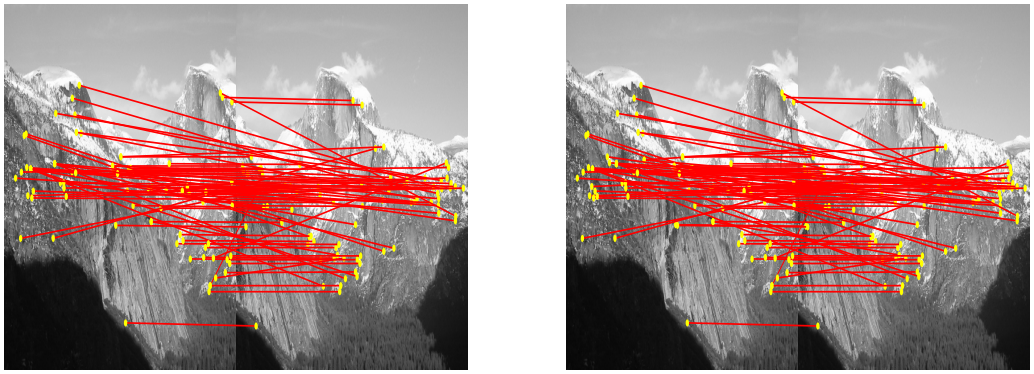
Figure 3.19 gives an example of a couple of overlapping images where the features are matched using reduced search region (RSR) and open search region (OSR), respectively. In terms of precision, RSR yields a 8% increase compared to OSR. Precision values are given in Table 3.4.

Figure 3.20 is another example comparing RSR and OSR. The RSR technique yields a 6% increase in precision as compared to OSR. Precision values are given in Table 3.4.

	Test set 1	Test set 2
RSR	0.93	0.46
OSR	0.86	0.43

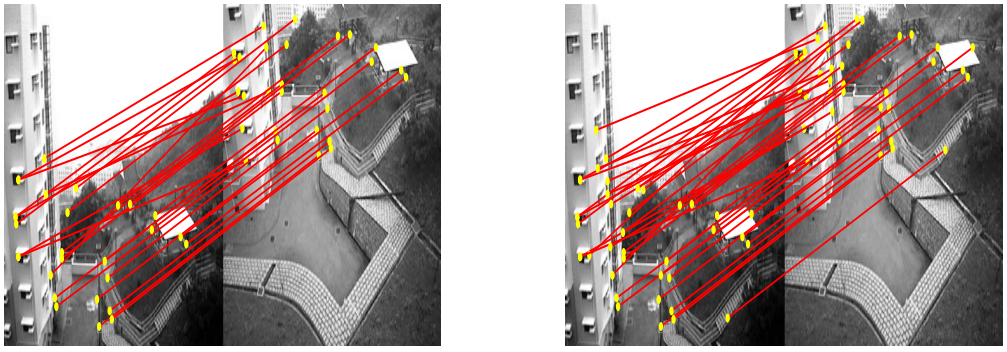
Table 3.4: *Matching precision from applying RSR and OSR to two image sets. Test set 1 is given in Figure 3.19. Test set 2 is given in Figure 3.20.*

We now highlight the most important contribution of RSR matching. This is the saving in terms of matching time when features from more than 2 images are matched together. As previously mentioned, features are colour coded and



(a) (b)

Figure 3.19: *Test set 1. Comparison showing matched features with RSR and OSR. (a) A set of overlapping images matched with RSR. (b) A set of overlapping images matched with OSR.*



(a) (b)

Figure 3.20: *Test set 2. Another comparison showing matched features with RSR and OSR. (a) A set of overlapping images matched with RSR. (b) A set of overlapping images matched with OSR.*

matched with similarly coded features from other images. Features are extracted, coded and clustered together depending on these codes and new coded features from an incoming image are only matched to same colour features from the cluster. These are then added to the cluster along with an image tag, specifying which image they belong to.

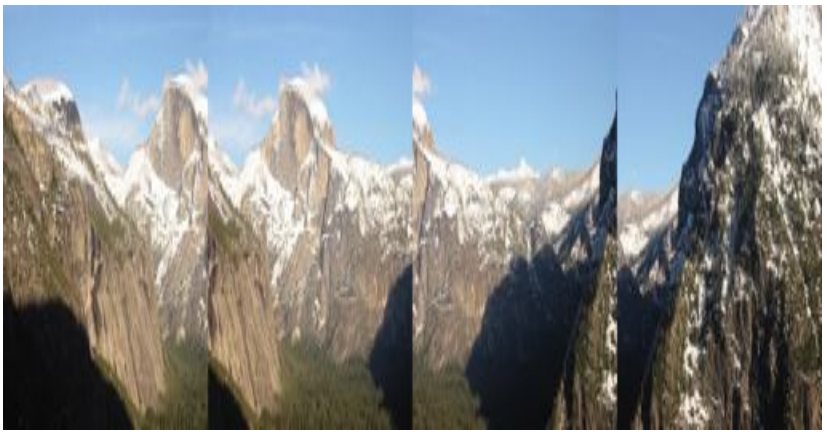
Figure 3.21 shows a set of four images from which an image mosaic is constructed. We include the image mosaic for completeness, Figure 3.21(b). As a descriptor we use 1SD, though it can work equally well with 2SD. It is to be noted that we are comparing matching times affected by reducing the search region for matching so the type of descriptor used is not impotant. Indeed, using 1SD or 2SD helps in clustering in that they are stored in vector form before the matching process. Intensity vectors are extracted within the matching process and deleted as soon as they are not required.

Figure 3.22 gives a bar chart giving the time in seconds taken by RSR and OSR to match features from all four images. The time for each is broken down into how long it takes for an image to be matched to the base mosaic. For initialisation purposes, the first image is the base and the second image is matched to it. As expected, as more images are added the time to match increases since the number of features in the base mosaic increase. The RSR technique, however, copes very well with this increase. It takes almost 5 times less to match the features.

3.4 Chapter Conclusion

In this chapter, we have introduced image features (interest points). The Harris feature detector is chosen as the detector of choice because of its speed and reliability: Harris features are stable in the presence of noise, invariant to rotation and are easy to implement.

Two different types of feature descriptors are investigated, raw intensity descriptors and distribution based descriptors. Of these, the distribution based descriptors perform the best. It is more robust to changes in noise, though they lose distinctiveness. To deal with this issue, a novel two signature descriptor, dubbed 2SD, is



(a)

(b)

Figure 3.21: *(a) A set of overlapping images, 4 images taken from a moving camera. (b) Mosaic built from four images.*

introduced. It combines the robustness of histograms with the distinctive power of colour. The gradient histogram is augmented with a 6 colour bin, where based on a pixel's individual colour code extracted from the CIExy chromaticity chart, they are stored. This, as the evaluation shows, adds noticeably to the precision, an average increase of almost 20%. The increase in computational expense of constructing 2SD is limited as the cluster information required for assigning colour is predefined.

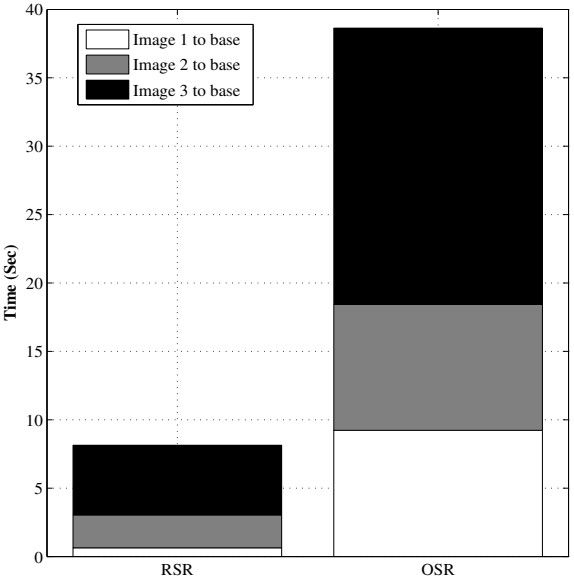


Figure 3.22: *Showing time taken to match features between images in Figure 3.21(a). Time taken is broken down in to each time an image is to be matched to the base mosaic.*

We have also included a novel feature clustering technique based on colour codes that allows for precise and fast feature matching by reducing the search region for finding candidate matches. Example tests show how a 5 times increase in matching speed when dealing with a number of images.

4
Robust Homography Estimation

Due to distortions caused by real world cameras imaging a scene, common features between images do not completely fall under a projective mapping (Chapter 2). This leads to uncertainty in the feature's location and manifests itself as a measurement error in model estimation, i.e., estimating the homography $\mathcal{H}$. In this chapter, we explore techniques that allow us to deal with feature location uncertainty and estimate a robust $\mathcal{H}$. We begin by exploring how normalised feature are used to estimate $\mathcal{H}$ with a reduced error. Normalised points are predominantly used to tackle ill conditioned systems encountered when using pixel locations, but they also have an inherent capability to deal with location error, as will be shown.

Then we introduce a novel recursive least squares (RLS) solution for estimating homographies. The use of such a technique comes from its ability to deal with corrupted and periodic measurements to provide the best solution. Furthermore, its capacity for providing reliable results for time varying parameter estimation also motivates its use in the context of real time cooperative image mosaicing [82]. Here, optimal transformation between mobile platforms is likely to change due to motion.

This is followed by a novel two part technique that reduces image feature location error and consequently uses it to estimate a robust $\mathcal{H}$ with H_∞ filtering. We consider feature information from all three channels of an RGB image and apply information fusion to reduce localisation uncertainty. The novelty of the technique is not in the feature detector itself but in how we associate localisation error and parameter estimates. The H_∞ class of filter used is capable of dealing with system and measurement uncertainty simultaneously [83] and we show how it outperforms

covariance weighted optimisation techniques. Additionally, we introduce a novel coupled H_∞ filter application for simultaneously estimating the geometric and photometric transformations between two images. It gives good results with low reprojection error compared to standard $||L||_2$-norm techniques. The filter is also tested for time varying parameter estimation and shows promising results.

Finally, we compare sparse non linear optimisation to the Kalman and H_∞ filter for good and bad initial estimates. We show that the H_∞ filter performs best for cases that are badly initialised and, also, that it is computationally less expensive.

4.1 Estimation via Feature Location Normalisation

Recalling from Chapter 2, we can estimate the parameters of $\mathcal{H}$ by solving the linear system $C\mathbf{h} = 0$, where $\mathbf{h}$ consists of the parameters of the homography and C is populated with location of feature matches. As already mentioned, feature locations contain error. This combined with the fact that using feature locations in pixels can cause ill-conditioned systems, leads to a bad estimate of the homography [84].

The point transformation introduced in [84] deals with this issue by translating and then scaling the feature locations to reduce their sensitivity to noise and, therefore, improve the systems condition number χ [1]. They are

- The points are translated so that their centroid is at the origin
- The points are then isotropically scaled so that the average distance from the origin is equal to $\sqrt{2}$

Let $\mathbf{x}_i$ represent a vector of feature locations for set "i", then the normalised points are given by

[1] measures how much the output value of the function can change for a small change in the input argument

$$\hat{\mathbf{x}}_i = \begin{bmatrix} \tau_i & 0 & -\tau_i x_{c_i} \\ 0 & \tau_i & -\tau_i y_{c_i} \\ 0 & 0 & 1 \end{bmatrix} \mathbf{x}_i \tag{4.1}$$

where x_{c_i} and y_{c_i} are the means of the locations in x and y and τ_i is the scaling factor given by

$$\frac{\sqrt{2}}{\sum_{i=1}^{k} \sqrt{(x_i - x_{c_i})^2 + (y_i - y_{c_i})^2}/k}. \tag{4.2}$$

A homography estimated using normalised points $\hat{\mathbf{x}}$ and $\hat{\mathbf{x}}'$ is de-normalised using

$$\mathcal{H} = inv\,(T_2)\,\hat{\mathcal{H}}\,T_1 \tag{4.3}$$

where T_i is the normalisation matrix in Equation (3.1).

Experimental analysis Figure 4.1 shows a couple of images of the same scene[2]. The right hand side of the figure is the transformed image and is done so by applying a known homography. We estimate the underlying $\mathcal{H}$ between this couple by solving a linear system $C\mathbf{h} = 0$ from normalised and un-normalised image features. To simulate location uncertainty, we introduce error into the feature locations.

At this stage, we introduce the error metric used for comparison of model estimation techniques in terms of goodness of $\mathcal{H}$. We call this error the back projection error e_{bp} given by

$$e_{bp} = \frac{\sum^{k} ||\mathbf{x} - \mathcal{H}\mathbf{x}'||}{k} \tag{4.4}$$

where $||.||$ is Euclidean distance. This error determines the average back projection error of features projected from one image to another through $\mathcal{H}$.

Figure 4.2 compares e_{bp} from estimation of $\mathcal{H}$ between images in Figure 4.1 using normalised and unnormalised feature locations. A location uncertainty of 0.05 pixels is introduced for case given in Figure 4.2(a) and 0.5 pixels in Figure 4.2(b).

[2]The Hilton hotel in Manchester, UK

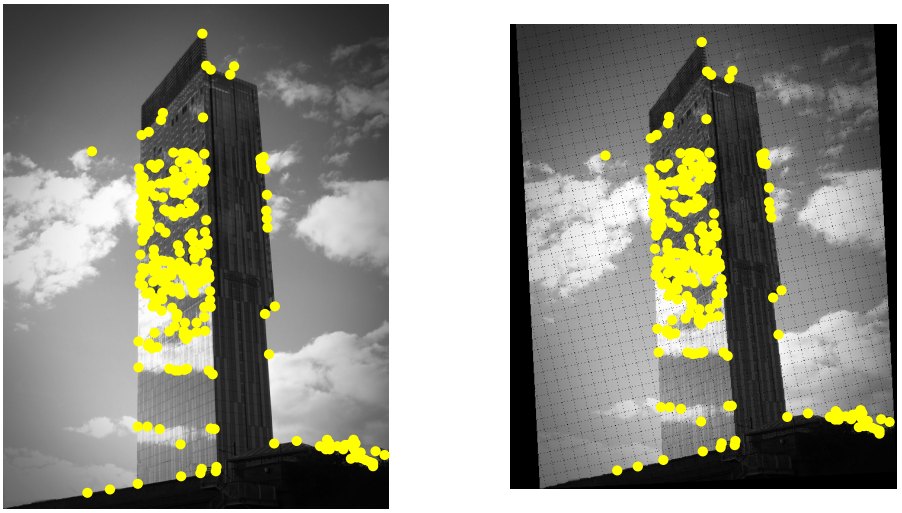


Figure 4.1: *Set of overlapping images with Harris features. Right hand side image transformed with known homography.*

We can easily see that using normalised feature locations makes the system more robust to noise. For Case 1, using normalisation results in almost 50% less e_{bp} compared to estimation with unnormalised feature locations. For Case 2, the difference is greater than 50%.

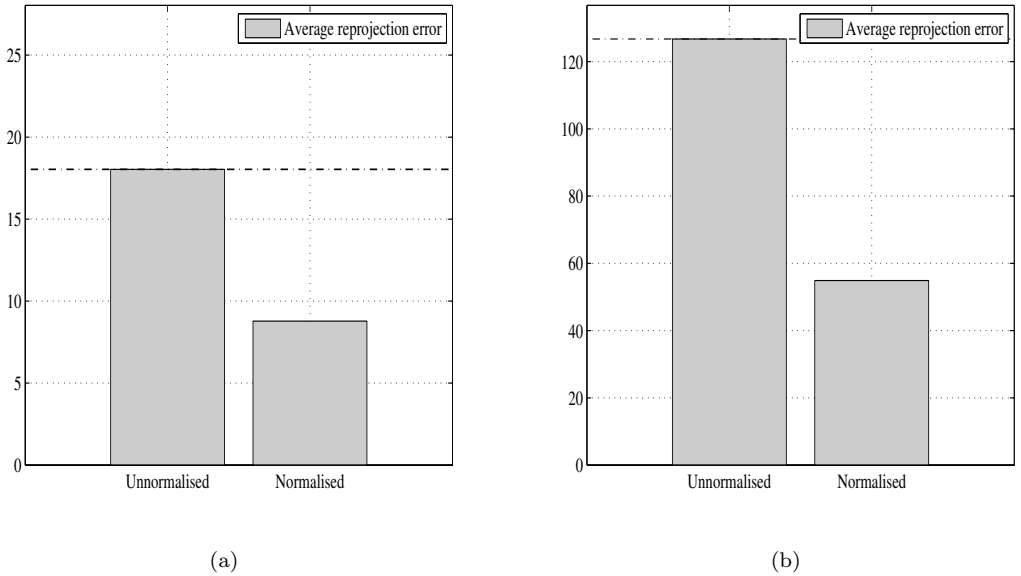


Figure 4.2: *Back projection errors from normalised and unnormalised points tested on image in Figure 4.1. (a) Case 1: with 0.05 pixel error in feature locations. (b) Case 2: with 0.5 pixel error in feature locations.*

In terms of χ, the value is of order of magnitude 6 for the unnormalised system and is of order of magnitude 1 for the normalised system. The low condition number for the normalised system is the reason why it is more robust to slight changes in parameters, i.e., feature location error.

Figure 4.3 gives another example of a set of overlapping images for which the underlying $\mathcal{H}$ is estimated from normalised and un-normalised image feature locations. The images are courtesy of our sponsors BAE Systems. The ground truth homography is known, as previously. Figure 4.4(a) and 4.4(b) compare e_{bp} for two different localisation errors, 0.05 and 0.10 pixels. It is apparent that for both cases using normalised coordinates provides better robustness to noise. For case 1 the error is 58% less for the normalised coordinates and is almost the same for case 2. The condition number χ, the value is again of order of magnitude 5 for the unnormalised system and is of order of magnitude 1 for the normalised system.

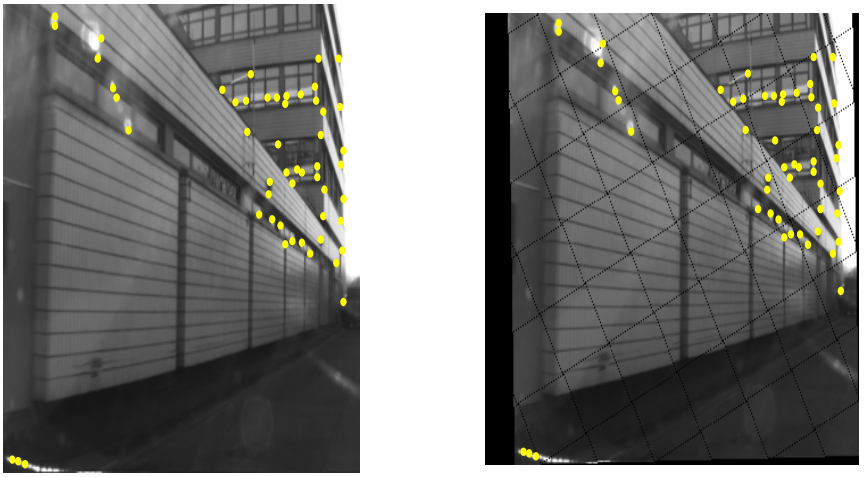
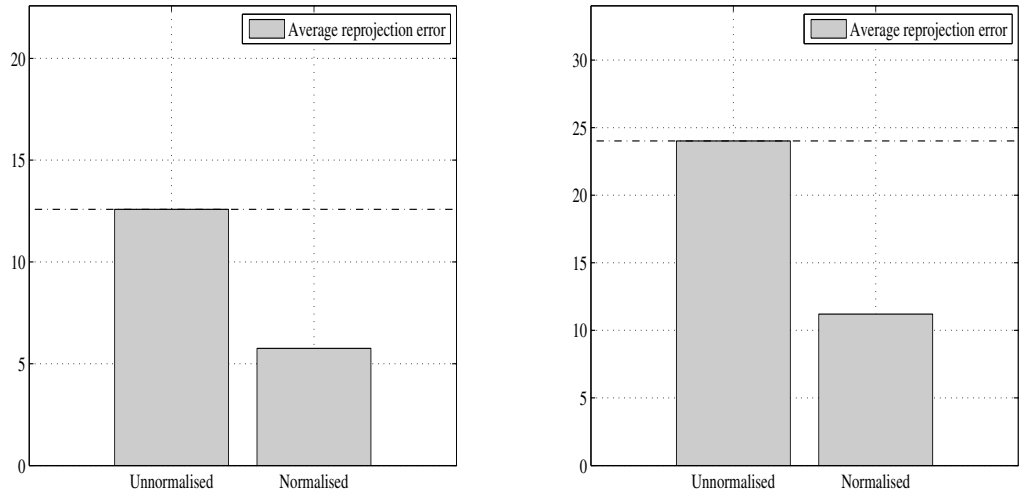


Figure 4.3: *Another example of a set of overlapping images with Harris features. Left hand side image transformed with known homography. Images are courtesy of our sponsors BAE systems.*

It is to be noted that we introduce a pixel error in to each feature artificially. Using exact values for uncertainty will be tackled shortly. Nevertheless, using normalised points does provide better estimates of the homography for real world images.



(a) (b)

Figure 4.4: *Back projection errors from normalised and unnormalised points tested on image in Figure 4.3. (a) Case 1: with 0.05 pixel error in feature locations. (b) Case 2: with 0.10 pixel error in feature locations.*

4.2 Recursive Least Squares Solution for $\mathcal{H}$

Up until now, we have only looked at a homogenous solution for solving $\mathcal{H}$, Chapter 2. Here, an inhomogenous solution in a recursive least squares RLS sense is developed. First, we need to revisit the system of equations for estimating the homography. Reproducing the homographic relation given by Equation (2.6) below

$$x^{'}\left(ux+vy+sz\right)-r_1x-r_2x-t_xz=0$$
$$y^{'}\left(ux+vy+sz\right)-r_3x-r_4x-t_yz=0.$$

By imposing scale factor $s=1$ an inhomogenous system of equations can be formed

$$\begin{bmatrix} x & y & 1 & \mathbf{O} & -xx^{'} & -yx^{'} \\ \mathbf{O} & -x & -y & -1 & xy^{i} & yy^{'} \end{bmatrix} \mathbf{h} = \begin{pmatrix} x^{'} \\ y^{'} \end{pmatrix} \tag{4.5}$$

where $\mathbf{O}$ is a 3 vector of zeros and $\mathbf{h}$ is a vector of parameters of $\mathcal{H}$ without s. The system takes the form $C\mathbf{h}=b$. For an over-determined solution, i.e., more than 4 feature matches, the system can be solved by the normal equation

$$\mathbf{h}^* = \left(C^T C\right)^{-1} C^T b \tag{4.6}$$

which minimises the square error by finding the projection of b in the column space of C.

In the case of noisy measurements, which are expected, $\mathbf{h}$ will not be exact. The beauty of putting the estimation problem into a inhomogenous form is that future measurements can be taken into account to update the solution recursively. In the context of cooperative image mosaicing, once an initial homography is established it can be updated periodically by a single image feature correspondence. Therefore, changing homographies can be tracked between platforms.

The recursive form for least squares solution for $\mathbf{h}$ is given by the following equations [82].

$$\mathbf{h}_k = \mathbf{h}_{k-1} + K_k \left(b_k - C_k \mathbf{h}_{k-1}\right) \tag{4.7}$$

$$K_k = \frac{\Phi_{k-1} C_k}{\lambda + C_k' \Phi_{k-1} C_k} \tag{4.8}$$

$$\Phi_k = \frac{\left(I + K_k b_k\right) \Phi_{k-1}}{\lambda} \tag{4.9}$$

where K_k is the gain, λ is the forgetting factor and Φ_k is the covariance matrix. C and b are the system and measurement matrix, respectively. Equation (4.7) is the update equation that updates the solution conditioned on the new and old measurement and the forgetting factor.

Experimental analysis We start by showing results from an implementation of RLS to synthetic data. Using a ground truth homography, known data points $\mathbf{x}$ are transformed to $\mathbf{x}'$. A corrupted initial estimate of $\mathbf{h}$ (from ground truth $\mathcal{H}$) together with erroneous measurement data (feature locations) of order of magnitude 1 is input to RLS. Figure 4.5 shows decreasing e_{bp} for increasing measurements with

a bad initial estimate and corrupted measurements. An almost 50% decrease in error is observed.

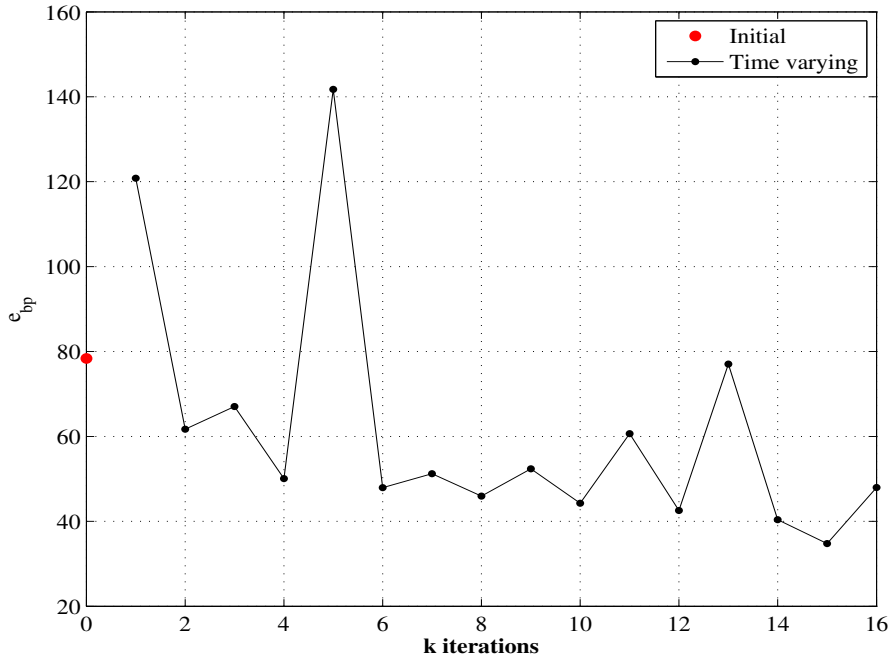


Figure 4.5: *Decreasing e_{bp} with increasing number of measurements (feature points). The red marker indicates initial error. $\Phi = 1e5$, $\lambda = 0.95$.*

We now test RLS on real image data and also compare it to the non-linear optimisation technique called Levenberg-Marquardt (LM). Figure 4.6 gives a couple of overlapping images for which the underlying $\mathcal{H}$ is determined. The RLS is initialised with a heavily corrupted initial estimate and further erroneous measurements of $\Delta = 1$ are added to it, where Δ is the error added to the feature location. The decrease in error towards zero is clearly visible in Figure 4.6(b). At the final iteration e_{bp} is more than 10% less compared to the initial estimate. Figure 4.6(c) compares RLS to the non-linear LM optimisation algorithm which is used in similar fashion to RLS to refine the model. The cost function used for optimisation is given in Chapter 2 Section 2.3.1, Equation (2.7). It is obvious from the figure that RLS performs much better compared to LM optimisation with the same initialisation. This is because LM, as most optimisation techniques, require accurate initialisations. RLS, on the other hand, does not require a good initialisation and derives the error down regardless of it. Furthermore, RLS is computationally less expensive compared to LM as it does not require the Jacobian (matrix of first

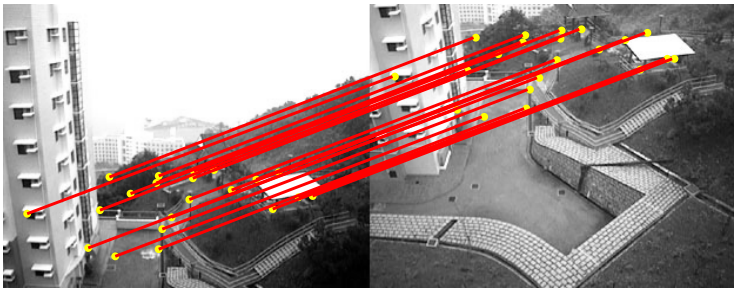
order partial derivatives) at every step. This is quantified in Section 4.6 of this chapter.

Figure 4.7 is example of two overlapping images for which underlying $\mathcal{H}$ is determined. Similar trends as in the previous case are observed. The RLS, after an initial spike, reduces e_{bp} to less than 2 after 12 iterations to almost a steady state value. Comparing it to LM optimisation, RLS has 4% less error. It is to be noted that the initial error in this example is less compared to the previous case and RLS still performs better compared to the expensive non-linear optimisation.

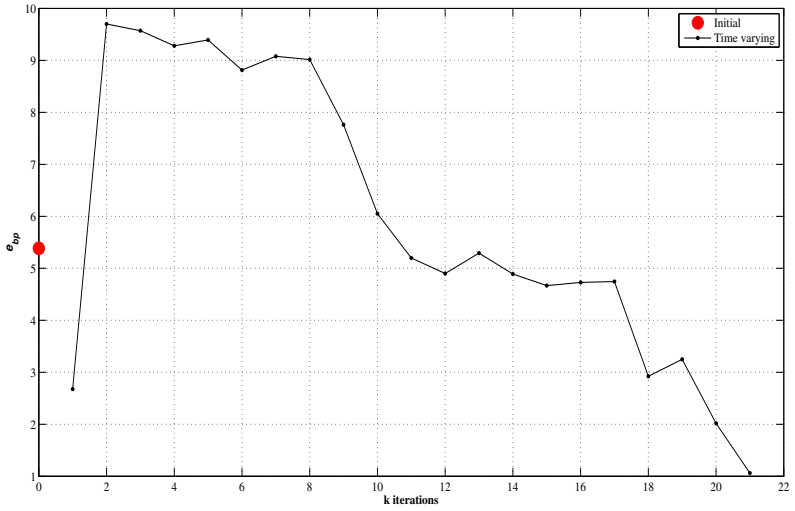
Figure 4.8 is an example of overlapping images for which the underlying homography is estimated. Again RLS performs well compared to LM optimisation. The error e_{bp} decreases, after an initial spike, when more measurements are introduced and the final value is 15% less compared to the LM optimisation.

Figure 4.9 is an example of overlapping images from BAE Systems for which the underlying homography is estimated. Again, RLS performs well compared to LM optimisation. The error is more than 60% less for RLS.

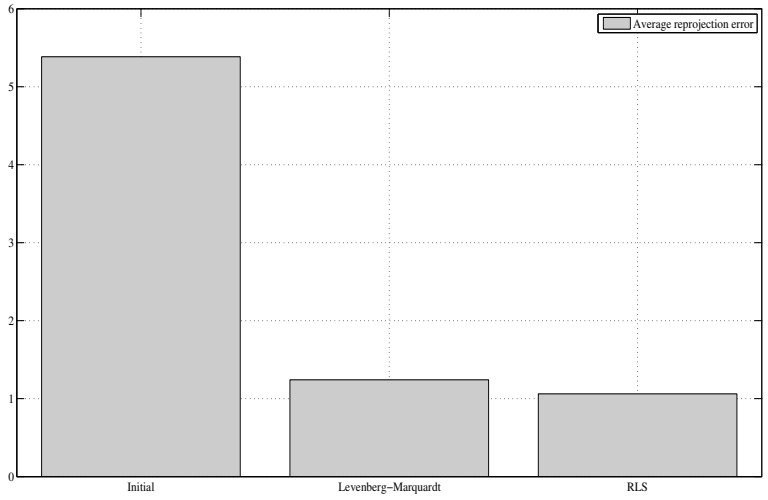
The initial spikes for all cases can be due to a high feature location error. The RLS technique, nonetheless, deals with this uncertainty and drives the error down.



(a)

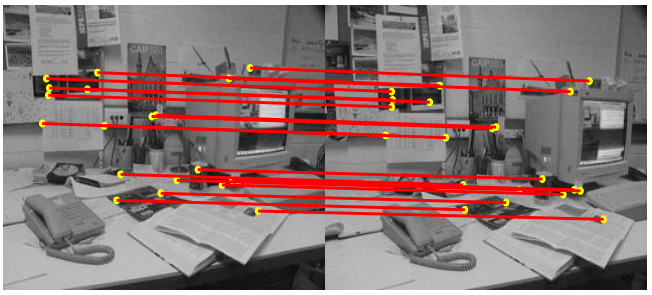


(b)

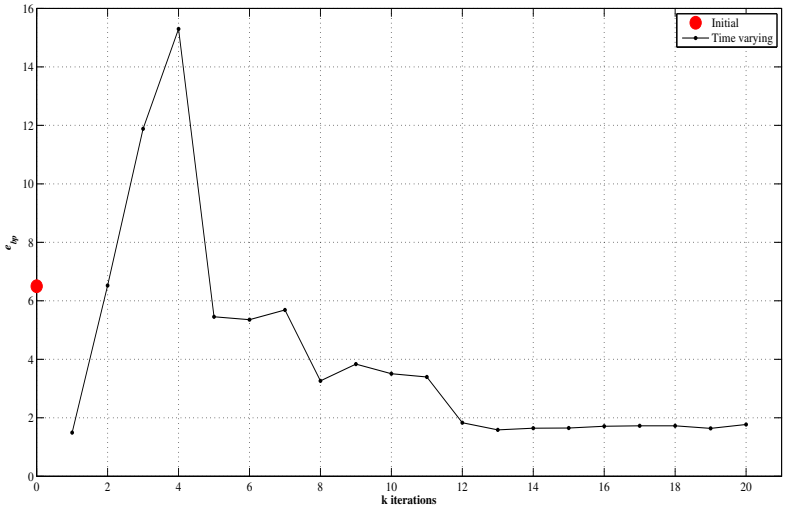


(c)

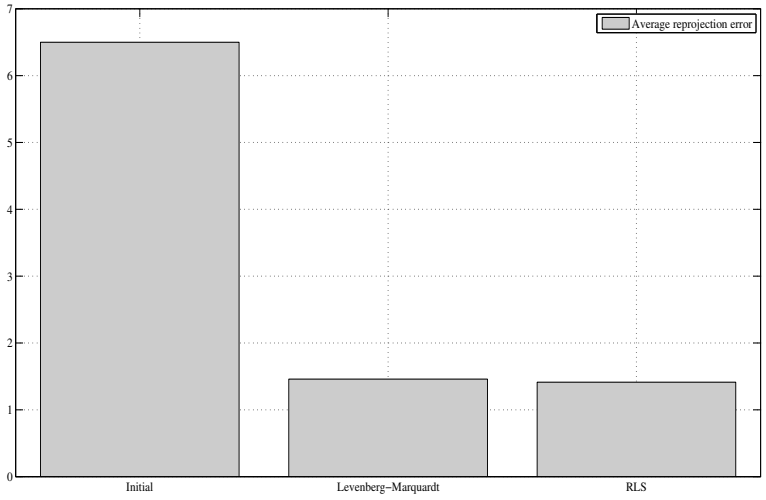
Figure 4.6: *Analysis of RLS on real image data. (a) A set of overlapping images. Harris features are used with 1SD feature descriptors. (b) Decreasing e_{bp} with added erroneous measurements. Red dot is error from initial estimate. (c) Comparing RLS to non linear LM optimisation. RLS performs much better compared to LM for same initial estimate.*



(a)

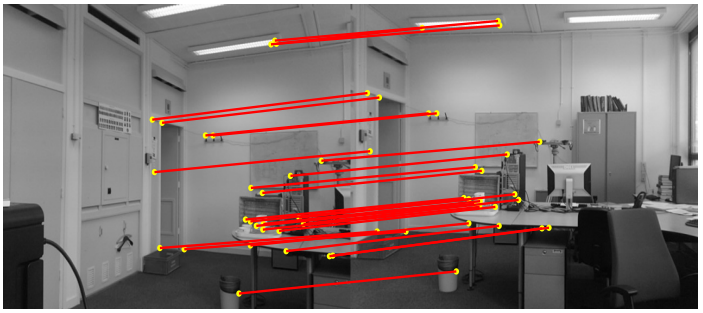


(b)

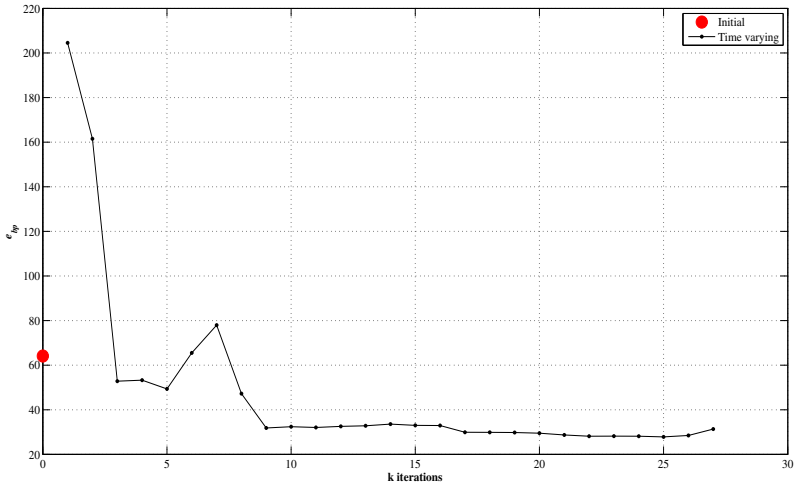


(c)

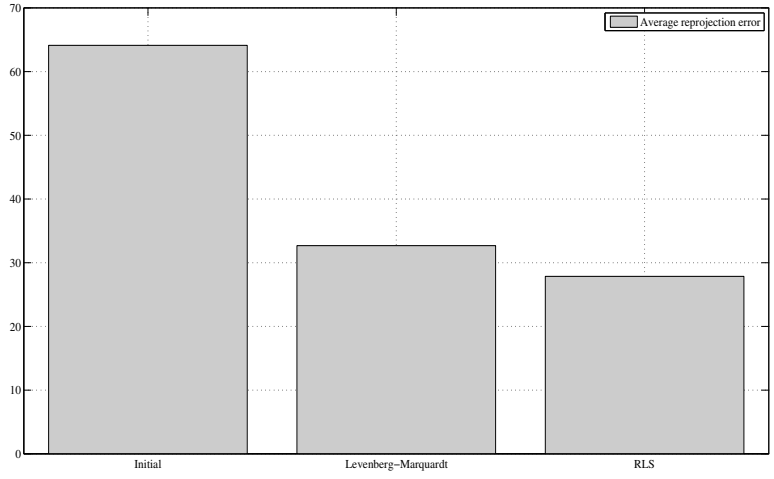
Figure 4.7: *Analysis of RLS on real image data. (a) A set of overlapping images. Harris features are used with 1SD feature descriptors. (b) Decreasing e_{bp} with added erroneous measurements. Red dot is error from initial estimate. (c) Comparing RLS to non linear LM optimisation. RLS performs better compared to LM for same initial estimate.*



(a)

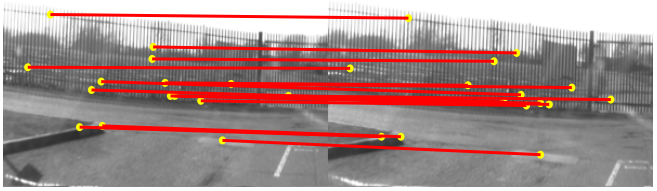


(b)

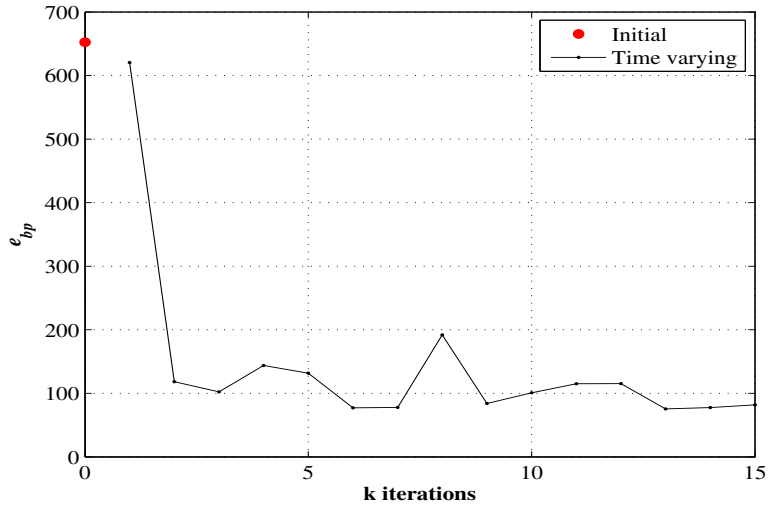


(c)

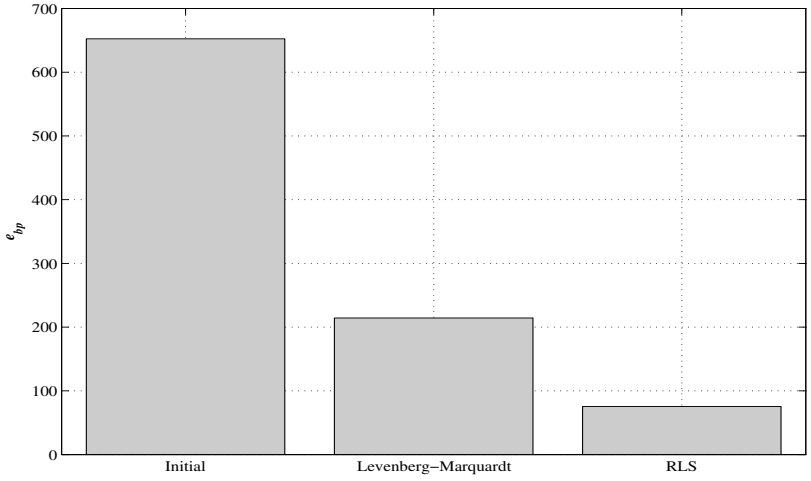
Figure 4.8: *Analysis of RLS on real image data. (a) A set of overlapping images. Harris features are used with 1SD feature descriptors. (b) Decreasing e_{bp} with added erroneous measurements. Red dot is error from initial estimate. (c) Comparing RLS to non linear LM optimisation. RLS performs much better compared to LM for same initial estimate.*



(a)



(b)



(c)

Figure 4.9: *Analysis of RLS on real image data from BAE systems. (a) A set of overlapping images. Harris features are used with 1SD feature descriptors. (b) Decreasing e_{bp} with added erroneous measurements. Red dot is error from initial estimate. (c) Comparing RLS to non linear LM optimisation. RLS performs much better compared to LM for same initial estimate.*

4.3 Robust $\mathcal{H}$ Estimation with Reduced Feature Location Uncertainty

Feature detection comes with associated localisation errors. These localisation errors are well studied in literature [85–87]. Most approaches subscribe to the Gaussianity of the error and so characterise the uncertainty using a 2D covariance matrix. The same assumption will be followed here. Intuition dictates that incorporating this uncertainty information in terms of a weighted cost function to be minimised should lead to better parameter estimates. In [85], however, it is argued that incorporating localisation error in estimating the homography does not lead to substantially improved results. In [86] on the other hand, a reduced error estimate for the fundamental matrix after including uncertainty covariance estimates. The same conclusion is drawn in [87] who consider feature covariances over different image scales.

Following on from this, we argue that measuring and incorporating feature localisation errors does improve estimates of $\mathcal{H}$. First, we propose a novel technique for reducing localisation errors of features using covariance intersection (CI) by fusing uncertainty information from all three channels of an RGB image. The novelty of the technique is not in the detector itself but in how we associate localisation error and parameter estimates. We then use this reduced localisation error to estimate an improved $\mathcal{H}$ by recasting the problem into a class of robust H_∞ filter capable of overcoming feature location uncertainty. We show that our technique outperforms covariance weighted optimisation techniques for estimating the homography.

4.3.1 Feature Location Uncertainty

To reduce feature location uncertainty, we first need to quantify it. Here we introduce a way to determine uncertainty for every feature in an image.

Let $\mathbf{x}(x, y)$ represent an image feature obtained from a gradient based feature extractor, i.e., Harris corner detector [60, 71]. The true position of this feature is

given by $(\tilde{x}, \tilde{y})$. Errors in its location are then $\Delta x = x - \tilde{x}$ and $\Delta y = y - \tilde{y}$. If these are considered as random variables, then we can define their covariance matrix as

$$\mathcal{P} = \begin{bmatrix} E\left[\Delta x^2\right] & E\left[\Delta xy\right] \\ E\left[\Delta xy\right] & E\left[\Delta y^2\right] \end{bmatrix} \tag{4.10}$$

where $E\left[.\right]$ denotes expectation. The covariance matrix $\mathcal{P}$ gives the spread of uncertainty of the feature $\mathbf{x}$ in the axial directions. Indeed, this spread is characterised as a Gaussian distribution, as will be shown [85, 87].

There are two techniques for determining $\mathcal{P}$ as described in [85]. One is a residual based approach and the other a derivative based approach. We employ the latter for its ease of use and implementation. The same technique is used in [87].

We define a matrix of first order partial derivates squared for spatial coordinates x and y as follows

$$H = \begin{bmatrix} \sum_{(x,y)\epsilon N_p} w_{xy} I_x^2 & \sum_{(x,y)\epsilon N_p} w_{xy} I_x^2 I_y^2 \\ \sum_{(x,y)\epsilon N_p} w_{xy} I_x^2 I_y^2 & \sum_{(x,y)\epsilon N_p} w_{xy} I_y^2 \end{bmatrix} \tag{4.11}$$

where I_x and I_y denote the partial derivatives, w_{xy} is a weighting function, normally Gaussian and N_p is the neighbourhood of the feature.

The above expression, known as the second moment matrix, describes the curvature distribution around a point and is the same function as used in the Harris corner detector, Chapter 3. The greater the change in curvature the more accurately the corner can be located and vice versa. We therefore define the covariance as the inverse of this expression. A small covariance implying a large change in curvature. A similar expression is used in [87] where feature uncertainty is quantified over different scales.

It is worth highlighting that throughout this study we use the Harris feature detector to measure features from images. An objective of our work is to introduce a novel technique to reduce feature location uncertainty that can be used for parameter estimation via covariance weighted optimisation. Incidentally, although features extracted via SIFT or SURF are scale invariant, they still have inherent localisation errors as concluded in [87].

Once the covariance is estimated for a feature $f(x, y)$, it can be visualised using error ellipses as shown in Figure 4.10.

Figure 4.10: *Harris features with location covariances visualised as error ellipses. The ellipses are centralised over each detected feature (yellow).*

4.3.2 Reducing Feature Uncertainty Using Fusion

Now that a way to determine feature location covariance has been established, we proceed by introducing a method to reduce it.

When using image features for model estimation, most of the feature detection is done on grey scale images. Even when the input is a colour image it is first converted to a grey level image, resulting in loss of information. In the context of feature location, as weighted sums are taken over the three bands some of the sharpness is lost which is reflected on the feature's true location and associated localisation error.

Figure 4.11 gives a colour image with associated RGB channels. As each channel produces a different response to a stimulation, i.e., reflection of a feature, we can use information fusion to get better estimates of feature location uncertainty

with reduced covariances. This is done by employing a covariance intersection (CI) algorithm. Figure 4.12 shows common features between all three channels and their associated covariance (covariances are shown over a grayscale image for clarity).

Figure 4.11: *A colour image with its associated RGB channels.*

Covariance Intersection for Reducing Location Uncertainty

CI is a technique that allows us to fuse information from different sources with varying uncertainty for better estimates. As a result, it finds use in data fusion architectures where information about signal sources is incomplete [88]. For example, positional data from two different sensors or a measurement from a sensor and an estimation from a prediction model.

Let $\hat{j}_1,\ \hat{j}_2 ...\ \hat{j_N}$ represent unbiased estimates of an unknown state vector j_0, i.e.,

$$E\left[\hat{j_n}\right] = j_0,\ (n = 1, 2..., N)\,. \tag{4.12}$$

Figure 4.12: *Image feature with error bounds given as ellipsoids. The three different covariances from the RGB channels are visible. Covariances from each respected channel is coloured according to its respective channel.*

The corresponding covariances matrices for each such estimate is given by $\mathcal{P}_1$, $\mathcal{P}_2$... $\mathcal{P}_N$. It is assumed that the estimates are consistent

$$\mathcal{P}_n - \tilde{\mathcal{P}}_n \geq 0, (n = 1, 2..., N) \tag{4.13}$$

where

$$\tilde{\mathcal{P}}_n = E\left[\left(\hat{j}_n - j_0\right)\left(\hat{j}_n - j_0\right)^T\right] = \left[\tilde{j}_n \tilde{j}_n^T\right] \tag{4.14}$$

denotes the covariance matrix of the n-th estimate $\hat{j}_n$.

The CI filter is given by the convex combination

$$\mathcal{P}_0^{-1} = \sum_{n=1}^{N} = w_n \mathcal{P}_n^{-1} \tag{4.15}$$

$$\mathcal{P}_0^{-1}\hat{j}_0 = \sum_{n=1}^{N} = w_n \mathcal{P}_n^{-1}\hat{j}_n \tag{4.16}$$

where $0 \leq w_n \leq 1$.

An analytical procedure for determining the weighting coefficients is given in [88]. For $N \geq 2$, the nonnegative weighting coefficients $w_1, w_2..., w_n$ are determined under the linear constraint

$$w_1 + w_2 + ... + w_N = 1. \tag{4.17}$$

The second constraint applied is

$$tr\left(\mathcal{P}_n\right) w_n - tr\left(\mathcal{P}_{n+1}\right) = 0, (n = 1, 2..., N)\,. \tag{4.18}$$

With $\mathcal{E}_n := tr\left(\mathcal{P}_n\right)$, combining the two constraints yields the linear system

$$\begin{bmatrix} \mathcal{E}_1 & \mathcal{E}_2 & 0 & ... & 0 \\ 0 & \mathcal{E}_2 & -\mathcal{E}_3 & ... & 0 \\ ... & ... & ... & ... & ... \\ ... & ... & ... & \mathcal{E}_{N-1} & -\mathcal{E}_N \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ ... \\ w_{N-1} \\ w_N \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ ... \\ 0 \\ 1 \end{bmatrix} \tag{4.19}$$

solving which gives the required weights.

Illustrative example Consider an example. Let $\hat{j}_1, \hat{j}_2$ denote two unbiased estimates of the unknown state vector x_0 with covariance matrices

$$\mathcal{P}_1 = \begin{bmatrix} 1 & 0 \\ 0 & 4 \end{bmatrix},\ \mathcal{P}_2 = \begin{bmatrix} 7 & 0 \\ 0 & 0.5 \end{bmatrix}. \tag{4.20}$$

We use Equation 4.19 to first determine the weighting coefficients, which together with the original covariances is applied to Equation 4.15 to update the covariance of our estimate. Figure 4.13 shows the error ellipses corresponding to original estimates and the intersected covariance. The new covariance estimate has a trace less compared to the original estimates $tr\left(\mathcal{P}_0\right) = 1.6$. The results from this illustrative example follow the claims made previously. The weighting coefficients are chosen in such a way as to reduce this trace. More advanced methods based

on optimisation are also present to determine these weighting coefficients but are computationally expensive [89].

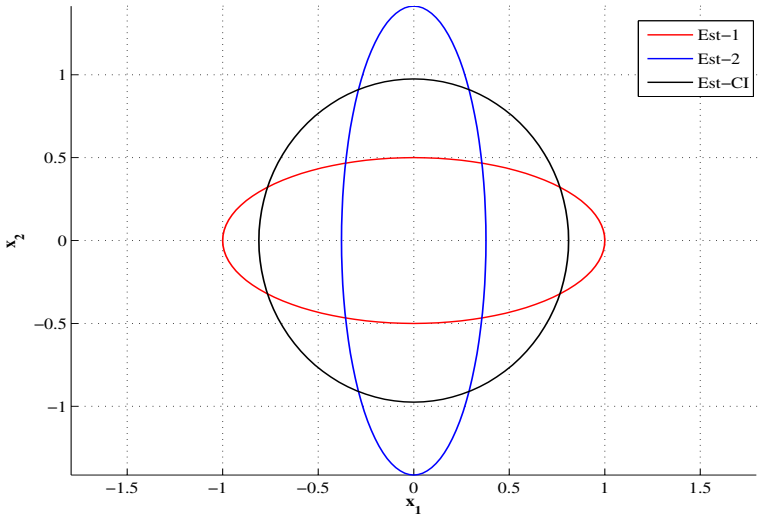


Figure 4.13: *Red and blue lines give covariances of the two inputs shown as error ellipses and the black error ellipse shows the result from CI.*

4.3.3 Robust H_∞ Estimation of $\mathcal{H}$

Now we introduce the robust L_∞ norm minimisation technique to estimate an optimum $\mathcal{H}$. This is a filtering technique that takes into account feature location uncertainty just defined and is an innovative way to estimate the homography.

The filtering technique here minimises the L_∞ norm instead of the L_2 norm used in the classic Kalman filter and the Direct Linear Transformation (DLT) algorithm for parameter estimation [24, 84]. Very recently, the L_∞ norm has been adopted within the computer vision community as a norm of choice to solve some geometric vision optimisation problems [90, 91]. Localising non-overlapping cameras using second order cone programming (SOCP) to minimise the L_∞ norm is done in [91]. It shows good performance of SOCP for camera centre localisation with relatively low errors. In [90], it is shown that by using the infinity norm a number of vision problems like homography estimation as a quasiconvex problem can be formed and solved using the bisection-method but with linear programming. Again using linear programming, [92] solve the structure and motion problem using the ∞-norm approach.

Although latest developments for L_∞ norm based optimisation can provide accurate and globally minimum solutions, the technique used is computationally heavy. Our method, on the other hand, adopts an L_∞ norm based filtering technique which is computationally attractive to bound the worse-case error estimate to solve the geometric problem of computing the homography.

The H_∞ Filter with Uncertainty

Re-writing Equation (4.5) below gives

$$\begin{bmatrix} x & y & 1 & \mathbf{O} & -xx^{'} & -yx^{'} \\ \mathbf{O} & -x & -y & -1 & xy^{i} & yy^{'} \end{bmatrix} \mathbf{h} = \begin{pmatrix} x^{'} \\ y^{'} \end{pmatrix}$$

in compact form

$$C\mathbf{h} = b. \tag{4.21}$$

Now the uncertainty in feature locations for each correspondence k can be written as

$$(C + \Delta C)_k \, \mathbf{h}_k = b_k \tag{4.22}$$

where ΔC represents feature uncertainty (or modelling uncertainty) and b_k is the measurement. The above equation is similar to the linear expression in Equation 4.21. Our aim is to determine an optimum $\mathcal{H}$ utilising the uncertainty information in the system.

Introducing a scheme that can deal with systems such as Equation (4.22), a system with modelling uncertainties can by given by

$$\begin{aligned} \mathbf{h}_{k+1} &= (A_k + \Delta A_k)\, \mathbf{h}_k + w_k \\ b_k &= (C_k + \Delta C_k)\, \mathbf{h}_k + v_k \end{aligned} \tag{4.23}$$

where $\{w_k\}$ and $\{v_k\}$ are uncorrelated zero-mean white noise processes with covariances Q_k and R_k. Matrices ΔA_k and ΔC_k represent uncertainty in the system and are given by

$$\begin{bmatrix} \Delta A_k \\ \Delta C_k \end{bmatrix} = \begin{bmatrix} H_{1,k} \\ H_{2,k} \end{bmatrix} F_k N_k \tag{4.24}$$

where $H_{1,k}$, $H_{2,k}$ and N_k are known matrices with appropriate dimensions and F_k is the norm-bounded time varying uncertainty satisfying $F_k^T F_k < I$ [83, 93, 94].

The problem is to design a state estimator of the form

$$\hat{\mathbf{h}}_{k+1} = \hat{\mathcal{A}}_k \hat{\mathbf{h}}_k + \hat{\mathcal{K}}_k b_k \tag{4.25}$$

where $\hat{\mathbf{h}}$ is the state estimate and $\hat{\mathcal{A}}_k$ and $\hat{\mathcal{K}}_k$ are the filter parameters to be determined with the following characteristics [93]

- The estimator is stable, i.e., the eigenvalues of $\hat{\mathcal{A}}_k$ are less than one in magnitude.
- The estimation error $\tilde{\mathbf{h}}_k$ satisfies the following worst-case bound

$$max_{w_k, v_k} \frac{\| \tilde{\mathbf{h}}_k \|_2}{\| w_k \|_2 + \| v_k \|_2 + \| \tilde{\mathbf{h}}_0 \|_{S_1^{-1}} + \| \mathbf{h}_0 \|_{S_2^{-1}}} \tag{4.26}$$

- The estimation error $\tilde{x}_k$ satisfies the following RMS bound

$$\mathcal{E}(\tilde{\mathbf{h}}_k \tilde{\mathbf{h}}_k^T) < \Theta_k. \tag{4.27}$$

The solution is given by

$$\begin{aligned} \hat{A}_k = A_k + \left(A_k - \hat{\mathcal{K}}_k C_k \right) \Theta_k E_k^T \\ \left(k_k^{-1} I - E_k \Theta E_k^T \right)^{-1} E_k \end{aligned} \tag{4.28}$$

and

$$\hat{\mathcal{K}}_k = \left[k_k^{-1} H_{1,k} H_{2,k}^T + A_k \left(\Theta_k^{-1} - k E_k^T E_k\right)^{-1} C_k^T\right] R_{1,k}^{-1} \tag{4.29}$$

where

$$\begin{aligned} R_{1,k} = R_k + k_k^{-1} H_{1,k} H_{2,k}^T + C_k P_k C_k^T \\ C_k \left(\Theta_k^{-1} - k E_k^T E_k\right)^{-1} C_k^T \end{aligned} \tag{4.30}$$

such that the state estimation error variance satisfies the boundedness condition. The parameter k_k is a sequence of positive scalars.

It is worth noting that in the framework proposed here, the system matrix A_k in Equation 4.23 is the identity matrix.

Experimental Analysis with synthetic data Here, we show capability of our H_∞ filter to robustly estimate $\mathcal{H}$ in the presence of feature location uncertainty with synthetic data. A detailed evaluation on real world image data is included in the next section.

We begin by formulating the homography estimation into a filtering problem. This is done as given below. Consider the left part of Equation 4.5. Adding uncertainty Δ in the pixel location leads to

$$C + \Delta C = \begin{bmatrix} \tilde{x} & \tilde{y} & 1 & \mathbf{O} & -\tilde{x}\tilde{x}' & -\tilde{y}\tilde{x}' \\ \mathbf{O} & \tilde{x} & \tilde{y} & 1 & -\tilde{x}\tilde{y}' & -\tilde{y}\tilde{y}' \end{bmatrix} \tag{4.31}$$

where $\tilde{m} = m + \Delta m$. For Δ we use the standard deviations σ_d for each individual matched feature obtained from its location uncertainty covariance matrix, Section 4.3.2. We also introduce error in the measurement w_k and again quantify it using the standard deviations from the covariance matrix. The measurement equation therefore becomes

$$b_k = (C + \Delta C)\, \mathbf{h}_k + w_k. \tag{4.32}$$

In our formulation the location uncertainty is assumed to be in the second image. This expression is similar to Equation 4.23.

The filter is initiated with an initial estimate of the homography as an initial state estimate. The optimum state estimate is obtained by solving Equation (4.25) using Equation (4.28) and (4.29).

Consider Figure 4.14. Two data sets are related through a ground truth homography $\mathcal{H}_g$ (top figure). We estimate $\mathcal{H}$ for these given sets of data points for 31 instances with increasing localisation error and compute the average back projection error for each instance (bottom figure). We then take the average of this error. The localisation error is simulated with a Gaussian with standard deviation σ_d increasing from 0 to 1.5. We do this for 10 trials. The average error from each trial is given in Table 4.1. The filter is initialised with $S_1 = I$ and $S_2 = I$ and $k_k = 0.001$.

The tabulated results clearly show how the average error is kept to an almost steady state value with increasing localisation error. The mean error after filtering is approximately 60 percent less compared to the initial error, even with increasing localisation error.

	Average trial error
$\mathcal{H}_{ini} = 72.6$	20.4
	36.7
	37.1
	19.0
	18.9
	36.0
	18.1
	22.9
	32.3
	27.6

Table 4.1: Average error for each iteration. Within a single iteration the σ_d is increased incrementally. The data set used is given in Figure 4.14, which also shows an iteration.

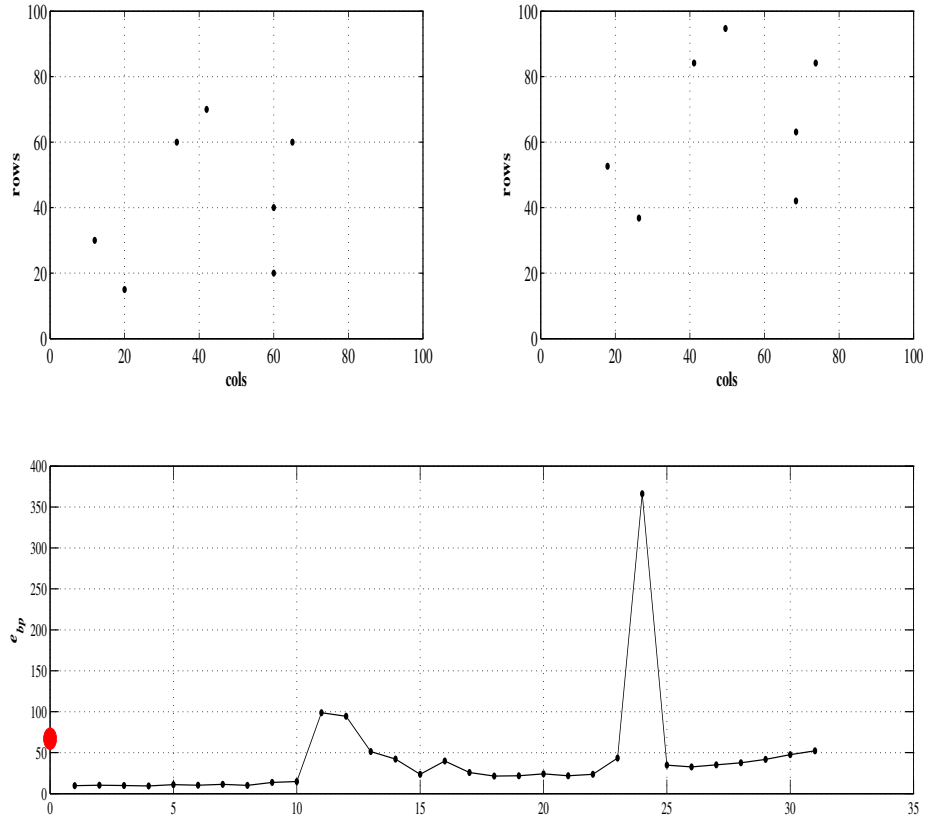


Figure 4.14: *Set of synthetic data related through a known homography $\mathcal{H}_g$. The bottom figure shows e_{bp} as localisation error is increased. The error is Gaussian with increasing σ_d. The x-axis gives the number of runs in a trial.*

4.3.4 Comparison with Covariance Weighted Optimisation

Here, we compare performance of our L_∞ norm based $\mathcal{H}$ estimation technique with standard covariance weighted non-linear optimisation techniques that minimise a weighted cost function [85]. This technique is the only technique available in literature that takes into feature location uncertainty for model estimation [95].

We start by showing how using CI to reduce feature location uncertainty impacts on non-linear optimisation, therefore underlining the need for such a technique. This is followed by a comparison of H_∞ filtering with covariance weighted LM optimisation for estimation of the homography. The tests are performed on real images from the Oxford data set [73].

Results from Non-Linear Optimisation with CI

Here, we show how reducing feature location uncertainty using CI positively affects model estimation. The same covariance weighted estimation technique as in [85] is employed with and without CI for estimating $\mathcal{H}$. This way we can determine whether reducing uncertainty affects estimation of $\mathcal{H}$ or not.

Taking into account feature covariances, a covariance weighted cost function to be minimised is introduced in [85] given as

$$J(\mathcal{H}) = \frac{1}{N}\sum_{k=1}^{N}\left(\mathbf{x}_k^{'} \times \mathcal{H}\mathbf{x}_k, W_k\left(\mathbf{x}_k^{'} \times \mathcal{H}\mathbf{x}_k\right) + (\mathcal{H}\mathbf{x}_k)\right) \tag{4.33}$$

where

$$\begin{aligned} W_k = {} & \mathbf{x}_k^{'} \times \mathcal{H}V_0\left[\mathbf{x}_k\right]\mathcal{H}^T \times \mathbf{x}_k \\ & + (\mathcal{H}\mathbf{x}_k) \times V_0\left[\mathbf{x}_k^{'}\right] \times (\mathcal{H}\mathbf{x}_k) \end{aligned} \tag{4.34}$$

and k is the number of corresponding features. (a, b) denotes the inner product of vectors a and b, $a \times A$ is the matrix whose columns are the vector products of a and the columns of A, $A \times a$ is the matrix whose rows are the vector products of a and the rows of A. $V_0[..]$ for feature set $\mathbf{x}$ or $\mathbf{x}^{'}$ is given by

$$V_0[..] = \begin{bmatrix} \mathcal{P}_{..} & 0 \\ 0 & 0 \end{bmatrix} \tag{4.35}$$

where $\mathcal{P}$ is the covariance for each feature. The uncertainty information is contained within W_k.

The above cost function $J(\mathcal{H})$ gives more weight to features with low error covariance and this allows them to have greater influence on the results. By applying CI we reduce the uncertainty of a feature's locations and in turn its covariance therefore allowing more features to influence the results. This means more information is used.

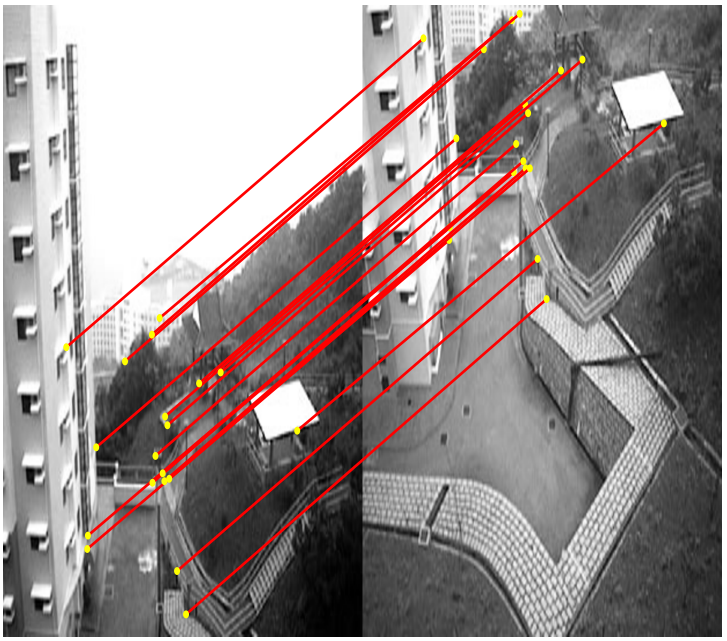


Figure 4.15: *Example 1, a couple of overlapping images used to test CI. Features are Harris features.*

First, we compare the covariance weighted (CW) cost function $J(\mathcal{H})$ to the reprojection (RP) cost function, reproduced below, for a set of overlapping images in Figure 4.15 and 4.16. Covariance intersection is applied to image features used in the weighted cost function. The features are Harris corners and the number is kept low for clarity. The cost functions are minimised using the LM optimisation algorithm with a severely corrupted initial estimate. Please note that all images are colour images but are shown in grey scale to emphasise the feature matches.

$$\sum d\left(\mathbf{x}_k, \mathcal{H}^{-1}\acute{\mathbf{x}}_k\right)^2 + d\left(\acute{\mathbf{x}}_k, \mathcal{H}\mathbf{x}_k\right)^2$$

Table 4.2 gives the back projection error e_{bp} from the final iteration. We can see that taking feature location uncertainty into account to weight the influence of features does yield a better final result. The e_{bp} for the weighted estimated is more than 3 times less compared to the reprojection error cost function for example 1 and 1.3 times less for example 2.

Also shown, is how employing covariance intersection to reduce feature uncertainty impacts on the optimum value of $\mathcal{H}$ compared to not applying CI and using the

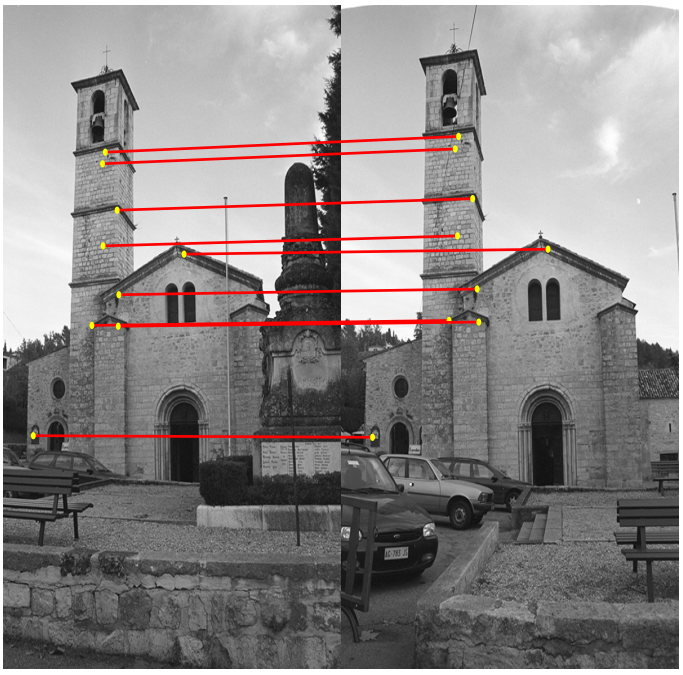


Figure 4.16: *Example 2. A couple of overlapping images used to test CI. Features are Harris features.*

	e_{bp}^{ini}	e_{bp}^{RP}	e_{bp}^{CW}
Example 1	827.3	118.7	33.1
Example 2	443.9	395.3	288.4

Table 4.2: *e_{bp} from the initial estimate and the final iteration of the two cost functions minimised with LM. The two cost functions minimised are the covariance weighted e_{bp}^{CW} and the reprojection error cost function e_{bp}^{RP}. Data is included for two examples, Figure 4.15 and 4.16.*

original covariances. Consider Figure 4.16. We compare errors from optimised results for the homography with and without CI for this data set. This is done for 3 different initial conditions. The results are tabulated in Table 4.3. We optimise over the covariance weighted cost function $J(\mathcal{H})$.

e_{bp}^{ini}	e_{bp}^{CI}	e_{bp}^{noCI}
1944	267	377
43.9	3.85	65.5
415.1	391	435.5

Table 4.3: *Average e_{bp} of the initial estimate and the final iteration of the covariance weighted cost function with and without CI.*

From Table 4.3, we can see that in all three examples reduction in the covariance impacts the optimum value for $\mathcal{H}$ favourably as compared to keeping the original covariance from the grey scale image. By reducing the uncertainty around features we give them more influence on the estimation of the parameters and so more information is used.

Results from H_∞ Filter

We now apply our filter to a set of real images to estimate their inter-image homography given in Figure 4.15 and 4.16. We provide the same initial estimate to our filter as provided to the covariance weighted optimisation estimates. We iterate the filter through the same number of matches.

The e_{bp} at each iteration for example in Figure 4.15 is displayed in Figure 4.17. We can see how after an initial divergence the error reduces to almost a steady state value. We employ the strategy of "iterate and check" where we check after each iteration if the solution is improved or not. The filter yields an optimum solution with a back projection error of 3.32, 30 times less compared to the optimisation using covariance weighted cost functions. The error is almost 10 times less for example in Figure 4.16. The back projection error at each iteration is given in Figure 4.18.

For a through analysis of our filtering scheme, we apply it to images from Oxford's data set [73]. We compare nonlinear optimisation of the reprojection error cost function and covariance weighted cost function with CI to the H_∞ filter where uncertainty is quantified using the feature uncertainty covariances from CI. Table 4.4 gives e_{bp} from all the techniques. We use three data sets and estimate the homography between two overlapping images. The data sets include Wedham College (Figure 4.19), Merton College (Figure 4.20) and University Library (Figure 4.21).

From tabulated results we can conclude that the H_∞ filter outperforms both the back projection error minimisation (e_{bp}^{RP}) and covariance weighted (e_{bp}^{CW}) minimisation for all the three data sets. The differences are substantial in almost all the cases. The covariance weighted optimisation technique performs better than the

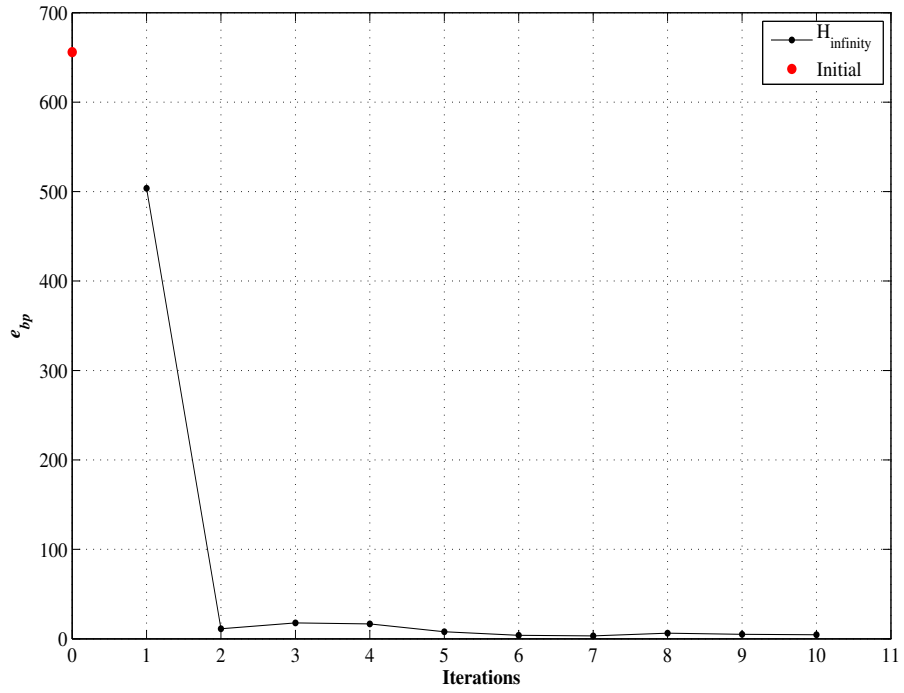


Figure 4.17: *Average e_{bp} for each estimation of $\mathcal{H}$ at every iteration for images in Figure 4.15.*

Data set	e_{bp}^{ini}	e_{bp}^{RP}	e_{bp}^{CW}	$e_{bp}^{H_\infty}$
Wedham College	14907	684.4	232.0	42.2
	385.4	542.89	505.51	213.8
	708.0	934.5	766.7	39.4
	30.1	780.3	731.7	20.5
Merton College I	26966.0	684.0	689.0	14.2
	644.3	716.0	716.0	12.3
University Library	10403	7635	654	37.49
	5755	571.9	335.7	57.3

Table 4.4: *Comparing H_∞ to non linear optimisation of two cost functions: re-projection error cost function and covariance weighted cost function. Corrupted initial estimates are fed to all three techniques.*

reprojection error optimisation in all but once instance (Merton College 1). The H_∞ filtering technique though outperforms all techniques for all examples.

Comparison of L_∞ with L_2 based Filter

Finally, we compare the H_∞ filter to the ubiquitous Kalman filter which minimises the L_2 norm. The Kalman filter is named after Rudolf E. Kalman one of its most

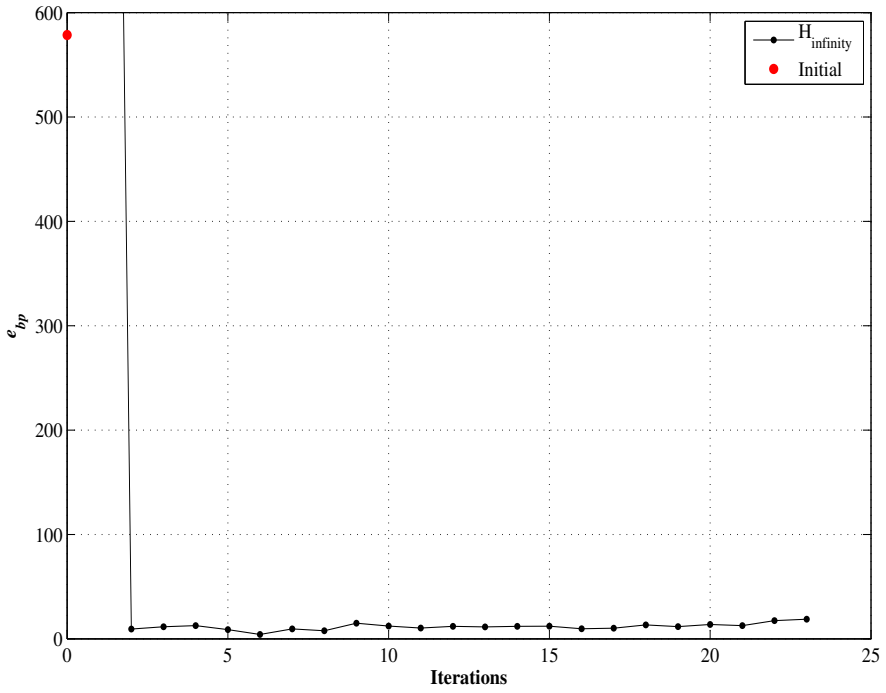


Figure 4.18: *Average e_{bp} for each estimation of $\mathcal{H}$ at every iteration for images in Figure 4.16.*

Figure 4.19: *Wedham College.*

famous inventors [96]. It is mostly applicable to linear systems with Gaussian process and observation noise therefore allowing an analytical solution to the Bayesian prediction and update step, although a version dealing with non-linear systems is also available [97].

Figure 4.20: *Merton College I*

Figure 4.21: *University Library*

As a recursive algorithm, the Kalman Filter is a set of equations that provides an efficient means to estimate the state of a process by minimising the mean of the squared error of these states. The Kalman filter equation are given below [82].

$$\hat{\mathbf{h}}_k = \hat{\mathbf{h}}_{k-1} + \mathcal{K}_k \left(b_k - C_k \hat{\mathbf{h}}_{k-1} \right) \tag{4.36}$$

$$\mathcal{K}_k = \frac{\mathcal{P}_{k-1} C_k}{\lambda + C_k' \mathcal{P}_{k-1} C_k} \tag{4.37}$$

$$\mathcal{P}_k = \left(I + \mathcal{K}_k b_k \right) \mathcal{P}_{k-1} \tag{4.38}$$

where $\mathcal{K}_k$ is the gain and $\mathcal{P}_k$ is the covariance matrix. The variable $\hat{h}_k$ contains the parameters of $\mathcal{H}$ and C and b are the system and measurement matrix, respectively. Equation (4.36) is the posteriori state estimation and is simply the latest prediction plus a weighting on the innovation which is $\mathcal{K}_k \left(b_k - C_k \hat{\mathbf{h}}_{k-1} \right)$. Equation (4.37)

is the Kalman gain and is chosen to minimise the squared error of the estimate. Equation 4.38 is the covariance matrix that defines the uncertainty in the system states.

Difference between the Kalman filter and H_∞ filter The Kalman filter assumes that the process has known dynamics and that inputs follow known statistical properties. These assumptions, however, do not hold for most of the state estimation problems [98]. In such a case, if the Kalman filter is applied then the optimal performance is not guaranteed. The L_∞ based filter, on the other hand, does not make any assumptions about the noise characteristics and instead of minimising the mean square error, as done in the Kalman filter, minimises the ∞-norm error. In such a way, it minimises the worst-case scenario. It has been shown that an ∞-norm based filter is less sensitive to parameter variations [98].

Figure 4.22 gives e_{bp} for each estimation of the homography at every iteration for the H_∞ filter with uncertainty and the Kalman filter without uncertainty for example in Figure 4.16. It is clear from the chart that the L_∞ norm filter performs better at each iteration and is gradually reducing the error. This shows the ability of the ∞-norm filter performs better at each iteration and is gradually reducing the error. For completeness, Figure 4.23 gives e_{bp} at each iteration with no uncertainty in the H_∞ filter. It can be seen from the figure that the ∞-norm filter is more stable compared to the Kalman filter and gives a lower overall error value.

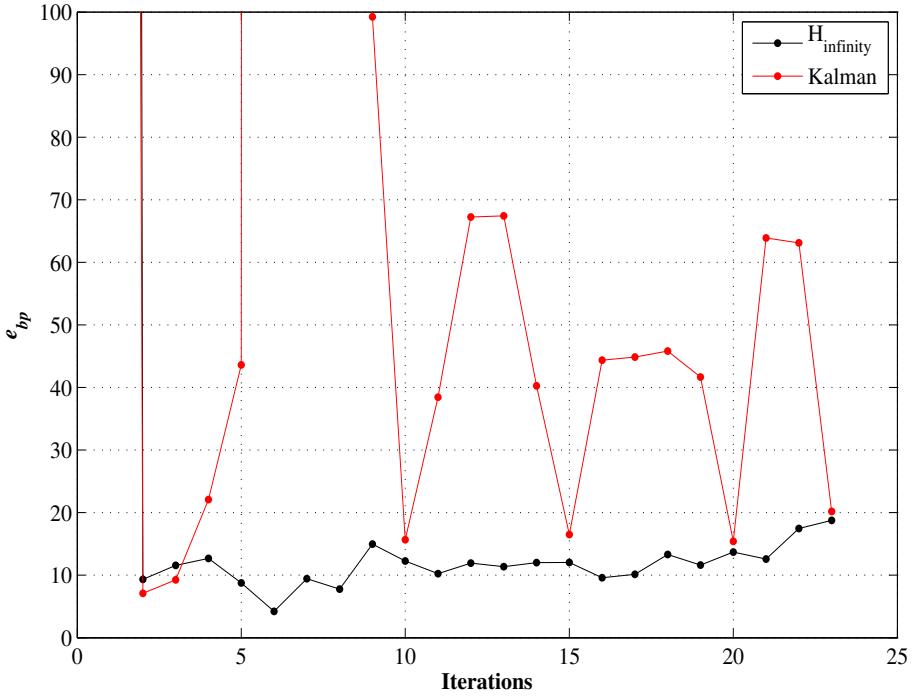


Figure 4.22: *Comparing e_{bp} at each iteration of H_∞ filter with uncertainty to Kalman filter.*

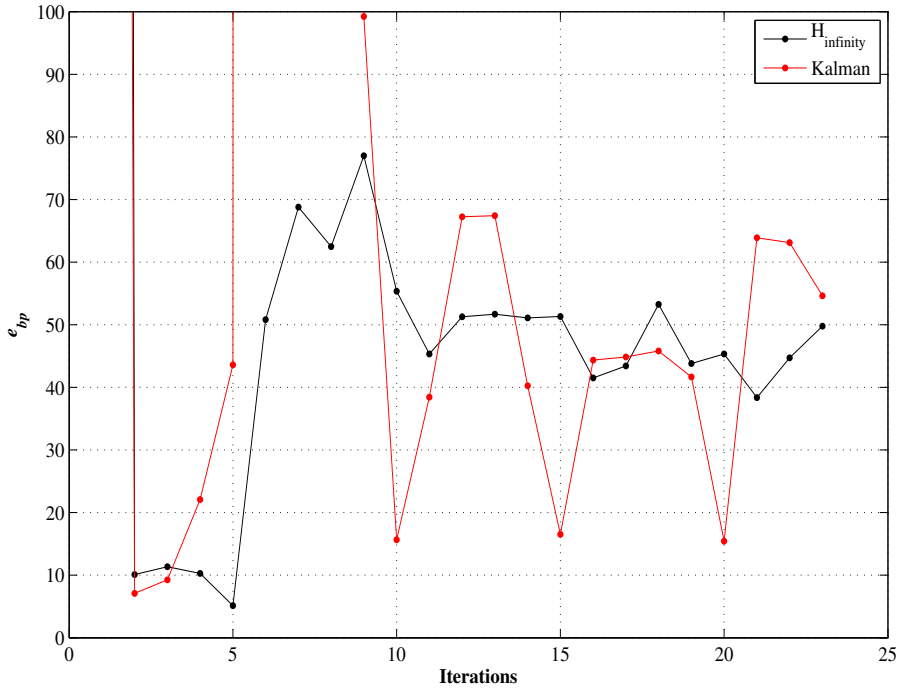


Figure 4.23: *Comparing e_{bp} at each iteration of H_∞ filter with no uncertainty to Kalman filter.*

4.4 Robust Coupled Filter

Up till now we have only computed geometric transformation $\mathcal{H}$ between overlapping images. Another transformation to be considered is the photometric transformation that often exists between images of the same scene. Two major sources of photometric difference are automatic camera adjustments, i.e., automatic gain control and illumination change, i.e., due to variation in lighting or relative motion between the camera. Figure 4.24 gives an example of two overlapping images with photometric differences caused by change in illumination and camera motion.

(a) (b)

Figure 4.24: *Photometric difference due to change in lighting and camera motion.*

In this section, we extend our L_∞ norm based filtering scheme to incorporate photometric registration. Therfore, in essence, we produce a filter that simultaneously estimates the geometric and photometric transformation between images.

4.4.1 The Photometric Model

We use a linear photometric model, since it is adequate for the types of deformations brought about by change in camera positions [1]. The model treats each of

the three colour channels independently. Within each channel, the variation between the two images is modelled as a linear transformation, having 2 parameters: a multiplicative term α and an additive term β [1].

The model is given as

$$\begin{pmatrix} r' \\ g' \\ b' \end{pmatrix} = \begin{bmatrix} \alpha_r & 0 & 0 \\ 0 & \alpha_g & 0 \\ 0 & 0 & \alpha_b \end{bmatrix} \begin{pmatrix} r \\ g \\ b \end{pmatrix} + \begin{pmatrix} \beta_r \\ \beta_g \\ \beta_b \end{pmatrix} \tag{4.39}$$

where a' relates to the second image. This model requires 6 parameters in total to be estimated which can be determined from regression. A minimum number of 6 equations is required and since it is a linear system, it can be broken up into 3 line-fit models. Accurately estimating the parameters requires that image features between images are matched without outliers.

Figure 4.25 gives an example where photometric transformation is applied to the image in Figure 4.25(a) from image in Figure 4.25(b). The result of this transformation is apparent in Figure 4.25(c), especially near the pathways. Figure 4.26 show the 2D line fit for each channel of the photometric model. As is obvious, the least squares technique tries to fit the best curve to the "available data". Furthermore, there are uncertainties in intensities between same pixels in different images that cannot be explained by the photometric model, i.e, modelling uncertainty.

4.4.2 H_∞ Simultaneous Geo-photometric Registration

To compensate for modelling uncertainty, we use the H_∞ filter developed previously and estimate the homographic and the photometric transformation between two images, simultaneously. The filter's inherent capabilities allow us to estimate parameters of differing models by coupling two or more systems together both with modelling and measurement uncertainty.

We include the linear model for estimating the photometric model with the one used for estimating $\mathcal{H}$ (Equation (4.5)). This is given as follows

(a) (b)

(c)

Figure 4.25: *Photometric transformation of left hand image to right hand image. (a) and (b) Original images. (c) Geo-photometric transformed images.*

$$\begin{bmatrix} x & y & 1 & 0 & 0 & 0 & -xx' & -yx' & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -x & -y & -1 & xy^i & yy' & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & r_1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & g_1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & b_1 & 1 \end{bmatrix} \begin{pmatrix} \mathbf{h} \\ \mathbf{g} \end{pmatrix} = \begin{pmatrix} x' \\ y' \\ r' \\ g' \\ b' \end{pmatrix} \tag{4.40}$$

where $\mathbf{h}$ contains the parameters of $\mathcal{H}$ and

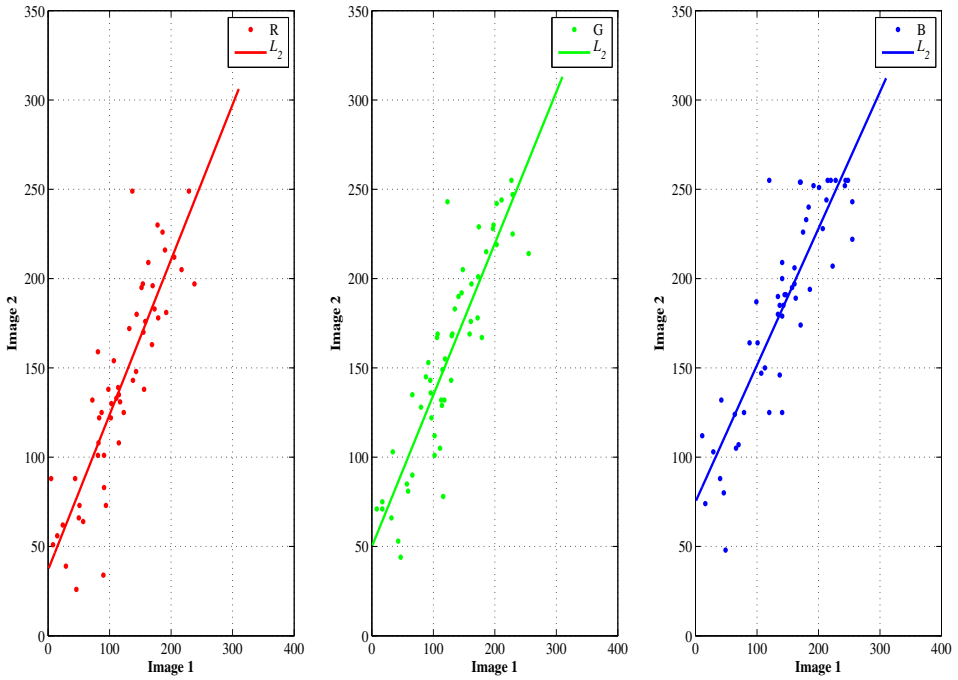


Figure 4.26: *Line fit to intensity data of all three channels for example given in Figure 4.25 using linear regression. There is almost a linear relationship as in the photometric model.*

$$\mathbf{g} = \begin{bmatrix} \alpha_r \\ \alpha_g \\ \alpha_b \\ \beta_r \\ \beta_g \\ \beta_b \end{bmatrix}. \tag{4.41}$$

The coupled system is clearly discernable. Adding uncertainty into the system leads to

$$\begin{bmatrix} \tilde{x} & \tilde{y} & 1 & 0 & 0 & 0 & -\tilde{x}\tilde{x}' & -\tilde{y}\tilde{x}' & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -\tilde{x} & -\tilde{y} & -1 & \tilde{x}\tilde{y}^i & \tilde{y}\tilde{y}' & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \hat{r} & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \hat{g} & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \hat{b} & 1 \end{bmatrix} \begin{pmatrix} \mathbf{h} \\ \mathbf{g} \end{pmatrix} = \begin{pmatrix} x' \\ y' \\ r' \\ g' \\ b' \end{pmatrix} \tag{4.42}$$

where $\tilde{m} = m + \Delta m$.

Using the same filtering equations as in Section 4.3.3, we can solve the above system for a robust $\mathbf{h}$ and $\mathbf{g}$.

Experimental analysis Here, we give results highlighting the capabilities of the coupled H_∞ filter to estimate geometric and photometric transformations. We will use geometric error as a metric for comparison between the regression technique and the coupled filter for estimation of the photometric model. We will additionally compare results of estimation of $\mathcal{H}$ from the H_∞ filter and LM using e_{bp}.

Figure 4.27 gives an example where geo-photometric registration is applied to two overlapping images. We show the mosaiced image before and after photometric transformation. The goodness of $\mathcal{H}$ is self evident from how good the mosaic is. Nonetheless, e_{bp} is given in Table 4.5 and proves again that the filter is more robust compared to LM. The effect of the photometric transformation is visible, more so near the hut where a smooth transition is visible, Figure 4.27(b). As a measure of comparison for photometric registration we use the offset error as given in Figure 4.28. This allows use to compare the filter based technique to the simple regression one. For the example given in Figure 4.27 the summed offset error from all three channels is 960.58 for the regression based technique and 498.39 for the H_∞ filtering technique. This amounts to a difference of almost 50%.

e_{bp}^{ini}	$e_{bp}^{H_\infty}$	e_{bp}^{RP}
380	6.36	318.72

Table 4.5: Comparing e_{bp} for H_∞ and non linear optimisation of reprojection cost function for estimation of $\mathcal{H}$ between images in Figure 4.27.

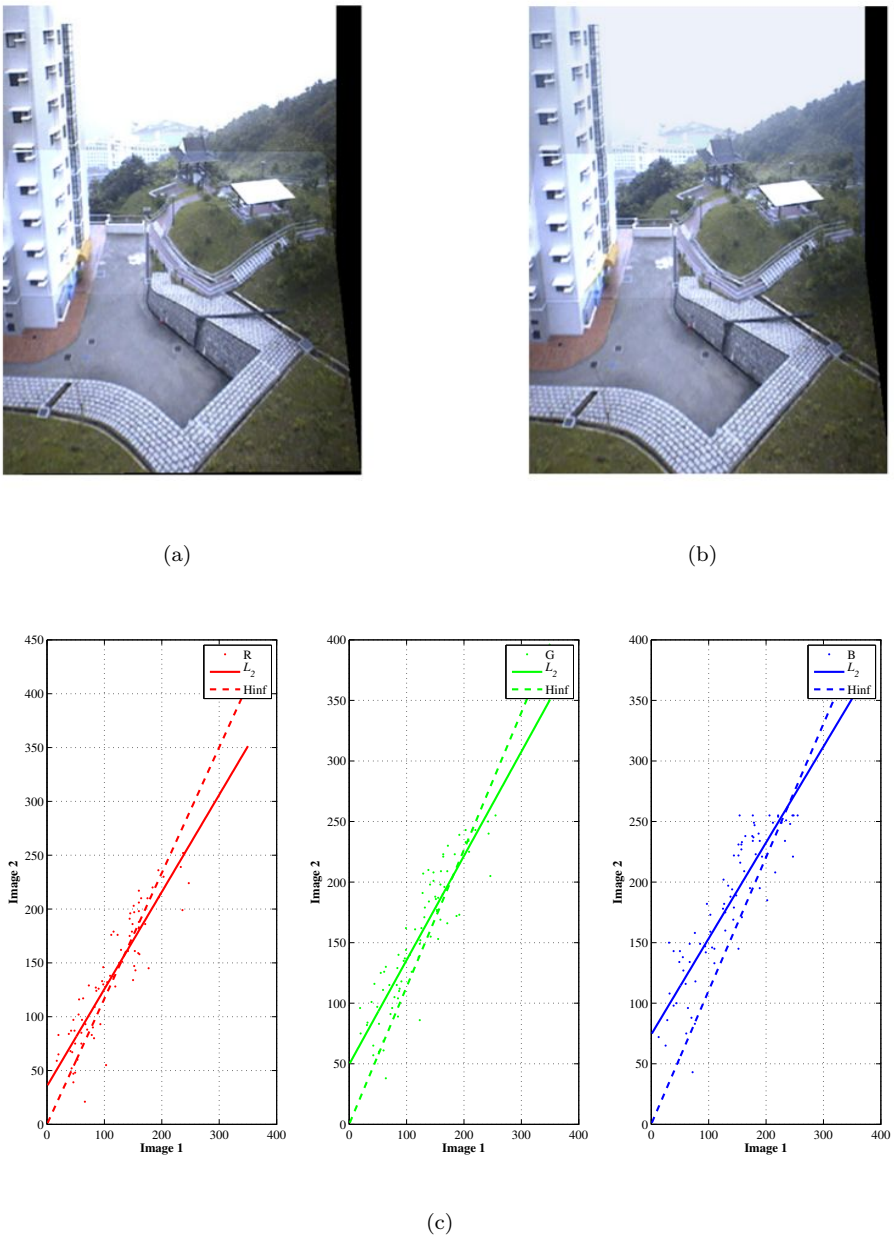


Figure 4.27: *Example 1 using the coupled filter. (a) Mosaic without photometric registration. (b) Mosaic with photometric registration. (c) Comparing line fits from linear regression and H_∞ filter.*

Figure 4.29 is another example where geo-photometric registration is applied to two overlapping images. Again, the effects of photometric transformation are evident. The colour of the rocky surface is more smooth after transformation as compared to before. In terms of the offset error, the filter gives a value of 434.38 whereas the regression baed technique gives an error of 484.38, a value which is 10%, Figure 4.29(c). Table 4.6 gives e_{bp} comparing H_∞ filter and LM optimisation. The filter

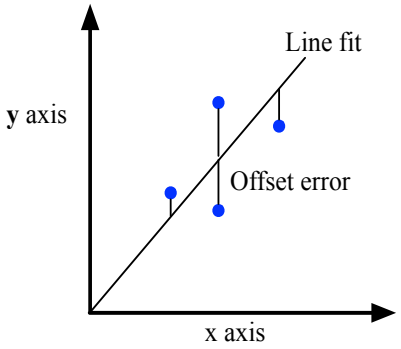


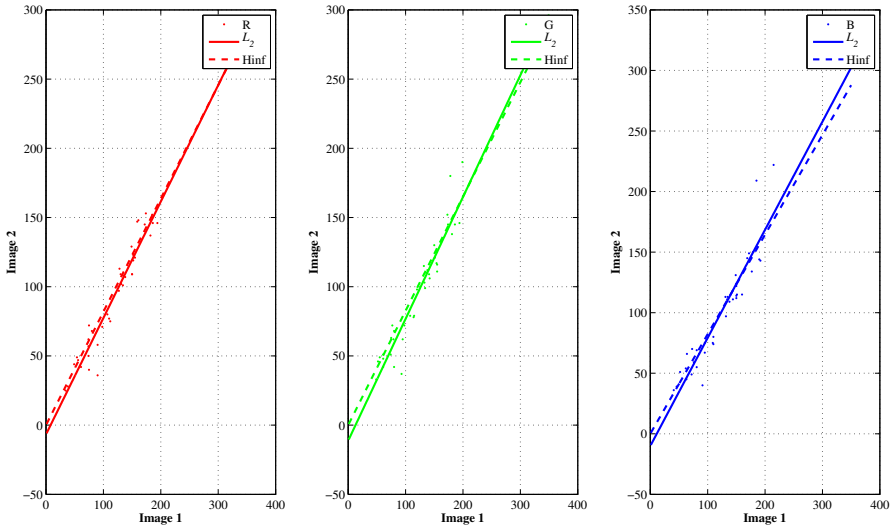
Figure 4.28: *The offset error. It is computed as the Euclidean distance from the point to the corresponding point on the line fit.*

outperforms the non-linear optimisation technique significantly, overcoming system uncertainty.

e_{bp}^{ini}	$e_{bp}^{H_\infty}$	e_{bp}^{RP}
351.85	1.97	353.65

Table 4.6: Comparing e_{bp} for H_∞ and non linear optimisation of reprojection cost function for estimation of $\mathcal{H}$ between images in Figure 4.29.

(a) (b)



(c)

Figure 4.29: *Example 2 using the coupled filter. (a) Mosaic without photometric registration. (b) Mosaic with photometric registration. (c) Comparing line fits from linear regression and H_∞ filter.*

4.5 Robust Time Varying $\mathcal{H}$ Estimation

In Section 4.2, we alluded to the point that the recursive least squares technique allows time varying parameter estimation. This ability enables the tracking of changing parameters as the conditions of the system change, i.e., the relative positions of the camera's change (especially in cooperative mosaicing scenarios).

Here, we apply our the H_∞ filter to track changing homographies in the presence of modelling and measurement uncertainties. To do this, we introduce a new recursive formulation.

4.5.1 New Recursive Formulation For Use in Filtering

Consider Equation (4.5). The system matrix C contains values of b, our measurement. Therefore we have a case where our system matrix is in someway a function of our measurement. This is not the ideal filter form and has to be fixed.

Consider Equation (2.5), by re-arranging and writing in full, we get

$$x^{'}\left(ux+vy+s\right)=r_1x+r_2y+t_x \tag{4.43}$$

$$y^{'}\left(ux+vy+s\right)=r_2x+r_3y+t_y \tag{4.44}$$

or

$$r_1x+r_2y-x^{'}ux-x^{'}vy-x^{'}s=-t_x \tag{4.45}$$

$$r_3x+r_4y-y^{'}ux-y^{'}vy-y^{'}s=-t_y \tag{4.46}$$

or in matrix form

$$\begin{bmatrix} x & y & \mathbf{O} & -xx^{'} & -yx^{'} & -x^{'} \\ \mathbf{O} & x & y & -xy^{'} & -yy^{'} & -y^{'} \end{bmatrix} \begin{bmatrix} r_1 \\ r_2 \\ r_3 \\ r_4 \\ u \\ v \\ s \end{bmatrix} = \begin{pmatrix} -t_x \\ -t_y \end{pmatrix} \tag{4.47}$$

where $\mathbf{O}$ is (0 0). The measurement equation now contains the translational parameters of $\mathcal{H}$ and we are left with estimating the remaining 7 parameters. Using this formulation, we have done away with the problem of the odd filter form. This formulation, however, requires that the translation parameters of the mapping be

known. Since we are considering no prior knowledge of the camera parameters and how the image acquisition device moves, these have to be determined before hand.

Estimating The Translations

Consider Figure 4.30 where a set of data points (set no.1) is transformed using a known projective homography into set no.2. It can be seen from the figure that set no.1 has been geometrically transformed by a translation, scale change and minor rotations. Now if we were to take a point in data set no.2 and subtract its location from the corresponding point in data set no.1, we will get the dominant translations. Using the difference method we get the determined translations as t_x = -0.32 and $t_y = 13.57$ (true translations are, $t_x = 10$ and $t_y = 30$). As expected, a significant disparity exists between the true translations and the evaluated translations. This is because the computed translations contain the effect of scaling and rotations. Therefore a method is required to remove these influences.

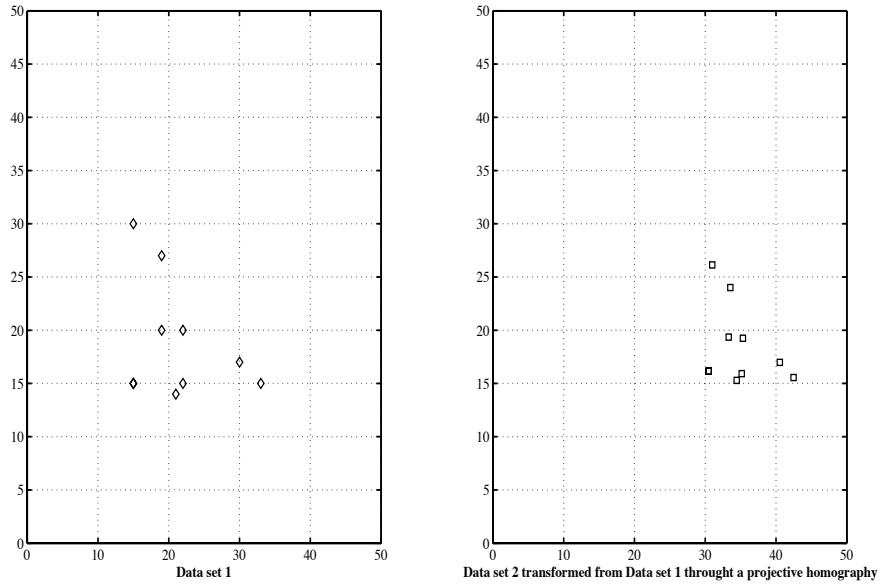


Figure 4.30: *A set of data points (left hand image) geometrically transformed using a known planar homography into data set no. 2 (right hand image).*

Incidentally, using the least squares solution to estimate $\mathcal{H}$ with these corrupted translations will yield the best possible answer that will depend on how far the values are from the true translations.

It is assumed that the effects of scale are the most influential. The discrepancies caused by the rotations are minimum since we are considering our image acquisition device will not be undergoing major rotation. We, need then, to eliminate the scaling effects which in essence requires us to know the scaling factor and then use the difference method to obtain the translations. One way of obtaining the characteristic scale is to use scale space representation as done in [57] and [62]. This procedure is however very involved and time consuming.

Another method which is simpler and much quicker, is to normalise the points. This method involves translating the points so that their centroid is at the origin, followed by an isotropic scaling [84].

Let $\mathbf{x}$ be a 2D point in homogenous coordinates. Using Equations (4.1) and (4.2) we can normalise the points. The translations evaluated using normalised points, when assembled in our new formulation to estimate $\mathcal{H}$ using least squares, yields a significantly better transformation compared to using unnormalised points. This is quantified by e_{bp}. Using normalised points the error is 1.6 and when using unnormalised points it is a significant 6 times more. This error is likely to increase when translations are in the order of magnitude of 2 and greater and the characteristic scale between the images is large. For example, for the same scale factor as in the synthetic data used in Figure 4.30 but with a 10 times bigger translations, the back projection error is 25 times more for the unnormalised case.

4.5.2 H_∞ Based Time Varying Estimation of $\mathcal{H}$

A truly appealing characteristic of using a filter is its ability to track time varying parameters. This is known as time varying parameter estimation. In the context of image mosaicing, if we have an imaging system which captures images for mosaicing at certain instances but not necessarily fixed instances or the acquisition system does not move a fixed amount each time, the mapping that will relate an image to another image will not be the same. Using linear estimation techniques as described in [24] and [1], an estimate of $\mathcal{H}$ can be obtained but the process can be computationally intensive.

We use the L_∞ norm based technique introduced previously and apply it to time varying $\mathcal{H}$ estimation using the equations given in Section 4.3.3. As previously done, we model Equation (4.47) with the uncertainty Δ.

Experimental analysis We start off by showing time varying $\mathcal{H}$ estimation using synthetic data.

Figure 4.31 shows a group of data points which are consequently mapped through variable $\mathcal{H}$, i.e., set no.1 is mapped to set no.2 using $\mathcal{H}_1$, set no.2 is mapped to set no.3 using $\mathcal{H}_2$ and so on. Every consecutive homography is different from the previous one as changes are incorporated into the translations, rotations and scale. So for every time step we try to estimate the varying $\mathcal{H}$ using data points which fall under this mapping. We would like to do this with as little computational effort as possible, using the least amount of measurements. We show how accurately each varying $\mathcal{H}$ is estimated by back projecting the points and visualising how they line up to the original data. It is seen that by using only one measurement we can track time varying parameters using the robust filtering method. Also shown is e_{bp} (bottom right figure) which is below 1 for each successive estimation of the mapping.

Figure 4.32 is another example with synthetic data, but with added system uncertainty of order of magnitude 2. It can be seen that the filter does well to cope with time-varying estimation of successive $\mathcal{H}$ in the presence of uncertainty.

Now we apply the filter to real world image data taken from moving cameras to estimate the underlying homographies. We do not display the feature matches for sake of clarity.

Figure 4.33 shows a sequence of images taken by a moving camera. The translational movement is obvious. We estimate the $\mathcal{H}$ between the consecutive images using our H_∞ recursive technique. We also add pixel error of order of magnitude 1 into our system matrix. It can be seen that the final mosaic is well aligned, even with system uncertainty. Figure 4.33(b) tracks the average back projection error at each iteration. From it, it can be seen how the filter manages to converge within 10 iterations and keep the error at a low value even in the presence of uncertainty.

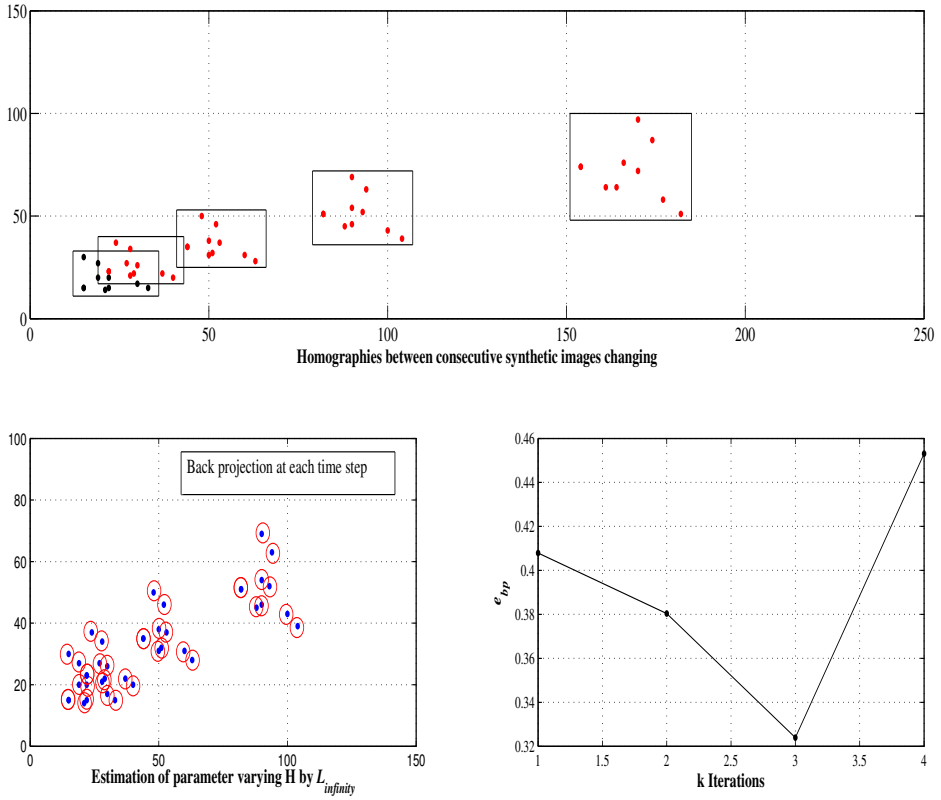


Figure 4.31: *Synthetic data used to highlight the effectiveness of our technique in tracking time varying parameters of $\mathcal{H}$. Top image shows a sequence of data points transformed using changing homographies, bottom left visualises the points back projected and bottom right reveals the average back projection error.*

Figure 4.34 gives result for the Wedham College data set [73]. As in the previous example, the filter does well to track the changing parameters between the second and third image, and after a brief divergence e_{bp} settles to a steady state value. When the fourth image is added the filter also copes well and maintains a low error value. In fact, it keeps reducing it, though only slightly, Figure 4.34(b). The final mosaic is well aligned, Figure 4.34(a).

Figure 4.35 gives results for BAE Systems data set. Again the filter does well to track changing $\mathcal{H}$, as seen in Figure 4.35(b). After an initial divergence, with more feature iterations, e_{bp} converges to a relatively low value. The final mosaic is well aligned.

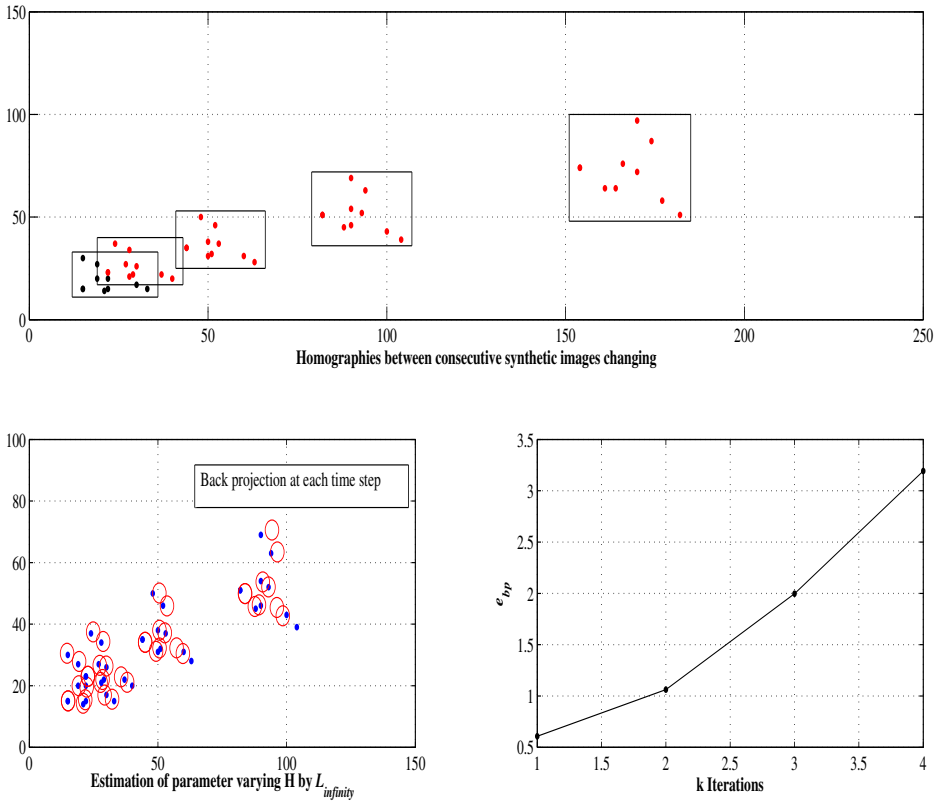
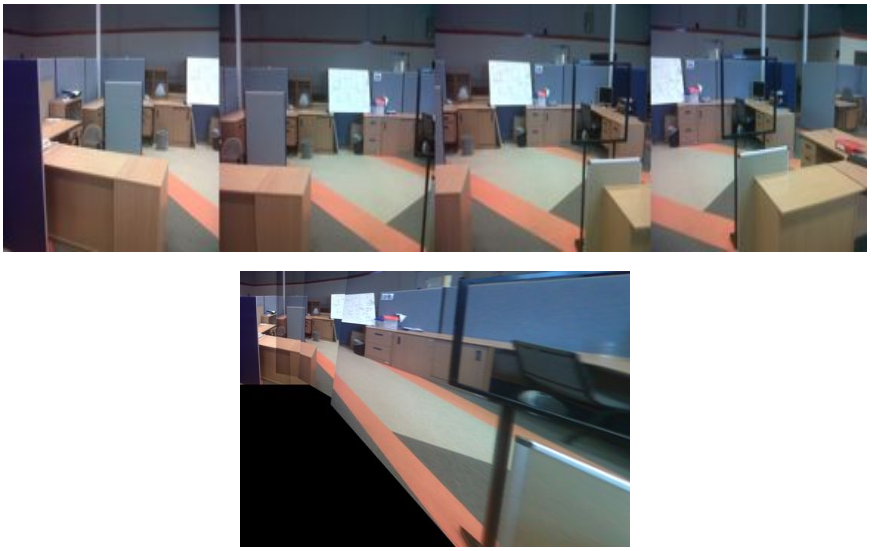


Figure 4.32: Synthetic data used to highlight the effectiveness of L_∞ technique in tracking time varying parameters of $\mathcal{H}$ with added system uncertainty. Top image shows a sequence of data points transformed using changing homographies, bottom left visualises the points back projected and bottom right reveals the average back projection error.

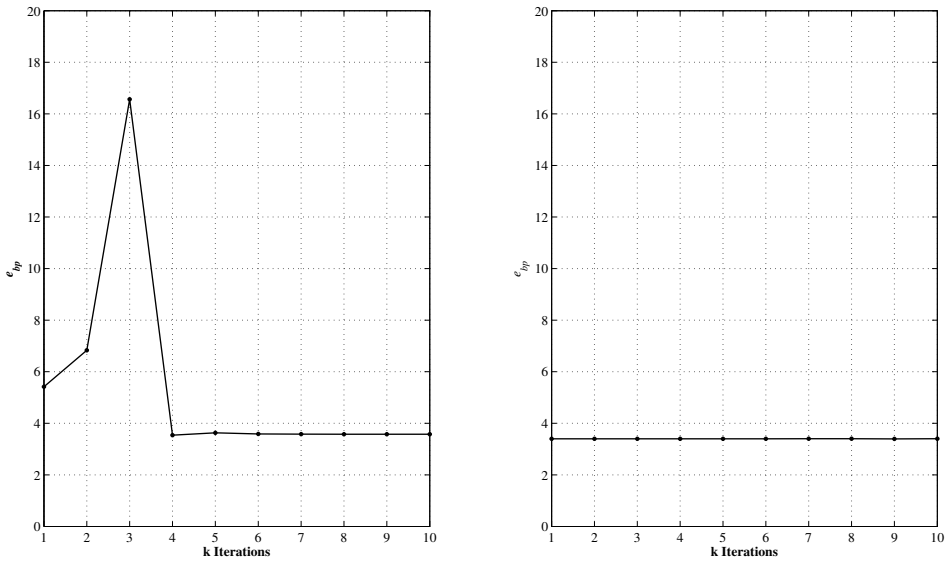
4.6 Bundle Adjustment vs. The Filters

The optimisation algorithm LM described in detail in Appendix A is used extensively in literature [24, 85] as standard. With a small number of parameters to be optimised, this implementation is suitable in terms of cost. Minimising a cost function with a large number of parameter, however, becomes computationally expensive. This is because the central step of LM, the solution of the normal equations, has complexity N^3 in the number of parameters and this step is a repetitive step [24].

There, however, exists a sparse block structure for the normal equations matrix that can be taken advantage of to reduce computational costs. We use the same

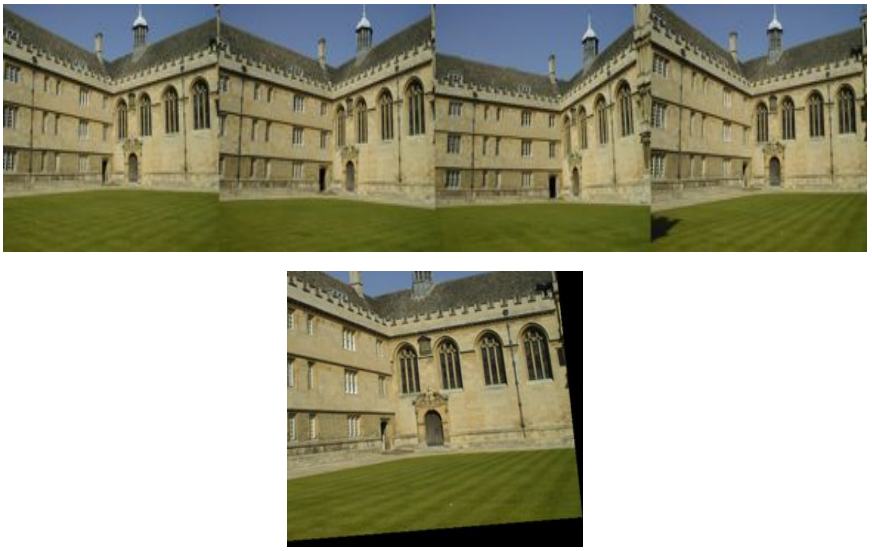


(a)

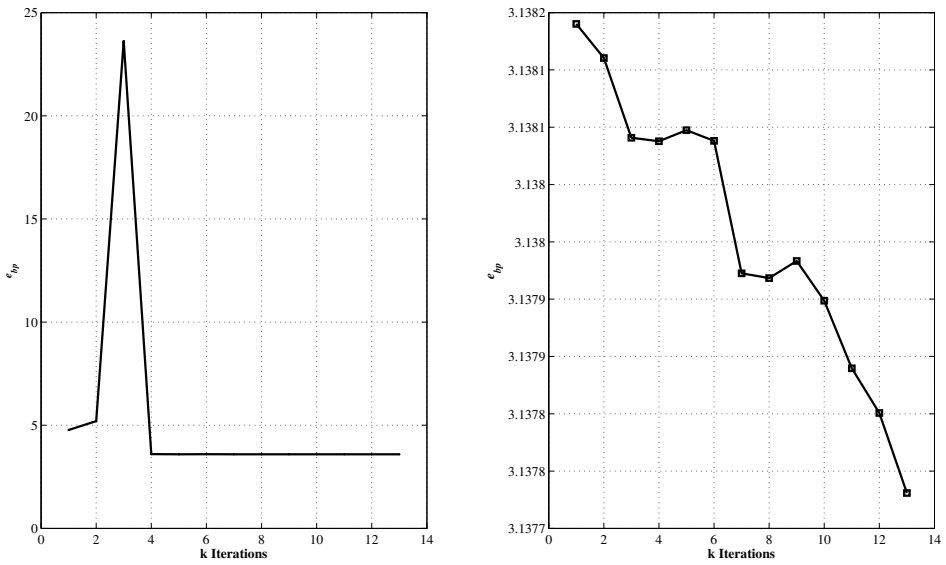


(b)

Figure 4.33: *Example 1 of time-varying* $\mathcal{H}$ estimation using the time-varying properties of H_∞ *filter. (a) Images from a moving camera and the consequent image mosaic. (b)* e_{bp} *between consecutive frames. Left side figure gives error between image 2 and 3, right side figure gives error between image 3 and 4.*



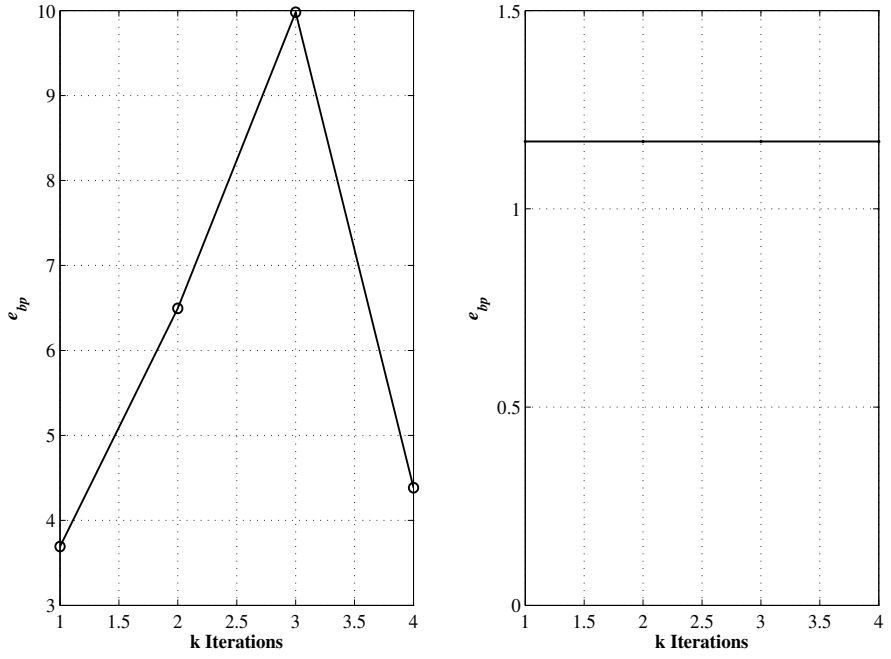
(a)



(b)

Figure 4.34: *Example 2 of time-varying $\mathcal{H}$ estimation using the time-varying properties of L_∞ filter on the Wedham College data set. (a) Images from a moving camera and the consequent image mosaic. (b) e_{bp} between consecutive frames. Left side figure gives error between image 2 and 3, right side figure gives error between image 3 and 4.*

(a)



(b)

Figure 4.35: *Example 3 of time-varying $\mathcal{H}$ estimation using the time-varying properties of L_∞ filter on BAE systems data set. (a) Images from a moving camera and the consequent image mosaic. (b) e_{bp} between consecutive frames. Left side figure gives error between image 2 and 3, right side figure gives error between image 3 and 4.*

idea to implement an optimisation technique to estimate globally consistent homographies between 2 or more views. Before this we, introduce the sparse LM algorithm.

Let ζ be a function that maps a parameter vector $\tau \epsilon \mathbb{R}^m$ to a measurement vector $\hat{\pi} = \zeta(\tau)$, $\hat{\pi} \epsilon \mathbb{R}^n$. Initial estimates for τ_0 and π are provided and the vector τ^* that best satisfies ζ locally, i.e., minimises the squared distance $\varepsilon^T\varepsilon$ with $\varepsilon = \pi - \hat{\pi}$ for all τ. A simple observation can lead to division of the parameter vector as $\pi = \left(\mathbf{c}^T, \mathbf{d}^T\right)$. Corresponding to this, the Jacobian matrix J has a block structure of the form $J = [A|B]$ where Jacobian submatrices are defined by

$$A = [\partial\hat{\pi}/\partial\mathbf{c}] \tag{4.48}$$

$$B = [\partial\hat{\pi}/\partial\mathbf{d}]\,. \tag{4.49}$$

The set of equations $J\delta = \varepsilon$ solved as the central step in LM takes the form

$$J\delta = [A|B] \begin{pmatrix} \delta_a \\ \delta_b \end{pmatrix} = \varepsilon. \tag{4.50}$$

If J obeys a sparseness condition then a computational advantage is to be had using the block structure. Suppose the measurement matrix π can be broken up into pieces as $\pi = (\pi_1, \pi_2, ..., \pi_n)$. Similarly, suppose that the parameter vector τ may be divided up as $\tau = (\tau_1, \tau_2, ..., \tau_n)$. The estimated value of π_i corresponding to a given assignment of parameters is denoted by $\hat{\pi}_i$. The sparseness assumption then is that, each $\hat{\pi}_i$ is dependent on $\mathbf{c}$ and $\mathbf{d}_i$ only, but not on the parameters $\mathbf{d}_j$. Corresponding to this division, J has a sparse block structure

$$\begin{bmatrix} A_1 & B_1 & & & \\ A_2 & & B_2 & & \\ \vdots & & & \ddots & \\ A_n & & & & B_n \end{bmatrix}. \tag{4.51}$$

In this form each step of the LM algorithm requires computation time linear in n. Without the spareness assumption the computation time can have complexity of order n^3 [24].

4.6.1 Application to $\mathcal{H}$ Estimation

We have previously applied LM for estimation of $\mathcal{H}$ between two images. Following the foregoing discussion, we can simultaneously estimate $\mathcal{H}$ and optimum position of each corresponding feature $\left(\mathbf{x}, \mathbf{x}'\right)$ with limited computational cost by minimising the reprojection error cost function, Equation (2.7).

We define a measurement vector $\pi_i = \left(\mathbf{x}^{\mathrm{T}}, \mathbf{x}'^{\mathrm{T}}\right)$. The parameter vector in this case is $\tau = \left(\mathbf{h}^{\mathrm{T}}, \left[\hat{\mathbf{x}}^{\mathrm{T}}, \hat{\mathbf{x}}'^{\mathrm{T}}\right]\right)$, where $\hat{\mathbf{x}}_i$ are estimated values of the feature points and $\mathbf{h}$ is a vector of parameters of $\mathcal{H}$.

The Jacobian matrices for this case take the special form

$$A = [\partial\hat{\pi}/\partial\mathbf{h}] \tag{4.52}$$

$$B = \left[\partial\hat{\pi}/\left[\partial\hat{\mathbf{x}}, \partial\hat{\mathbf{x}}'\right]\right]. \tag{4.53}$$

The form of the J for this case is best described visually as in Figure 4.36. Here, we are estimating the parameters of the homography between two images and the locations of corresponding features $\left(\mathbf{x}, \mathbf{x}'\right)$. This type of simultaneous estimation is called bundle adjustment (BA) [99, 100]. The sparse structure of the Jacobian matrix is obvious. This is because it is assumed that feature locations are independent of other features. This therefore leads to a desired reduction in computational effort.

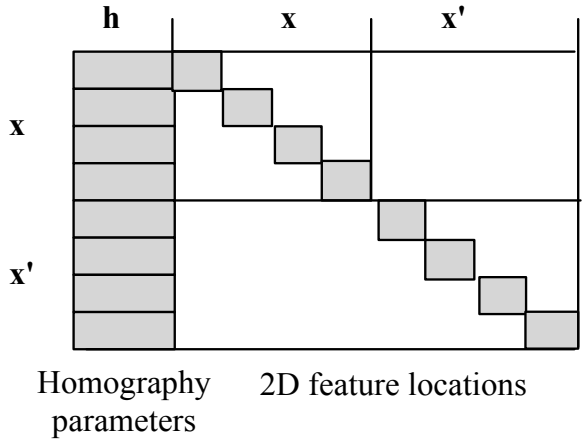


Figure 4.36: *Sparse structure of the Jacobian matrix estimating $\mathcal{H}$ between two cameras. The white spaces denote zeros and the grey blocks denote derivatives.*

4.6.2 Comparison to The Filters

We compare BA technique to the Kalman and H_∞ filter in term of speed and accuracy in the presence of noise. The filter equations are given in Section 4.3.3 and 4.3.4 and will not be reproduced here.

Experimental analysis We use the image data set given in Figure 4.37, which is taken by a rotating camera. We use e_{bp} to quantify performance.

Starting off with very good initial conditions, BA performs better than the two filters. A percentage decrease in e_{bp} of 50% on average is seen for the two consecutive $\mathcal{H}$ (Figure 4.37), whereas the filters shows an average increase in 200% in error. Now, with a slightly corrupted initial condition of order of magnitude -3, the filters perform much better than the BA. (It is important to note that the homography is very sensitive to noise). A percentage decrease in e_{bp} of almost 100% on average is seen. The percentage decrease in e_{bp} for the bundle adjustment is just 7%. Also, the H_∞ filter performs better than the Kalman filter by 25%. In terms of running costs, the bundle adjustment technique requires almost 30 times more time to converge compare to the filters. Values for e_{bp} are given in Table 4.7.

Figure 4.38 shows the structure of the Jacobian matrix for this example. The black spaces are indices with zero value gradients. We are estimating $\mathcal{H}$ for three

Figure 4.37: *Image set of an indoor environment. These images are used to compare BA and Kalman filter for estimation.*

		e_{bp}^{ini}	e_{bp}^{BA}	e_{bp}^{KF}	$e_{bp}^{L_\infty}$
No corruption	$\mathcal{H}_{12}$	1.66	0.99	5.83	4.10
	$\mathcal{H}_{23}$	1.38	0.53	4.03	5.11
With corruption	$\mathcal{H}_{12}$	1270.2	993.9	4.70	2.54
	$\mathcal{H}_{23}$	1428.0	977.6	4.10	3.83

Table 4.7: *e_{bp} values from comparison BA, Kalman filter and H_∞ filter on image data given in Figure 4.37. $\mathcal{H}_{12}$ refers to homography between Image 1 and 2 and $\mathcal{H}_{23}$ refers to homography between Image 2 and 3.*

consecutive images captured via a moving camera. The J consists of two $\mathcal{H}$s and points from three images.

Figure 4.39 is another example where the images are taken by a moving camera in an outdoor environment. Much of the same conclusions are drawn for this data set as previously. The BA technique performs well for good initial conditions. A percentage decrease in e_{bp} of almost a 100% on average is seen for the two consecutive $\mathcal{H}$, whereas the filters show an average increase of 80%. With a slightly corrupted initial estimate of order of magnitude -3, the filters perform better than BA. A percentage decrease of almost 100% on average between the three frames is seen for the filters, whereas for BA the decrease is less than 70%. Again, the H_∞ filter performs better than the Kalman filter giving a reduction in error of more

Figure 4.38: *Sparse form of the J for two $\mathcal{H}$s and image feature points. The black spaces indicate zeros and the white lines indicate derivates.*

Figure 4.39: *Image data set of outdoor environment taken using a moving camera. These images are used to compare BA and Kalman filter for estimation.*

than 25% compared to it. Values are given in Table 4.8. Furthermore, the time taken by BA for estimation is more than 70 times more compared to the filters.

		e_{bp}^{ini}	e_{bp}^{BA}	e_{bp}^{KF}	$e_{bp}^{L_\infty}$
No corruption	$\mathcal{H}_{12}$	0.89	0.01	1.71	1.59
	$\mathcal{H}_{23}$	0.55	0.01	1.22	0.56
With corruption	$\mathcal{H}_{12}$	434.57	151.85	1.71	1.32
	$\mathcal{H}_{23}$	471.65	61.41	13.04	9.88

Table 4.8: *e_{bp} values from comparison of BA, Kalman filter and H_∞ filter on image data given in Figure 4.39. $\mathcal{H}_{12}$ refers to homography between Image 1 and 2 and $\mathcal{H}_{23}$ refers to homography between Image 2 and 3.*

4.7 Chapter Conclusion

In this chapter, robust techniques to estimate 2D homography $\mathcal{H}$ are introduced. We outlined a RLS technique for updating $\mathcal{H}$ for a moving platform that is also capable of tracking changing parameters of the homography.

A novel technique that uses information from all three channels of an RGB image to reduce feature location uncertainty is developed by applying covariance intersection. It is shown that by reducing uncertainty better estimates of $\mathcal{H}$ are obtained. We have applied a novel H_∞ filtering scheme that takes into account system modelling uncertainties to homography estimation and have shown it to be very successful. This is done in context of simultaneous estimation of geometric and photometric transformation of images and also time varying estimation of $\mathcal{H}$.

We have applied a sparse Levenberg-Marquardt optimisation algorithm for estimation using the sparseness assumption and showed that it gives good results for global optimisation of parameters of $\mathcal{H}$ and feature locations, but fails in the presence of error. This technique is compared to the H_∞ and Kalman filter approach, which outperform it by a significant margin when encountering errors and are also computationally less costly.

5
Rapid 3D Reconstruction

In this chapter, we introduce techniques for swift 3D scene reconstruction. We begin by introducing a novel image feature matching technique based in the 3D space. This involves knowing the position of the camera's with respect to each other and determining the viewing overlap between them. Then by projection into the image plane of each camera, we can have relative search regions in which to look for feature matches between images. This has the advantage of reducing matching time and increasing precision, since the chance of false matching is reduced. We include this methodology in this section and not in Chapter 3 which deals with features matches, since it is set in the 3D domain. This is the explicit topic of this chapter.

Then, we introduce a 3D scene reconstruction algorithm that uses homographic line matching to solve the correspondence problem between images. We use the lines to do a semi-dense reconstruction and compare it to a full reconstruction of the scene in terms of time and visualisation.

Recently, applications of 3D scene recovery from two or more images have become prevalent. From model building to navigation their uses extend to a variety of domains [53, 101–109].

Due to advances in computational power and programming tools, scene reconstruction on a large scale is presently being attempted. One such algorithm recovering the scene in 3-space from a large collection of data is given in [101]. It is capable of scene reconstruction from a large collection of images in less than a day using parallel running modules. In [110], skeletal graphs are used to reduce the large collection of images only to ones which sufficiently represent a scene and therefore

reduce processing times. They show a relative increase in efficiency by an order of magnitude of one. A direct method based on spatio-temporal image gradients for ego-motion and depth recovery of a scene is given in [103].

In [111], a fast real time 3D reconstruction algorithm from video sequences is proposed combined with an ego-motion module. It uses an efficient large-scale stereo matching technique given in [112] that builds priors over the disparity sparse, therefore reducing matching ambiguities. Model reconstruction from video streams captured on a compact camera is shown in [113]. Here, Harr wavelet responses are used to generate an edge map for use in geometry estimation. A review of 3D reconstruction from video sequences is given in [114].

Underwater scene reconstruction is tackled in [115]. Here, a wide base-line stereo rig attachable to an autonomous underwater vehicle is used to compute a sparce reconstruction of underwater structures. The reconstruction can be used interactively to plot maps for further missions. Additionally, interactive 3D reconstruction tools are given in [116, 117], though require significant input from the user.

Photo tourism is a relatively new application of 3D scene reconstruction. It allows exploring photo collections in 3-space as done in [118] for data sets downloaded from Flickr and in [108] where an accuracy assessment is carried out on an online point cloud generator from uploaded images. Similar technology is also presented in [119]. Here, a collection of unstructured 2D images from a search can be registered, reconstructed and shown to the end user in an interactive interface. The interface also makes it easy to construct photo tours of scenic or historic locations. Although effective, the structure from motion module slows as the number of images increases.

Stereo based reconstruction is used for surveying structures in [120] to assess damage and maintenance schedules. Robust stereo matching using cost aggregation is given in [121] where pixels outside segmented areas of the image are treated as outliers. The technique shows good results compared to global stereo correspondence algorithms. A discussion on global versus local methods is given in [55]. A quasi-dense correspondence algorithm for use in surface reconstruction from uncalibrated images is given in [122]. Here, standard feature matches between two

images are used as seed points to look for more potential matches. They show good results for facial reconstruction.

Use of stereo algorithms for modelling urban scenes using meshes and geometric primitives is given in [123]. Making use of common characteristics of man made structures, namely piecewise planar structures, they propose a sampling technique to divide the mesh into primitive and mesh components. Application of this to complex structures shows good results in decent time. Piecewise planar models of scenes are built in [124, 125] and are well suited to modelling urban environments. Similar to this, [126] uses the "Manhattan world" assumption that all surfaces are aligned with three dominant axis. This greatly simplifies the complexities of the representation. Again based on urban scene assumptions, [127] constructs building interiors using images to a good degree. Aerial reconstruction is attempted in [128] using GPU based dynamic programming. Here too, structural assumptions of a city environment are made for urban scene modelling, and is done in good time.

Urban scene representation using meshes can also be done using laser scanners, which although effective, are costly [129, 130]. Environmental modelling combining lasers scans and video streams is achieved in [131] where texture is added to the planes from image data. Something similar is done in [132] where 2D vertical scans of city facades and images gathered from a moving car in normal traffic conditions are used. The large amounts of data is divided into psuedo-linear segments for easy handling, which are then processed individually.

Accurate 3D textured models using a monocular camera are given in [133] and [134]. In the latter, an automatic urban landscape reconstruction tool is built which uses textured planar surfaces to model the structures. Textured surface reconstruction of non-rigid shapes is given in [135] from a monocular camera. Using a learning algorithm, which relies on intensity patterns and local shapes, low textured objects are reconstructed.

Shape retrieval is also done by approximating the visual hull of the observed structure. The visual hull is an outer approximation computed as the intersection of visual cones from all image silhouettes [136, 137]. A combination of reconstruction from shape from silhouettes and stereo is proposed in [138], where the reconstruction problem is cast into a convex variational problem with consistency constraints.

Dynamic scene reconstruction by segmenting background and foregrounds is done in [139]. Using a novel matting technique foregrounds are separated from multiple views and then projected into the 3D space. The problem of image synchronisation for dynamic object reconstruction using visual hulls is studied in [140]. Here, the requirements for such a system are put forward based on test cases.

5.1 IMU Guided Feature Matching

We have included the guided feature matching technique here as it is based in the 3D space. The basic premise behind the proposed technique is that if relative position and orientation between two or more camera's is known plus their intrinsic calibration parameters, we can determine the viewing overlap of the cameras. This means we can isolate the common area viewed by the images taken by each camera.

Knowing this information can help reduce the search region when looking for feature matches between the images, leading to increase in matching accuracy and efficiency. We propose using an inertial measurement unit (IMU) based technique to determine the relative positions and orientations of the cameras.

Below we give a brief introduction to IMUs followed by a description of the algorithm and experimental analysis with real data.

5.1.1 Introduction to IMUs

Inertial navigation is a self-contained navigation technique that uses measurements from accelerometers and gyroscopes to track position and orientation of an object relative to a known starting point. They typically contain three orthogonal rate gyroscopes and three orthogonal accelerometers measuring angular rates and linear accelerations, respectively. By processing outputs from these sensors it is possible to track the position and orientation of a device with reasonable accuracy.

IMUs generally fall into one of two categories, i.e., stable platform systems or strapdown systems. The difference between the two categories is the frame of reference in which the gyroscopes and accelerometers operate. Briefly, in stable

platform systems the sensors are mounted on a platform isolated from any external rotation keeping it aligned with the global frame. Figure 5.1 gives a stable platform IMU unit. Note the gimbals in the illustration which allow the platform freedom in all three axes. Please refer to [141] for an in-depth explanation of the how stable platform systems work.

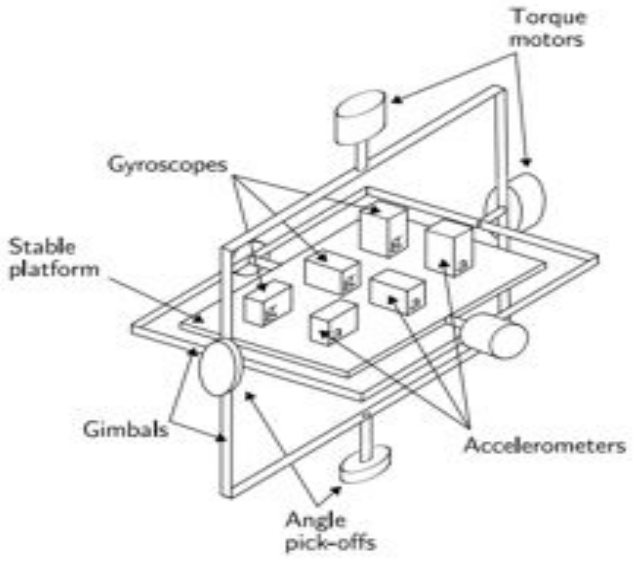


Figure 5.1: *A stable platform IMU.*

In a strapdown system, the inertial sensors are mounted onto the device (Figure 5.2, so outputting values are measured in the body frame instead of the global frame. The orientation is determined by integrating the signal from the rate gyroscopes. To track position, the three accelerometers are resolved into global coordinates using orientations determined from the gyroscopes. The global acceleration signals are then also integrated. The procedure is shown in Figure 5.3

5.1.2 Guided Feature Matching - Methodology

From Chapter 2, we know that a projective transformation from world coordinates to image coordinates is given by

$$\mathbf{x} = KR\left[I \mid -\tilde{\mathbf{C}}\right]\mathbf{X}, \tag{5.1}$$

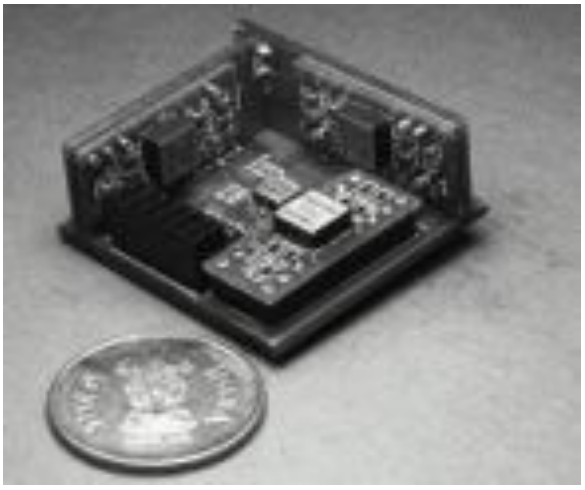


Figure 5.2: *An example of a strapdown IMU unit*

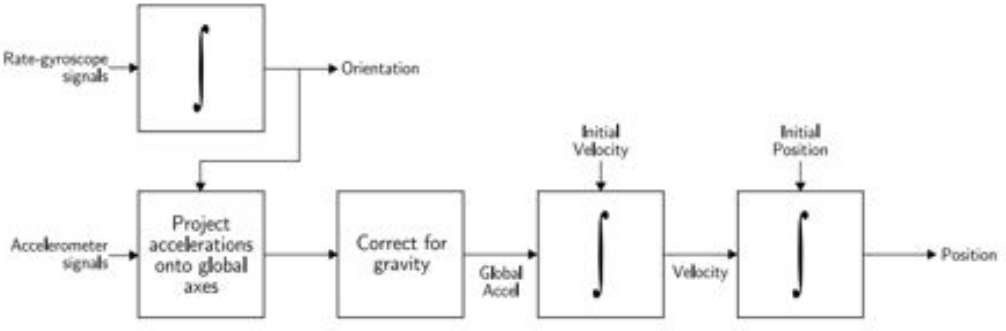


Figure 5.3: *Strapdown inertial navigation algorithm*

where K is the matrix of internal parameters, R and $\tilde{\mathbf{C}}$ are the rotation matrix and camera centre in world coordinates, respectively. An inverse of this transformation to 3-space is a point to line mapping. We therefore require triangulation of a corresponding feature to exactly locate it in the 3D world, Figure 5.4.

We can use a similar method to identify the field of view of a camera. From Equation (2.3) we can define the X and Y locations in 3-space as

$$\begin{aligned} X &= -x_0 + Zx/f \\ Y &= -y_0 + Zy/f, \end{aligned} \tag{5.2}$$

where x_0 and y_0 are the principal point offsets, f is the camera focal length and x and y are pixel coordinates. Note that we are assuming that the world coordinate frame aligns with the camera coordinate frame.

By knowing our camera parameters we know the size of the image in terms of pixels that will be produced. Using maximum and minimum of the locations of the pixels, i.e., its periphery, we can estimate roughly the field of view for a given depth Z. Figure 5.4 gives an example where a field of view (FOV) is created in 3-space using the above expression for a certain depth.

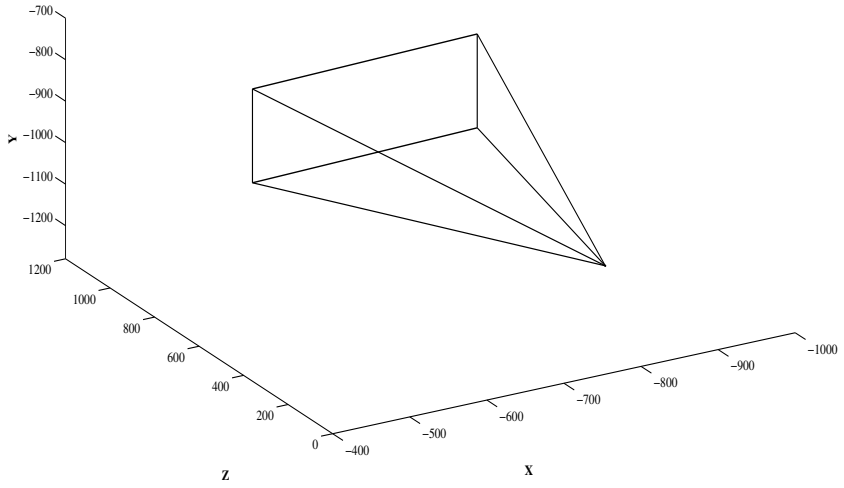


Figure 5.4: *A cameras FOV in 3-space at a certain depth Z.*

Figure 5.5 shows the variation in the FOV as Z is varied.

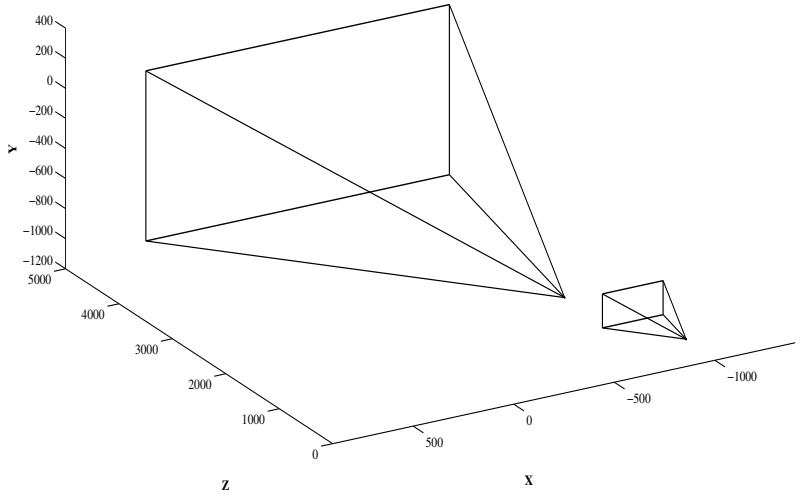


Figure 5.5: *Variation in the FOV as depth is varied.*

Illustrative example With the help of an example, we explain how to use FOVs in 3-space to match image feature between overlapping views.

Consider Figure 5.6 where three overlapping images are taken by a rotating camera with a pan of ± 30 degrees. Images are taken by a uEye camera with an f of -830.43 mm and x_0 and y_0 of 674.25 mm and 508.20 mm, respectively [53]. Using this information and the camera parameters, we construct FOVs for all the three images in 3-space as described in the previous section. This is illustrated in Figure 5.7.

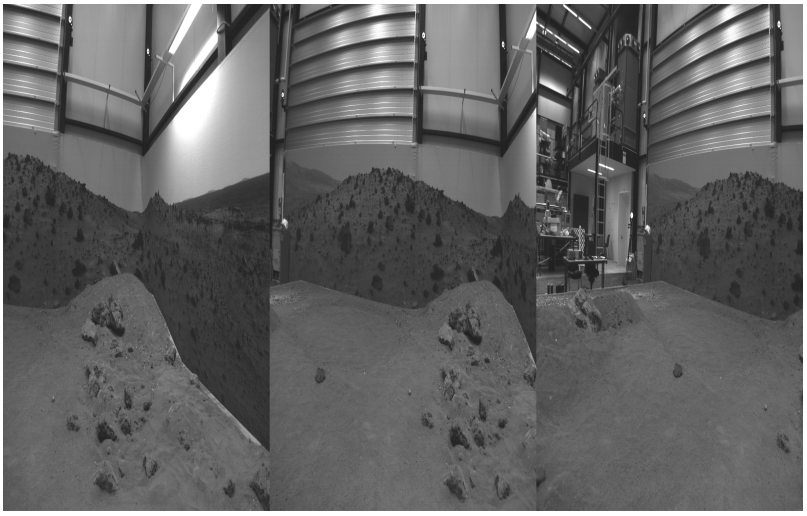


Figure 5.6: *Set of three images taken from a uEye camera with a pan of ± 30 degrees.*

We then locate the various points of intersections where the FOVs overlap each other using Equation (5.3) below.

$$(\mathrm{I}_x, \mathrm{I}_y) = \left(\frac{(x_1y_2 - y_1x_2)(x_3 - x_4) - (x_1 - x_2)(x_3y_4 - y_3x_4)}{(x_1 - x_2)(y_3 - y_4) - (y_1 - y_2)(x_3 - x_4)}, \right. \\ \left. \frac{(x_1y_2 - y_1x_2)(y_3 - y_4) - (y_1 - y_2)(x_3y_4 - y_3x_4)}{(x_1 - x_2)(y_3 - y_4) - (y_1 - y_2)(x_3 - x_4)} \right) \tag{5.3}$$

This expression finds the point of intersection $(\mathrm{I}_x, \mathrm{I}_y)$ between a line segment l_a given by points (x_1, y_1) and (x_2, y_2) and a segment l_b given by (x_3, y_3) and (x_4, y_4). Note that this solution gives the point of intersection for infinitely long lines but

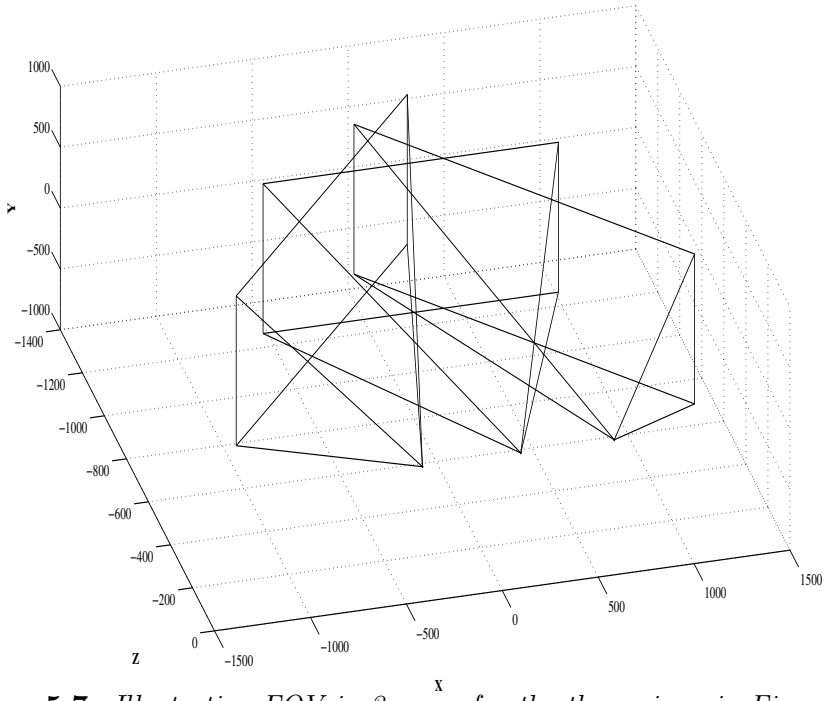


Figure 5.7: *Illustrating FOV in 3-space for the three views in Figure 5.6.*

only two points for each line segments are required. In our implementation, the line segments start from the principal points x_0 and y_0 are extended till Z. This solution will work for any values of Z.

This way we can identify an area of commonality between the views by connecting the point of intersections with a circular region as in Figure 5.8. We then project this region of overlap back into the image coordinates using Equation 5.1 and extract image features for just this section of the image, Figure 5.9. This helps in reducing computational costs of extracting and then consequently matching features. Furthermore, it also helps in greatly reducing the potential for false matches.

Using the projected areas of overlap, we can process just this area of the image and extract features over them, as shown in Figure 5.10 for the left and central image. It is obvious that the technique does well in isolating the areas of overlap between the images. For completeness, Figure 5.11 shows features extracted over all the image for the left and central image. A number of useless features are present that cannot be matched between the images.

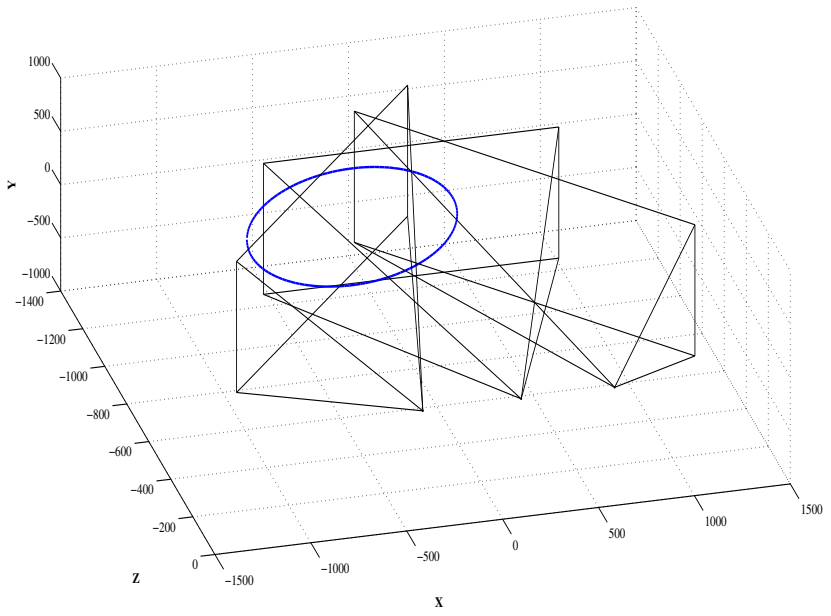


Figure 5.8: *Illustrating area of commonality between the views in 3-space. The*

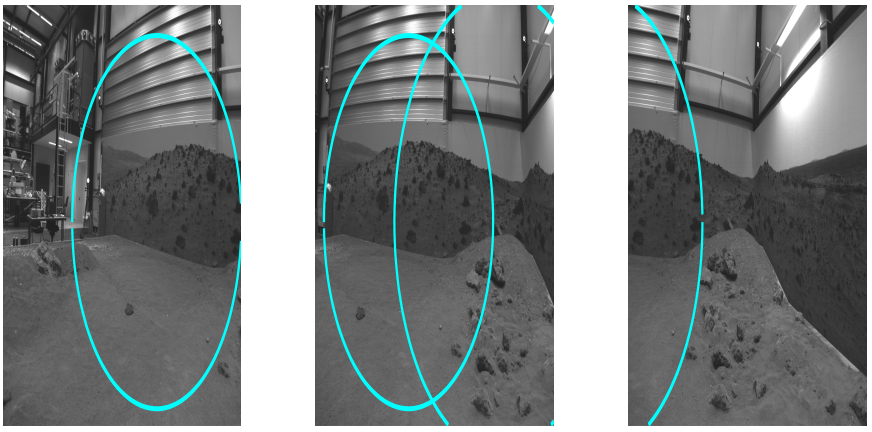


Figure 5.9: *Areas common to the different views is projected back into the image coordinates.*

Figure 5.12 and 5.13 give the time taken for extraction and matching with and without IMU assisted feature matching. Assisted IMU feature extraction is 4% faster compared to non IMU extraction. In terms of feature matching, assisted IMU is more than 55% faster compared to non IMU based feature matching.

Knowing the relative positions of the camera and its intrinsic parameters we can perform IMU guided feature matching. The computational complexities of determining the regions of overlap are simple only requiring line intersection, as shown

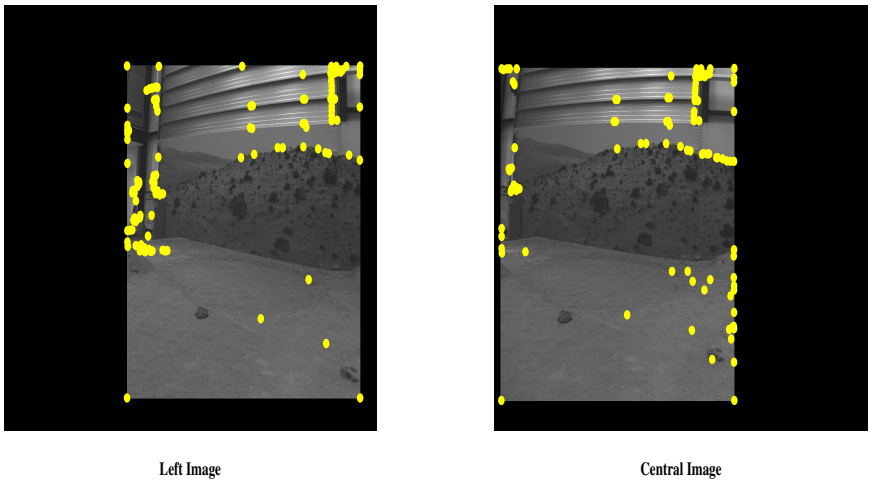


Figure 5.10: *Harris features extracted and matched only in the area of overlap for the left image and the central image.*

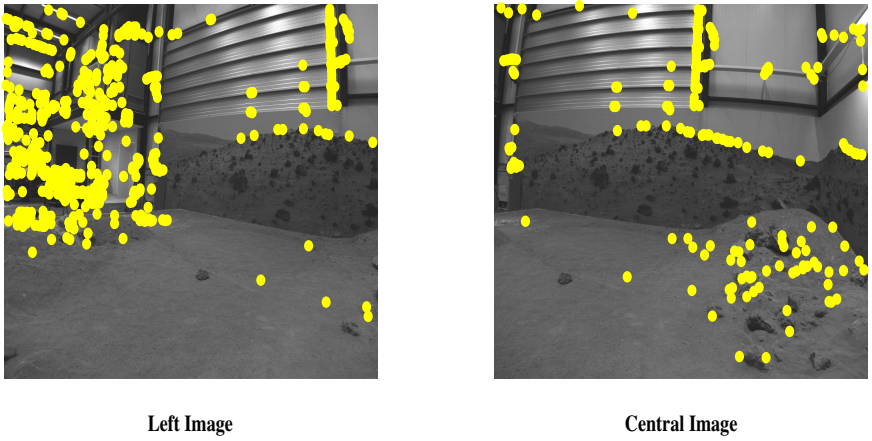


Figure 5.11: *Harris features extracted and matched over all the image area.*

in Equation (5.3) and are easy to implement. Furthermore, since most, if not all, unmanned vehicles are fitted with IMU units, this technique is a viable alternative solution for fast feature matching.

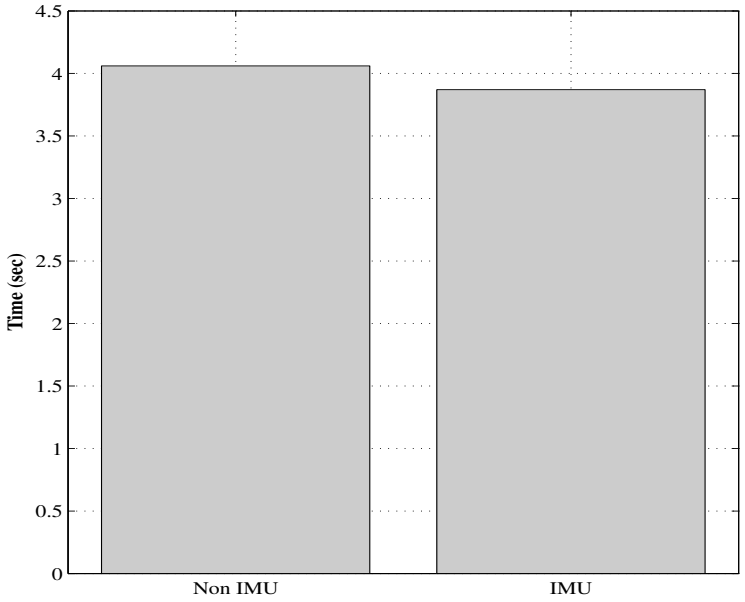


Figure 5.12: *Comparing time taken to "extract" Harris features from the entire image area and isolated image area using IMU assistance.*

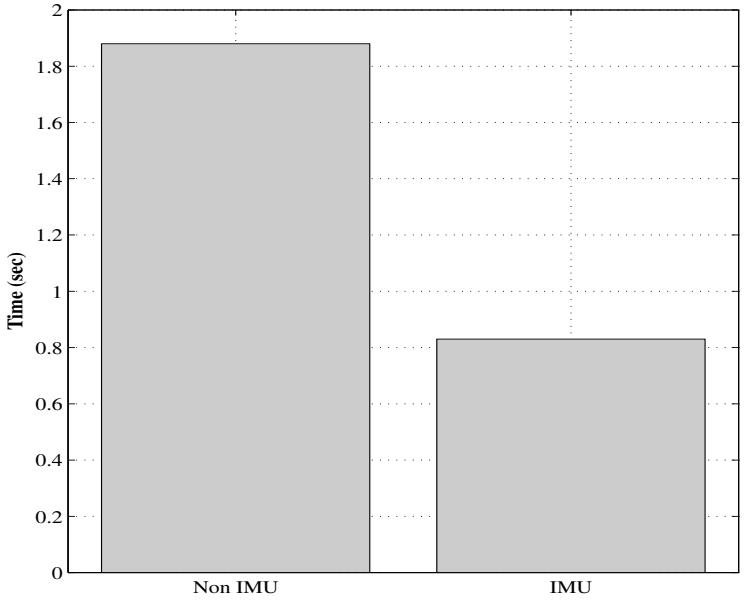


Figure 5.13: *Comparing time taken to "match" Harris features from the entire image area and isolated image area using IMU assistance.*

5.2 Semi Dense 3D Scene Reconstruction

3D scene reconstruction from images is an important topic in computer vision. It serves many applications which include vehicular localisation and navigation and

scene modelling for use in urban planning and virtual reality [53, 101, 102].

In this section, we introduce a semi-dense projective scene reconstruction method from two images without the need for image rectification. Additionally, we propose a novel homographic line based pixel matching technique that allows for fast correlation of pixels, leading to dense reconstruction. Next, we outline the methodology for 3D reconstruction from images followed by our algorithm with experimental results.

5.2.1 Outline of 3D Reconstruction

We introduced the Fundamental matrix in Chapter 2 that encapsulates the geometry of a scene visible to two images. A method to compute it and to extract the camera matrices from it is also given. Here, we include a general method for scene reconstruction, known as triangulation, given two camera matrices P and $\mathrm{P}^{'}$.

Given P and $\mathrm{P}^{'}$, let $\mathbf{x}$ be a feature in one image and $\mathbf{x}^{'}$ a feature in another image of a 3D point $\mathbf{X}$ satisfying the epipolar constraint $\mathbf{x}^{'\mathrm{T}}\mathrm{F}\mathbf{x} = 0$. Geometrically, this constraint can be interpreted as rays in space corresponding to the features. In particular, this means that the point $\mathbf{x}^{'}$ lies on the epipolar line $\mathrm{F}\mathbf{x}$ (Chapter 2). This in turn means that the two rays back-projected from points $\mathbf{x}$ and $\mathbf{x}^{'}$ line in a common epipolar plane. This plane passes through the two camera centres [24]. Since the two points lie in a plane, they will intersect in a point $\mathbf{X}$ in 3D space. This point $\mathbf{X}$ projects through the two cameras P and $\mathrm{P}^{'}$ to points $\mathbf{x}$ and $\mathbf{x}^{'}$ on the image plane, respectively. This is shown in Figure 5.14 and given as

$$\mathbf{x} = \mathrm{P}\mathbf{X}, \quad \mathbf{x}^{'} = \mathrm{P}^{'}\mathbf{X}. \tag{5.4}$$

Knowing the camera matrices and corresponding features $\left(\mathbf{x}, \mathbf{x}^{'}\right)$ we can retrieve the 3D point $\mathbf{X}$ from linear triangulation method. Equation (5.4) can be manipulated into the form $C\mathbf{X} = 0$, an equation linear in $\mathbf{X}$. Eliminating the homogenous scale factor, we get two linearly independent equations for each corresponding image feature. For example, for the first image

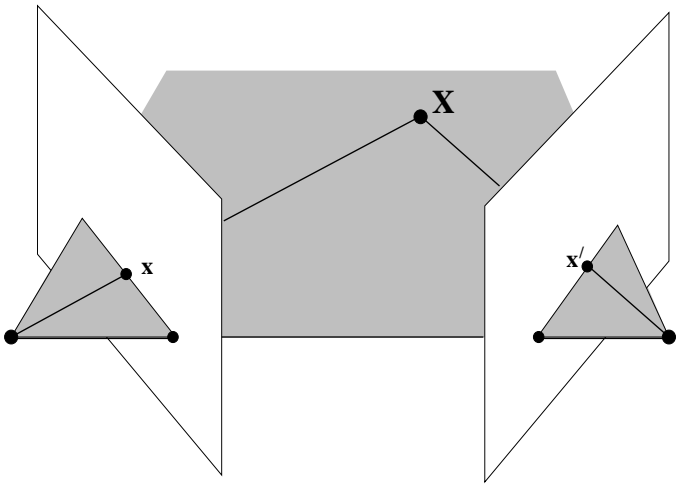


Figure 5.14: *Image points* $\mathbf{x}$ *and* $\mathbf{x}^{'}$ *back project to rays and intersect at* $\mathbf{X}$ *in 3-space.*

$$\begin{aligned} x\left(\mathbf{p}^{3\mathrm{T}}\mathbf{X}\right) - \left(\mathbf{p}^{1\mathrm{T}}\mathbf{X}\right) = 0 \\ x\left(\mathbf{p}^{3\mathrm{T}}\mathbf{X}\right) - \left(\mathbf{p}^{2\mathrm{T}}\mathbf{X}\right) = 0 \end{aligned} \tag{5.5}$$

where $\mathbf{p}^{i\mathrm{T}}$ are the rows of P. A system of the form $C\mathbf{X} = 0$ can then be formed

$$C = \begin{bmatrix} x\mathbf{p}^{3\mathrm{T}} - \mathbf{p}^{1\mathrm{T}} \\ y\mathbf{p}^{3\mathrm{T}} - \mathbf{p}^{2T} \\ x^{'}\mathbf{p}^{'3\mathrm{T}} - \mathbf{p}^{'1\mathrm{T}} \\ y^{'}\mathbf{p}^{3\mathrm{T}} - \mathbf{p}^{'2\mathrm{T}} \end{bmatrix} \tag{5.6}$$

where two equations have been included for each feature from each image, giving a total of four equations in four homogenous unknowns. We can solve for $\mathbf{X}$ using SVD which also applies to an overdetermined solution.

5.2.2 Homographic Line Based Matching

Here, we introduce a novel homographic line based matching technique that allows for fast and accurate pixel matching for semi-dense 3D reconstruction. The technique does not demand image rectification as for dense or semi-dense reconstruction [53, 102, 103] and only uses F. It is to be noted that standard techniques

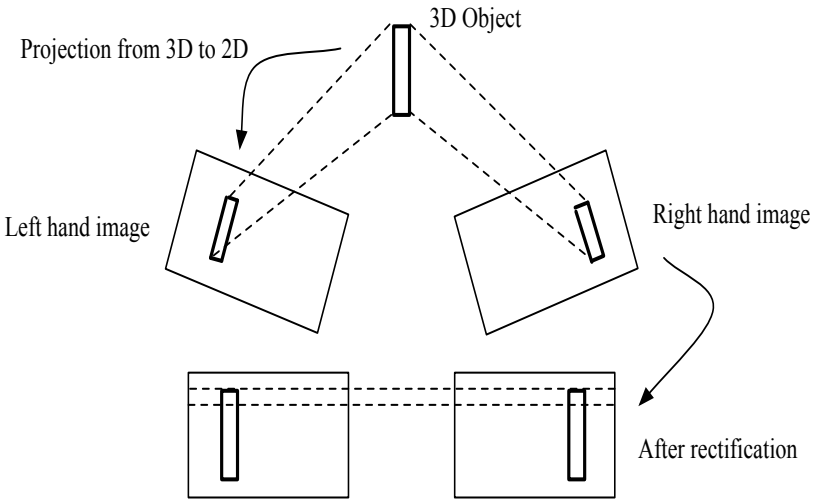


Figure 5.15: *Illustrating the effect of image rectification. The search space is reduced to corresponding lines.*

for reconstruction use rectified images for stereo matching. By doing so, the epipolar lines are aligned and one can search along the corresponding line in the other image. Figure 5.15 illustrates this.

Our technique uses homographic lines instead of epipolar lines artificially aligned together. It does away with the step of image rectification to solve the correspondence problem for scene reconstruction. Since feature matches used to compute F are already available, determining the homography between two images is a straightforward task.

Details of the Algorithm

We now introduce the algorithm to solve the correspondence problem for dense or semi-dense matching. This is done with the help of an example.

Consider two overlapping images given in Figure 5.16. The first step in using homographic lines is to find the area of overlap between the two images. The area of overlap between the images given in Figure 5.16 is near the bottom of the left hand side image and near the top of the right hand side image. To determine the area of overlap we choose one image, here the right hand side image, and populate it with a random number of points. These random points are obtained using a uniform distribution. Right hand side of Figure 5.17 gives an image populated

with random points. These points are projected onto the other image through the expression $\mathbf{x}^{'} = \mathcal{H}\mathbf{x}$. The homography is determined using feature matches, as done in Chapter 3. Using the projected points we determine which points are within the image boundaries and then using simple minmax operators on the projected points we find the area of overlap as shown in left hand side of Figure 5.17.

Figure 5.16: *A couple of overlapping images used to illustrate algorithm for homographic lines.*

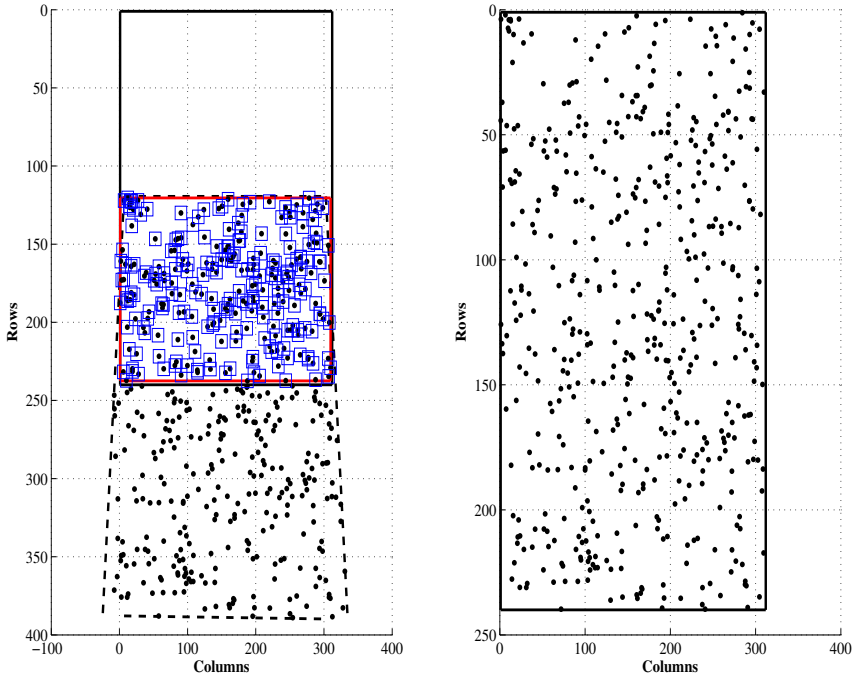
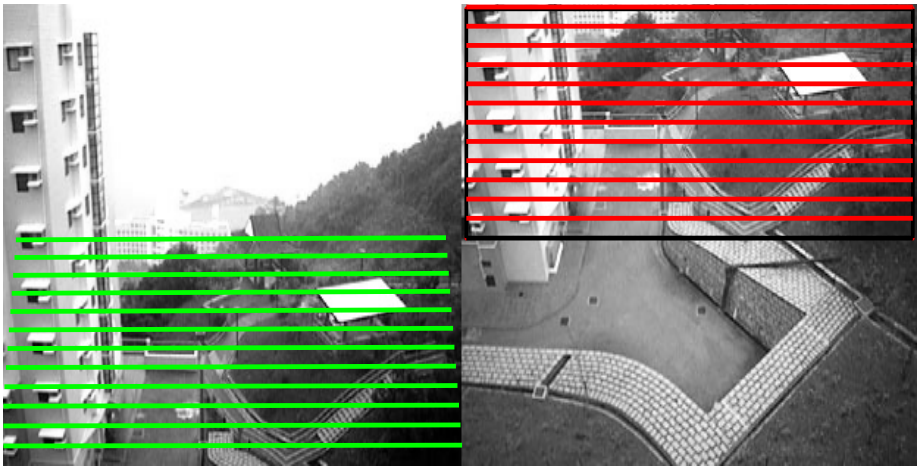
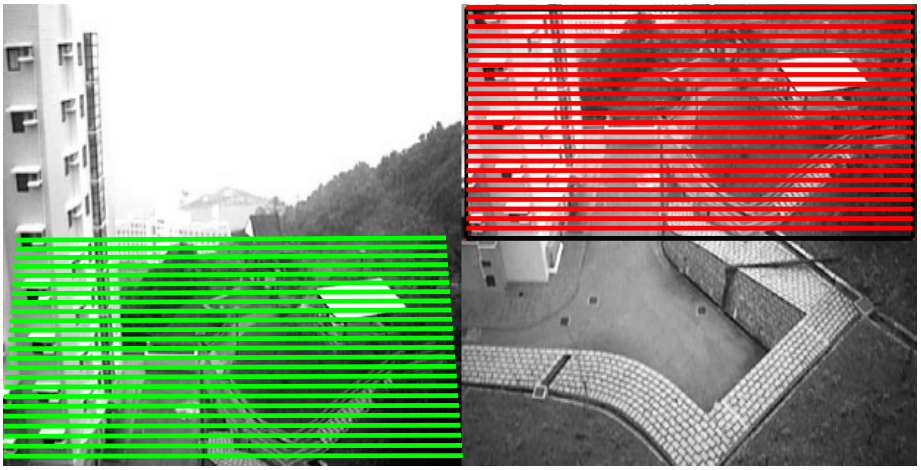


Figure 5.17: Determining area of overlap using random point projection. The blue squares represent points projected into the area of overlap.

After this, we divide the overlapping area in the right hand side images into segments. The number of segments depends on how dense a reconstruction is required. Figure 5.18 shows this division for two different segmentation values. These dividing lines can be projected into the other image into the area of overlap and so a reference is established for finding the corresponding pixel, as shown in Figure 5.18. These lines we call the homographic lines.



(a)



(b)

Figure 5.18: *Homographic lines projected from the right hand side image to the other image for two segmentation values.*

Once these lines are established, we can start searching for corresponding pixels between the two images to create a semi-dense scene reconstruction. This can be done by walking along each homographic line at a time and determining the corresponding point in the corresponding line in the other image. In order to confirm pixel-pixel correspondence, we use correlation. We take a small 3×3 patch around the pixels to be correlated p and $p^{'}$ and compute their correlation score by

$$\frac{1}{n}\sum_{x,y}\frac{(p-\bar{p})\left(p^{'}-\bar{p^{'}}\right)}{\sigma_p\sigma_{p'}} \tag{5.7}$$

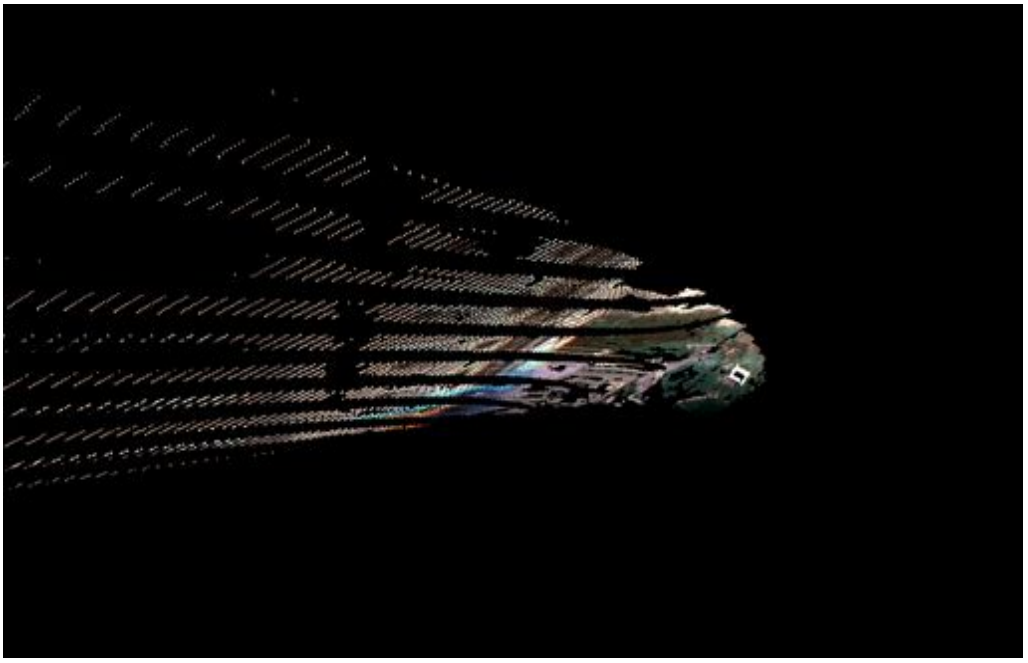
where n is the number of pixels in the patch, $\bar{p}$ is the average of p, σ is the standard deviation given by

$$\sigma_p = \sqrt{\frac{1}{n}\sum^{n}(p_i-\bar{p})}. \tag{5.8}$$

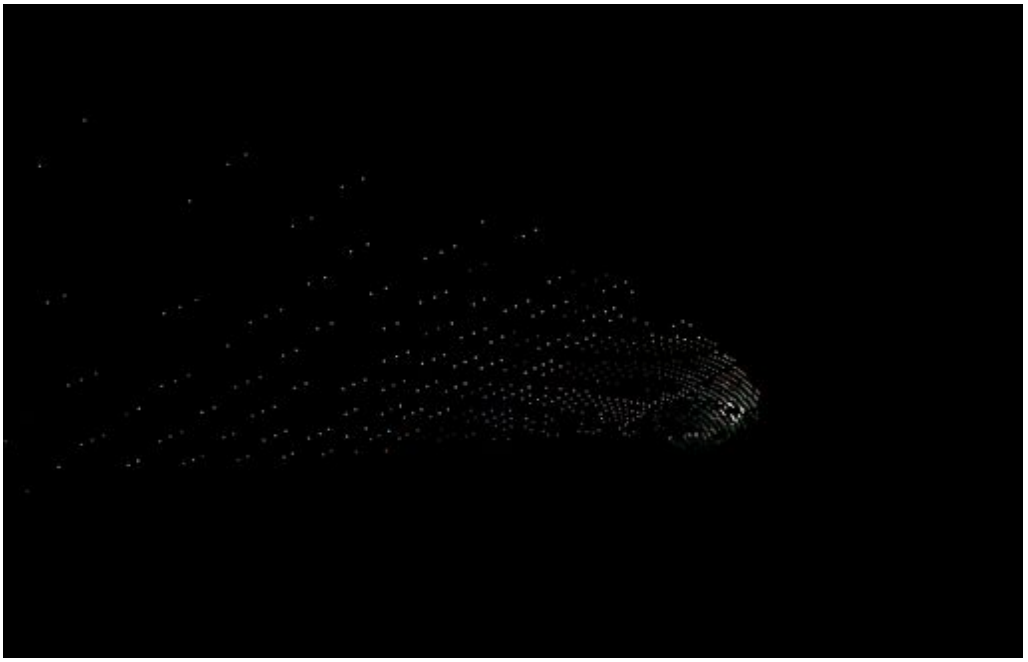
We also apply a restriction based on σ that only allows correlation if the patches have a big enough standard deviation. Meaning that there is enough variation in intensity in the patch. This allows us to differentiate between objects of interest and, for example, the sky which we do not wish to reconstruct.

Experimental analysis Here we include results from our semi-dense 3D reconstruction algorithm. The algorithm is written in C/C++ code using the OpenCV computer vision library for programming functions and OpenGL library for purposes of display. Because of the large number of reconstructed data points, we decided to use OpenGL for visualisation.

Figure 5.19 gives an example of scene reconstruction from the set of images given in Figure 5.16 for two different densities of points. Figure 5.19(a) gives the full scene reconstruction from the area of overlap, where pixel correspondences determined using our homographic line method. Figure 5.19(b) gives a semi-dense reconstruction where every fifth pixel in every fifth homgraphic line is considered. The effects of projective reconstruction are visible in both. Figure 5.20 give the



(a)



(b)

Figure 5.19: *3D scene reconstruction of Figure 5.17 with two different point densities. (a) Full scene reconstruction. (b) Semi scene reconstruction.*

time taken for reconstruction and it can be seen that semi-dense reconstruction is more than 25% faster.

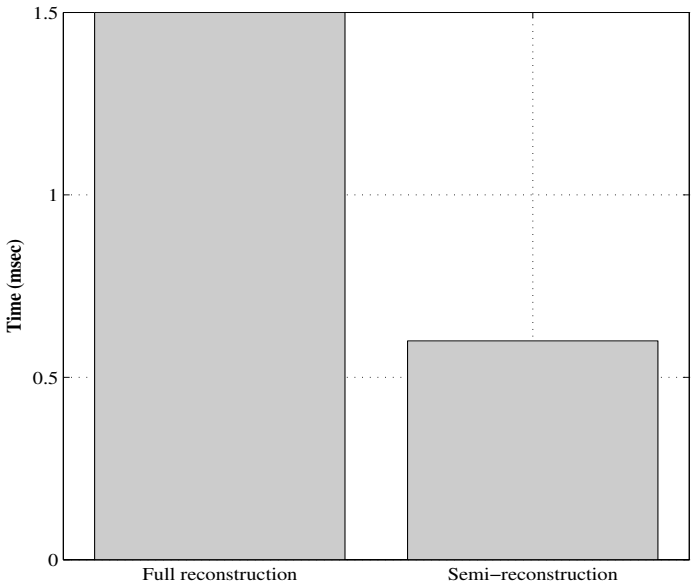


Figure 5.20: *Time taken in milliseconds for full and semi scene reconstruction of Figure 5.17.*

Figure 5.21 is an example of a couple of overlapping images reconstructed in 3D. Using random point projection we have easily determined the area of overlap and computed the homographic lines. Figure 5.22 gives the scene reconstruction for two different point densities, i.e., two different area of overlap divisions. As expected the denser reconstruction gives a better understanding of the scene. Though at greater computational expense, Figure 5.23. The time taken for full scene reconstruction is 75% more compared to semi-dense reconstruction.

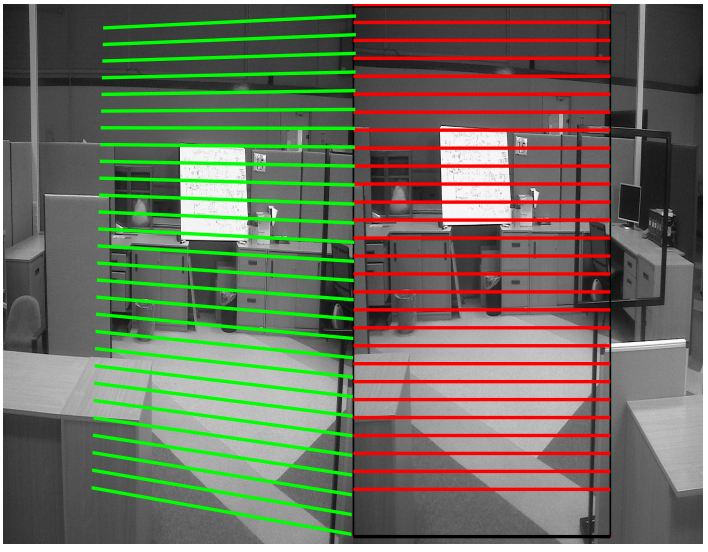
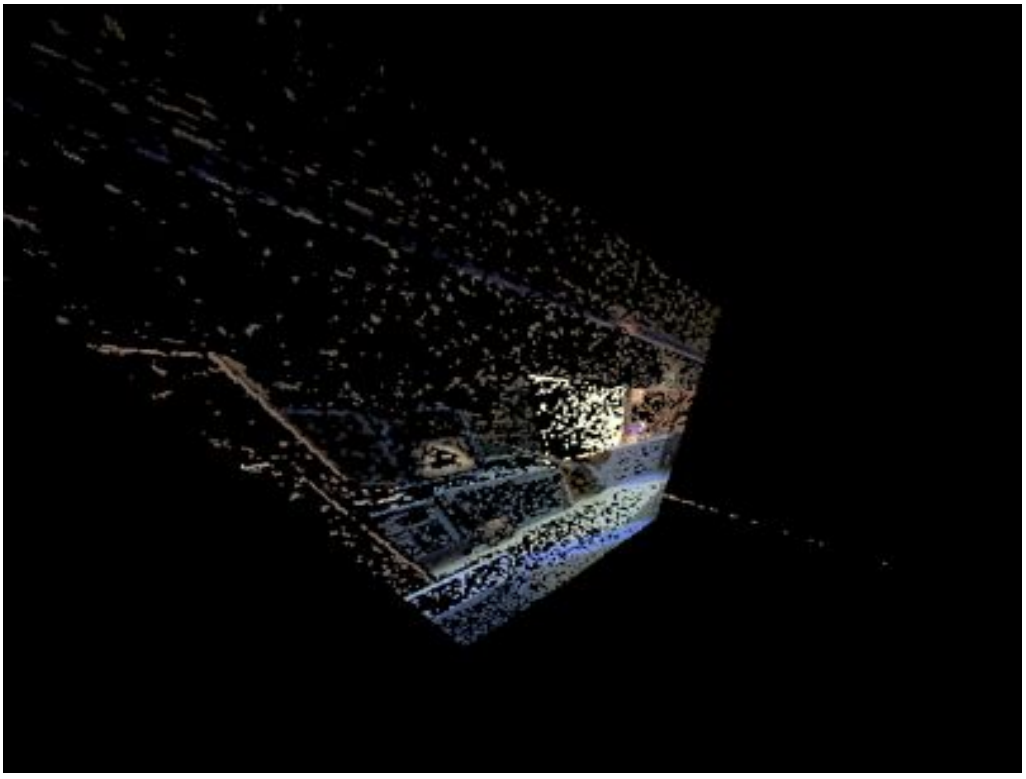
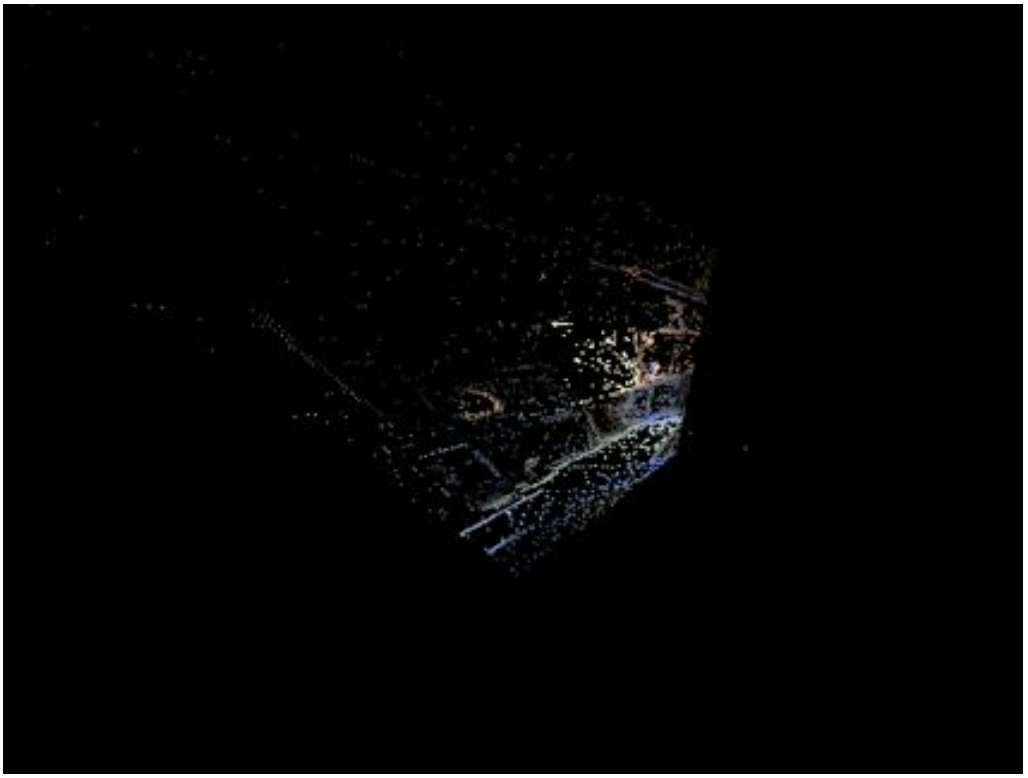


Figure 5.21: *Illustrating homographic lines in the area of overlap.*



(a)



(b)

Figure 5.22: *3D scene reconstruction of Figure 5.21 with two different point densities. (a) Full scene reconstruction. (b) Semi scene reconstruction.*

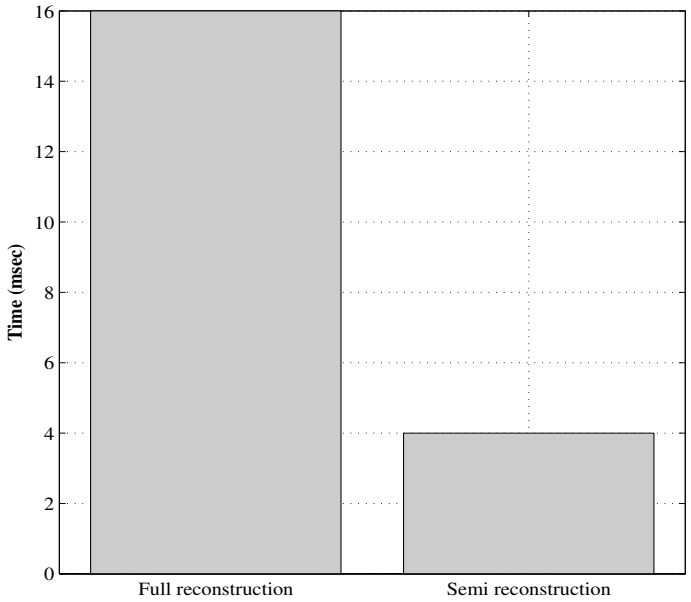


Figure 5.23: *Time taken in milliseconds for full and semi scene reconstruction of Figure 5.21.*

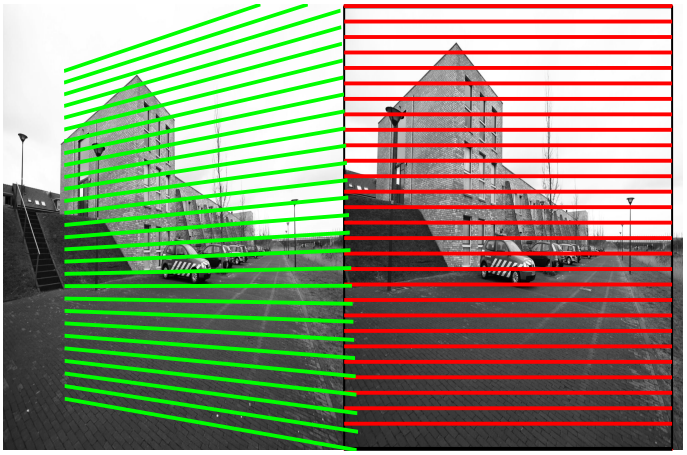
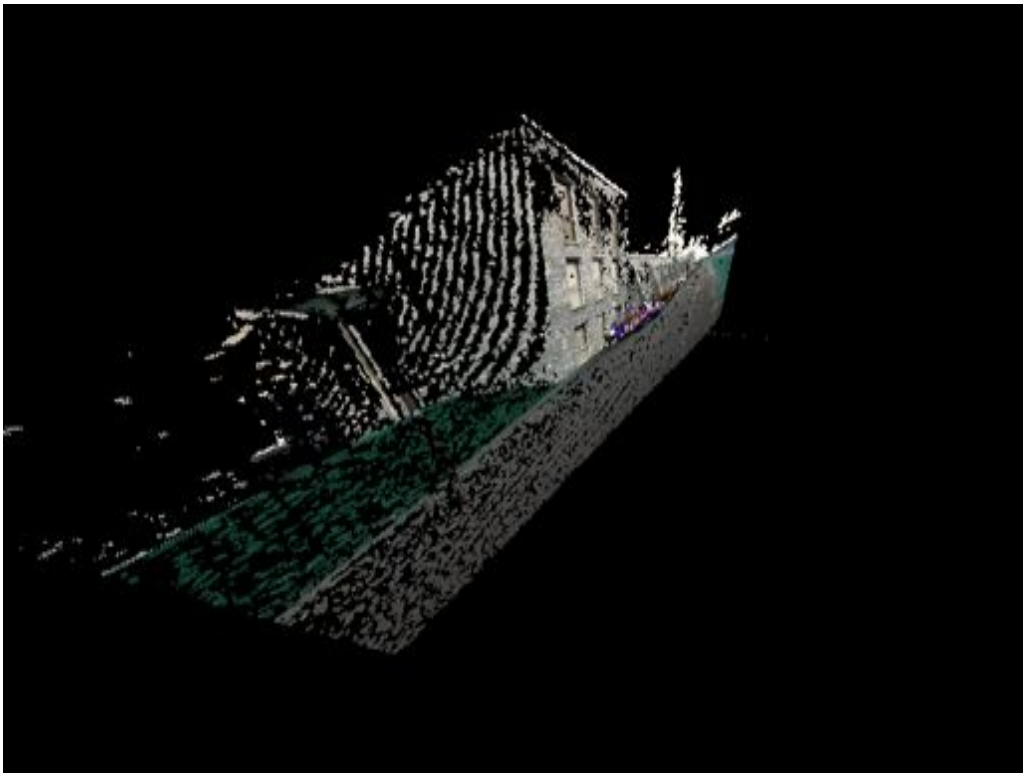
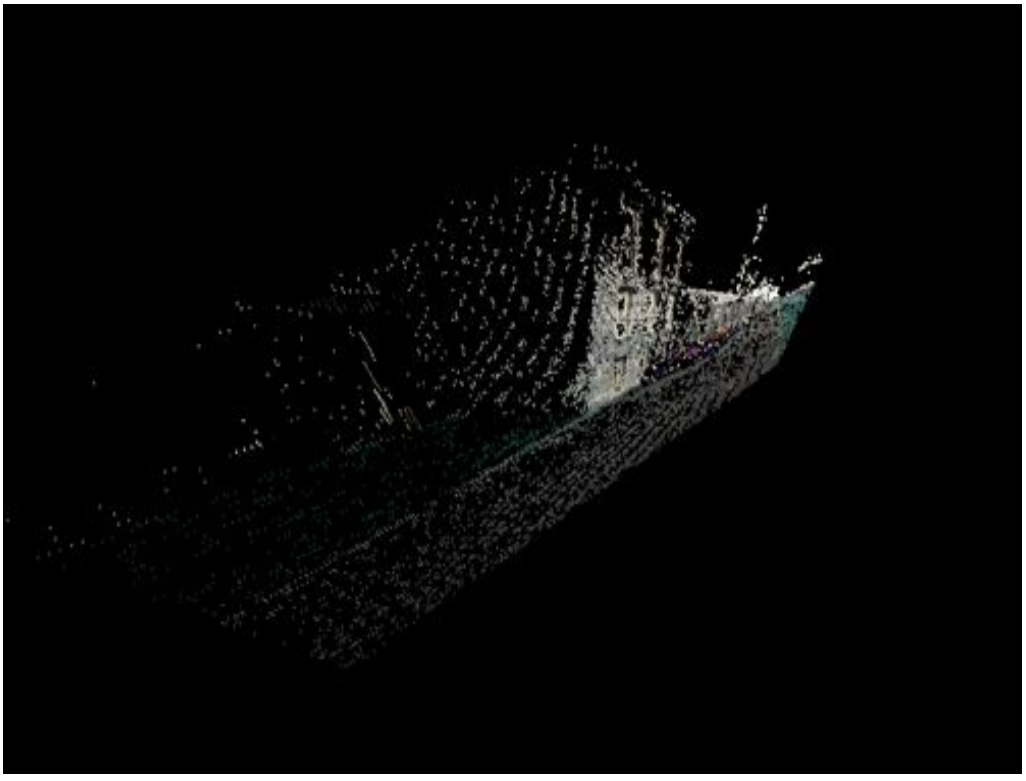


Figure 5.24: *Showing homographic lines in the area of overlap.*

Figure 5.24 is another example of a couple of overlapping images reconstructed in 3D. The area of overlap and homographic lines are shown. Figure 5.25 gives the scene reconstruction for two different point densities. The loss in detail is expected with a reduction in point density, as shown. Figure 5.26 gives the time taken for the reconstruction. The full reconstruction takes 80% more time to reconstruct the scene.



(a) Full scene reconstruction



(b) Semi scene reconstruction

Figure 5.25: *3D scene reconstruction of Figure 5.24 with two different point densities. (a) Full scene reconstruction. (b) Semi scene reconstruction.*

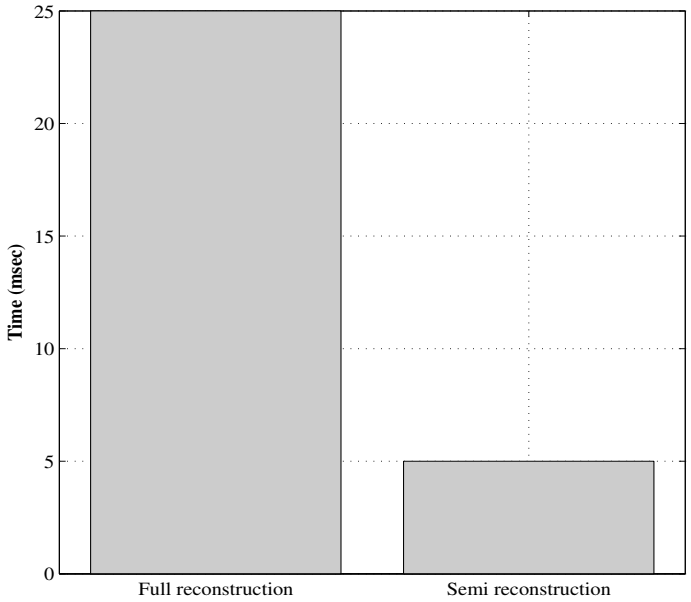


Figure 5.26: *Time taken in milliseconds for full and semi scene reconstruction of Figure 5.24.*

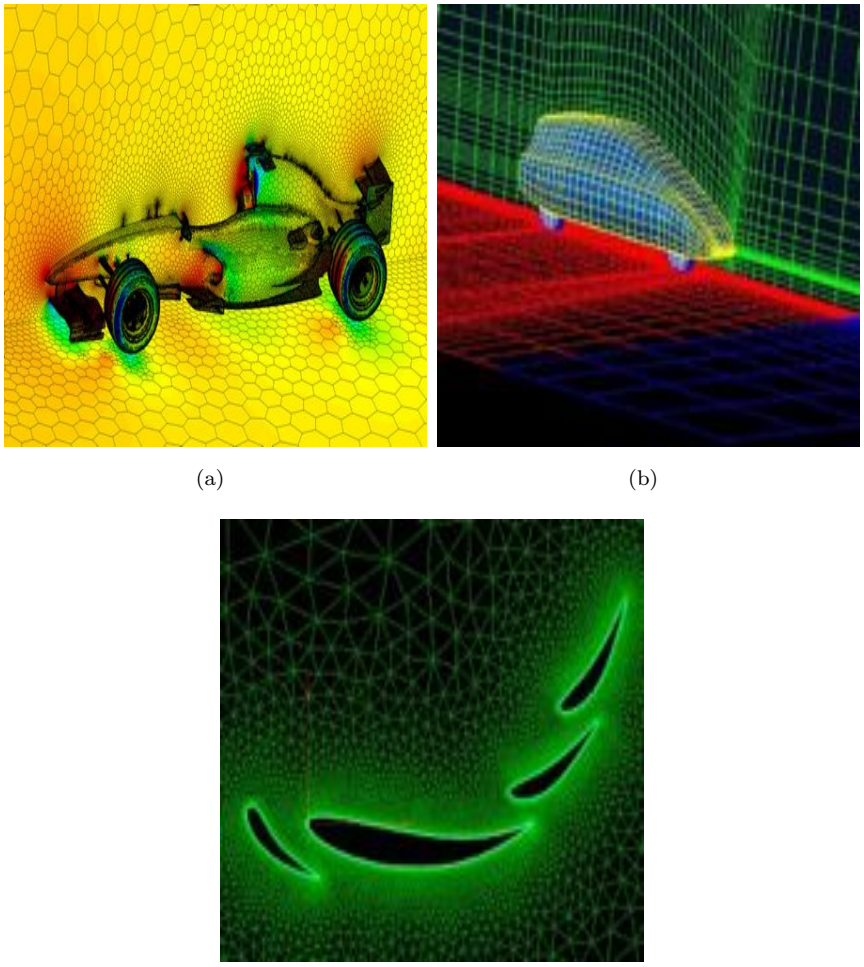
5.3 Closing Gaps in the Visualisation

In the previous section, we have shown semi-dense 3D scene reconstruction using homographic lines to solve the correspondence problem. Even when doing a dense scene reconstruction some gaps are left. This, although, not a major issue, can lead to an incomplete representation of the scene. When put into the context of surveillance, the more information that can be provided to the end user the better, allowing better judgement.

To overcome the issue of "gaps" in the reconstruction, we propose a meshing as a solution. Specifically, the technique is based on Delaunay triangulation. Following, we outline what Delaunay triangulation is, after which we apply it to scene visualisation in 3D.

5.3.1 Delaunay Triangulation

Triangulation is a major topic in computational geometry and is common to a multitude of applications, i.e., computer graphics, scientific visualisation, robotics



(a) (b)

(c)

Figure 5.27: *Types of meshes. (a) Showing a hexagonal mesh for use in flow modelling. (b) Showing a rectangular mesh for use in CFD. (c) Triangular mesh also used in modelling fluid flow.*

and computer vision, not to mention mathematical and natural science [142–144]. Figure 5.27 gives examples of different types of meshes.

Delaunay triangulation is a particular triangulation defined as follows.

Definition 5.1. Given a point set Q, the Delaunay triangulation (DT) is a specific triangulation, built on the points in Q, which specify the empty circum-circle property: the circum circle of each simplicial cell in the triangulation does not contain any input point $q \epsilon Q$.

This is illustrated in Figure 5.28, where no points $q \epsilon Q$ are within the circum-circles.

Figure 5.28: *Delaunay triangulation with no points in the circum-circle given by grey lines.*

Extending this, given a point set Q in Euclidean space E^d, a k-simplex[1], with $k \leq d$, is defined as the convex combination of k+1 affinely independent points in Q, called vertices of the simplex. A triangle is a 2-simplex and therefore has 3 vertices. An s-face of a simplex is the convex comibation of a subset of s+1 vertices of the simplex, i.e., a 2-face is a triangular facet.

A triangulation Γ defined on a point set Q in E^d space is the set of d-simplices such that,

1. a point q in E^d is a vertex of a simplex in Γ if $q \epsilon Q$
2. the intersection of two simplices in the triangulation Γ is either empty or a common face
3. the set Γ is maximal. Meaning there is not any simplex that can be added to Γ without violating the previous rules.

As previously mentioned, a triangulation Γ is a DT if the hypersphere circumscribing each simplex does not contain any point of the set Q [145]. Owing to the relationship that exists between DTs and Voronoi diagrams [145], some algorithms

[1]k-simplex is a generalisation of the notion of a triangle to arbitrary dimensions.

use this to construct DTs from Voronoi diagrams. However, direct construction methods are generally more efficient because of the fact that the Voronoi diagrams do not need to be computed and stored [142, 146].

Direct DT algorithms are classified as

- **local improvement**, where starting with a random triangulation the algorithm adjusts the faces according to the circum-circle criteria
- **incremental insertion**, starting with a convex hull of the points Q, these algorithms add points one at a time
- **incremental construction**, where the triangulation is done by successively building triangulations whose circum-sphere contain no points in Q
- **higher dimensional embedding**, consists of projecting the points into a E^{d+1} space
- **divide and conquer**, based on a recursive partioning and local triangulation, followed by a merging of resulting triangulations.

Out of these, incremental insertion methods hold the lower worst case time complexities [142]. Figure 5.29 gives an example of DT with two different data sets using incremental insertion.

Experimental analysis Here, we include results from application of DT to a sparse 3D scene reconstruction to aid visualisation. Using the feature correspondences between the images, we construct a DT on one of the image, here the left hand side image. These triangular meshes are then projected into the 3D space using the camera matrices. Texture is then applied to each patch from the corresponding patch in the image using linear interpolation.

Figure 5.30 gives a DT visualisation of the sparse 3D structure of Figure 5.16. The mesh is built over the sparse feature correspondences between the two overlapping images. These are visible as black outlined triangles in Figure 5.30(a), where the red dots are the feature points. Comparing it to the dense and semi-dense point reconstruction, DT visualisation fills all empty spaces.

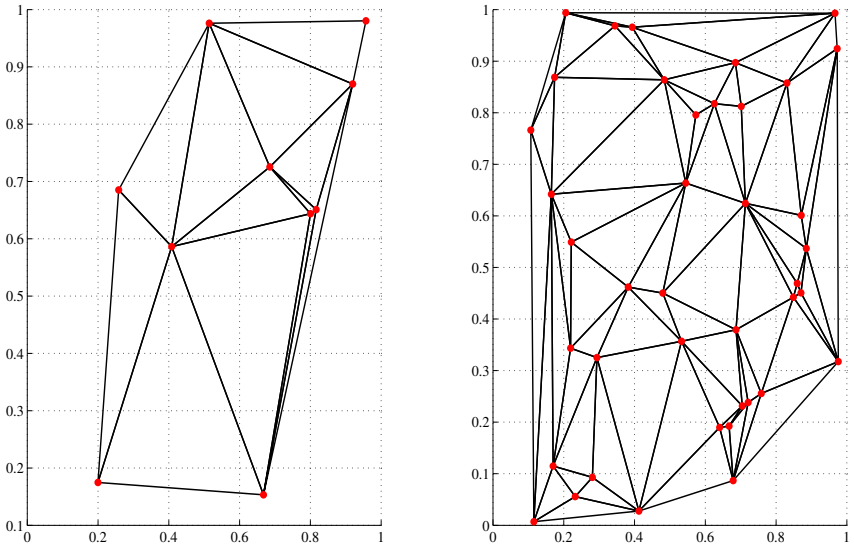
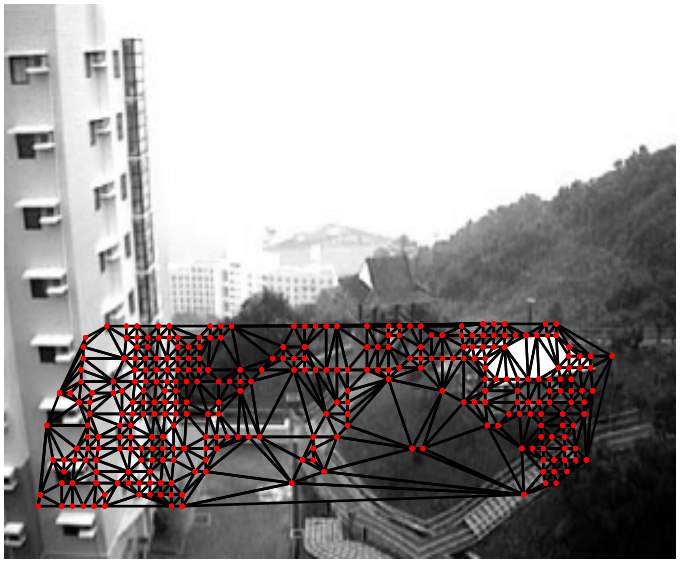
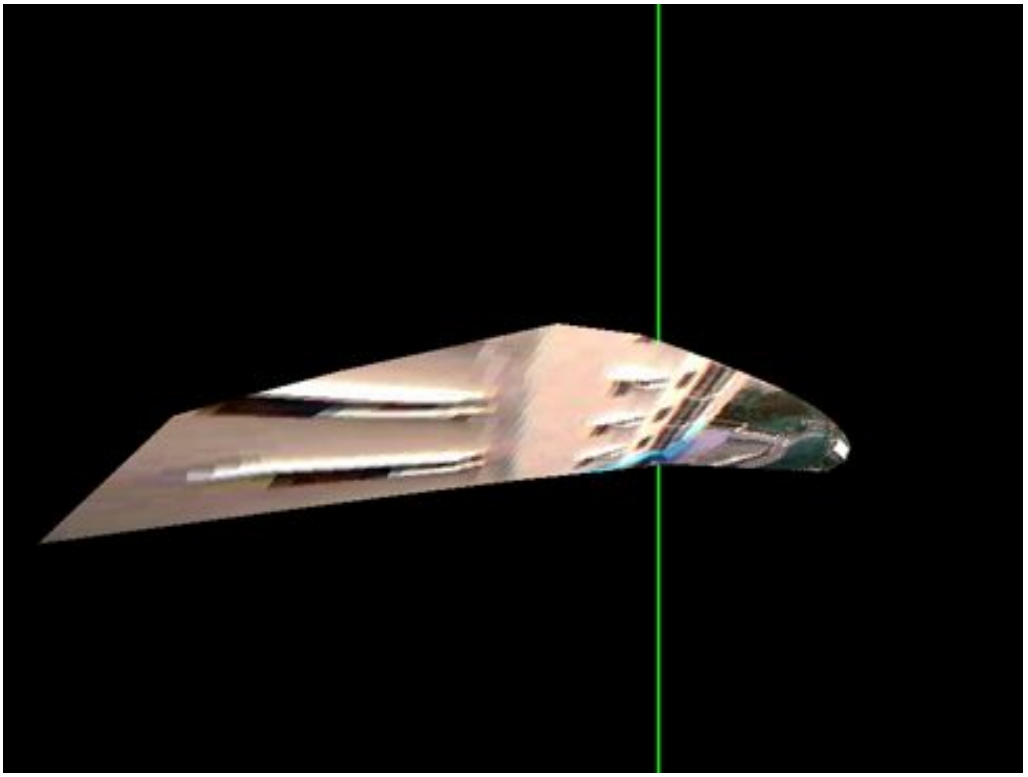


Figure 5.29: *Delaunay triangulation of two different data sets using incremental insertion. The right hand side figure contains more points.*

Figure 5.31 is an example of DT visualisation of the 3D scene from Figure 5.21. Again, we can notice how much more information is available to the end user from doing a mesh based visualisation plus rendering, since the gaps are filled. Figure 5.32 is another example of DT visualisation of the 3D scene from Figure 5.24 and same conclusions are derived.

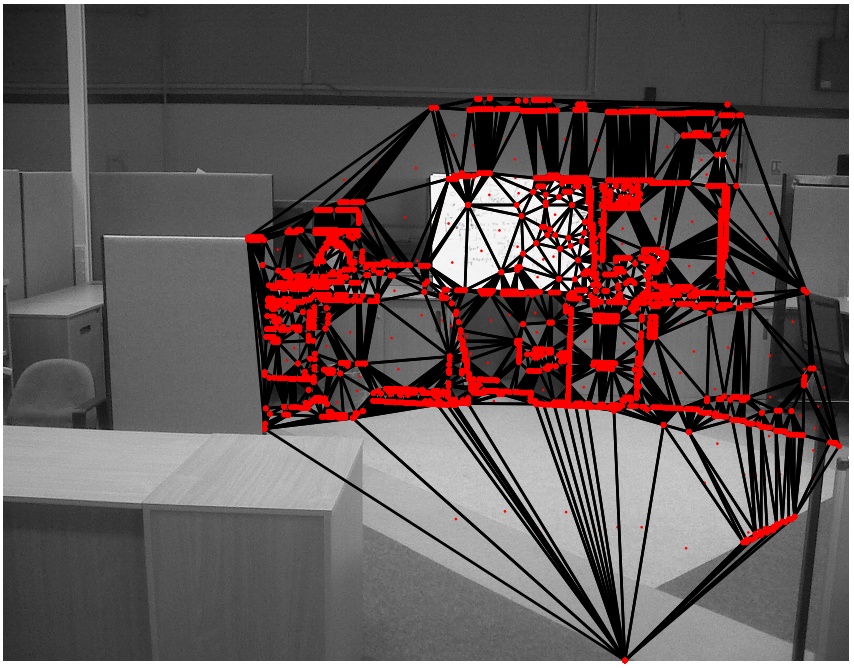


(a)

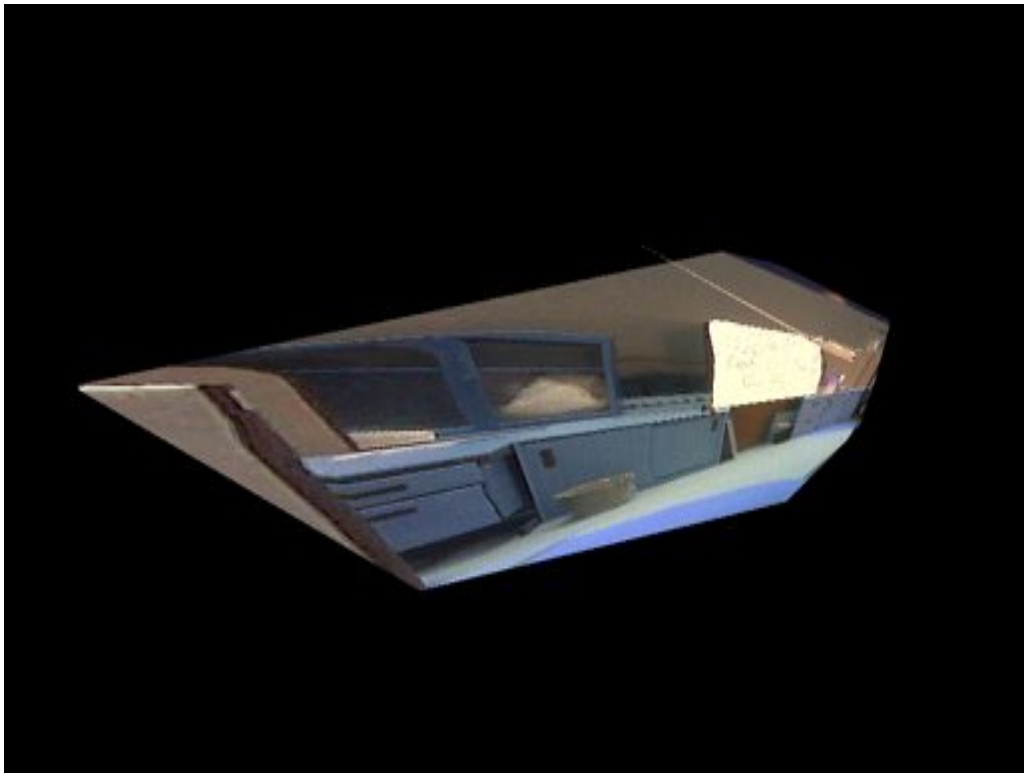


(b)

Figure 5.30: *Scene visualisation of Figure 5.16 using Delaunay triangulation. (a) 2D Delaunay triangulation of sparse feature correspondences. (b) 3D scene representation using Delaunay meshing combined with texture mapping from the image.*

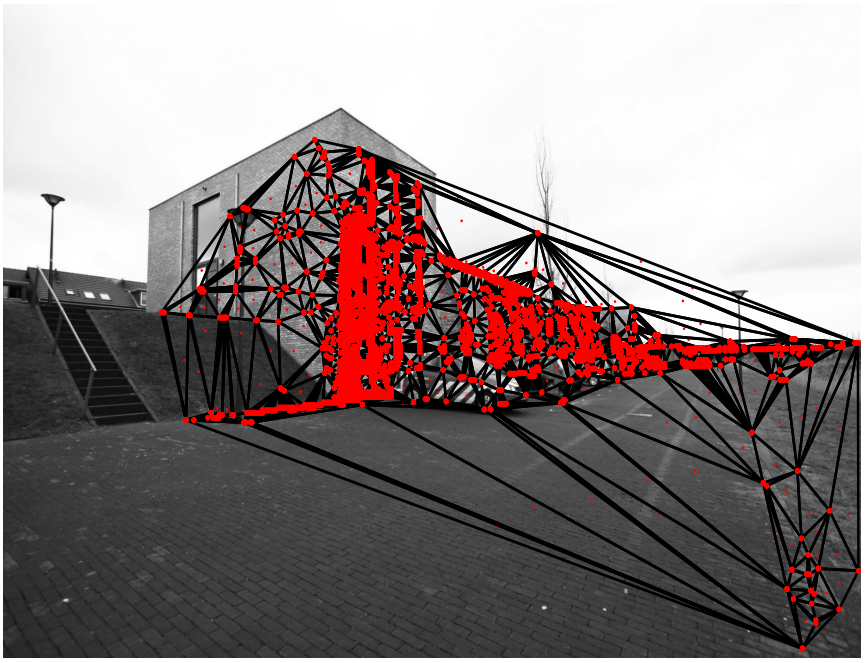


(a)

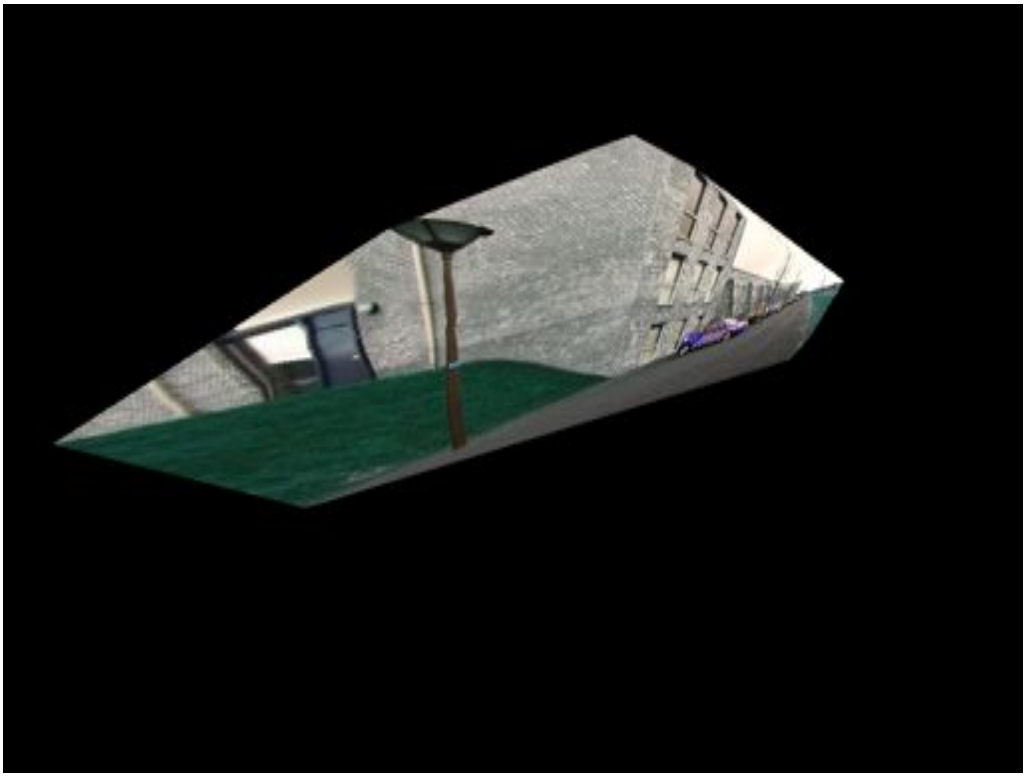


(b)

Figure 5.31: *Scene visualisation of Figure 5.21 using Delaunay triangulation. (a) 2D Delaunay triangulation of sparse feature correspondences. (b) 3D scene representation using Delaunay meshing combined with texture mapping from the image.*



(a)



(b)

Figure 5.32: *Scene visualisation of Figure 5.24 using Delaunay triangulation. (a) 2D Delaunay triangulation of sparse feature correspondences. (b) 3D scene representation using Delaunay meshing combined with texture mapping from the image.*

5.4 Chapter Conclusion

In this chapter, rapid techniques for 3D projective scene reconstruction are introduced. We have developed a novel IMU assisted feature matching technique that uses positional information of cameras provided by an autonomous platform to locate areas of overlap between images. This can be used for accurate and fast feature matching. Secondly, a new homographic line based 3D reconstruction technique is given that solves the correspondence problem without the need for image rectification. This leads to a dense or semi-dense projective scene reconstruction. We have also implemented a Delaunay mesh based visualisation of the 3D scene to cover the gaps in the reconstruction. This yields more information to the end user.

6
Cooperative Mosaicing Methods

In this chapter, we propose methods to tackle cooperative mosaicing in the 3D space. We consider a scenario with three autonomous ground vehicles surveying a scene and consequently reconstructing it. For instances when the vehicles' views overlap, we introduce a method to detect this instance and how to merge the reconstructions together. Autonomous platforms surveying a scene in the context of the scenario are shown in Figure 6.1.

We also propose image feature compression based on principal components that assists in communicating features between platforms with low communication costs. We show that, decompressing the features after communication and then using them for matching does not greatly affect matching precision.

Before moving forward, we state several assumptions made in the study.

Assumptions

- Camera parameters are unknown.
- The autonomous ground vehicles are able to communicate with each other and the main site without any restrictions.
- The vehicles are capable of decision making and autonomous navigation and have enough computing power to carry out computational tasks, e.g., image processing.

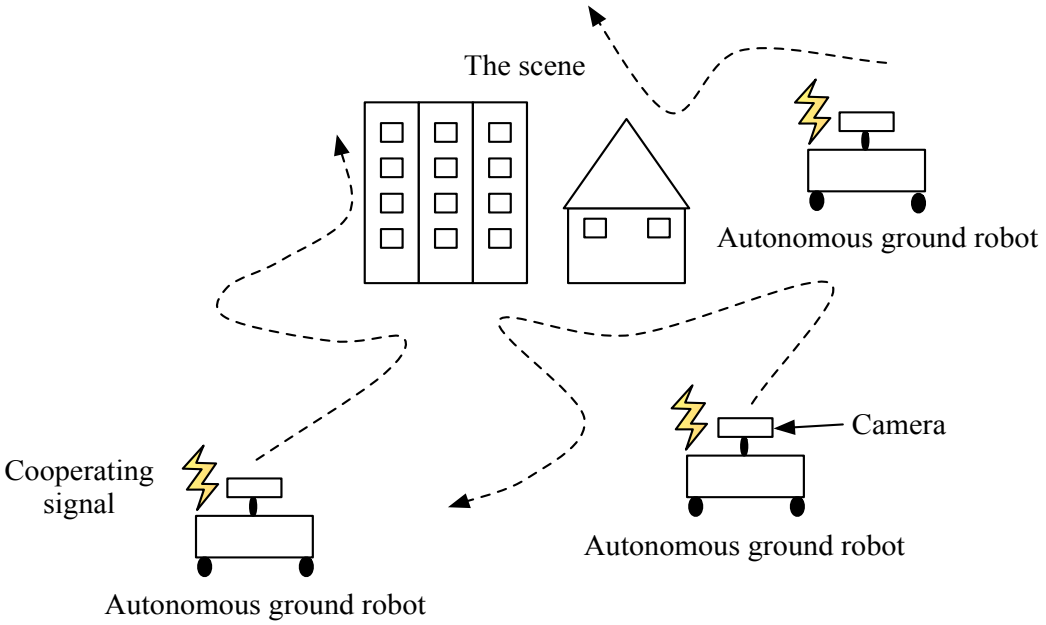


Figure 6.1: *A cooperative scenario with three platforms. The platforms are able to communicate with each other and are equipped with navigation and decision making ability.*

6.1 The Scenario

First, we define the context in which cooperative 3D mosaicing is carried out. Three autonomous ground vehicles are surveying a partially known or unknown limited space (the problem of navigation is not solved here). Each vehicle has its own path to traverse and survey. Figure 6.1 gives the general scenario.

During the survey there will be instances when the views of the three vehicles will overlap. At this point, we want the vehicles to share information and use the shared views to reconstruct the scene. At another instance, when views are common between the platforms, we want this same process repeated and require that this new reconstruction is merged to the former one.

Here, we propose methods to tackle issues pertaining to this task. In particular, feature transfer between images and determining 3D scene information from three views (image triplets) is under investigation.

6.2 Image Feature Compression

When devising a strategy for cooperative mosaicing, attention needs to be payed to costs of communicating necessary data between platforms. Here, we introduce a way to compress image feature descriptors to reduce load on the communication bandwidth. We show that even after data reduction, the features retain much of their distinctive characteristics and are able to produce good matches across overlapping images.

The technique used for compressing image feature data is a principal component analysis (PCA) based strategy. Next we highlight the theory and use of this technique to feature descriptor compression.

6.2.1 Principal Component Analysis

PCA is a procedure for determining the principal components of observed data, i.e., the principal directions in which the data varies. It ranks in descending order the direction of maximum variation in the data which is orthogonal to each other. The number of principal components is less than or equal to the number of original variables.

As the components are arranged in descending order of variance, it is possible to retain much of the information using a smaller number of principal components than the total resulting in data compression. This reduced data can be sent, for example, over a communication channel with a lower cost than the original data. Then, using the principal components, the original data can be recovered by matrix manipulations, as will be shown shortly. Furthermore, principal components of the data can be used to display multidimensional data with a reduced number of dimensions making it easier to analyse the data and identify relationships [147, 148].

Mathematical Derivation

Consider a data matrix $\mathbf{P}$, with zero empirical mean, where n rows represent a sample and m columns represent the variables of the data. PCA is a linear transformation of this data to an orthogonal subspace such that the first projection of the data contains the largest variance of data, the second projection the second largest variance of data and so on. We therefore need to determine the projections that maximise the variance.

Writing the projections as

$$\mathbf{T} = \mathbf{Pw} \tag{6.1}$$

where $\mathbf{w}$ is a unit vector, the variance is given by

$$\sigma_w^2 = \frac{1}{n} (\mathbf{Pw})^T (\mathbf{Pw}) \tag{6.2}$$

$$\sigma_w^2 = \frac{1}{n} \mathbf{w^T P^T P w} \tag{6.3}$$

$$\sigma_w^2 = \mathbf{w^T} \frac{\mathbf{P^T P}}{\mathbf{n}} \mathbf{w} \tag{6.4}$$

$$\sigma_w^2 = \mathbf{w^T V w}. \tag{6.5}$$

where $\mathbf{V}$ is the covariance matrix. We want to chose a $\overrightarrow{w}$ to maximise $\sigma^2_{\overrightarrow{w}}$ under the condition that $\overrightarrow{w}$ is a unit vector, $\overrightarrow{w}.\overrightarrow{w} = 1$. For this purpose, we look at constrained optimisation using the Lagrange multiplier.

Here, we want to maximise Equation (6.5) with the constraint $g(\overrightarrow{w}) = \overrightarrow{w}.\overrightarrow{w} = c = 1$. Re-arranging the constraint

$$g(\overrightarrow{w}) - c = 0. \tag{6.6}$$

Adding the Lagrange variable $\mathcal{L}$ to the problem leads to

$$\mathcal{L}\left(\overrightarrow{w}, \lambda\right) = f\left(w\right) - \lambda\left(g\left(w\right) - c\right) \tag{6.7}$$

where $f\left(\overrightarrow{w}\right)$ equates Equation (6.5). This is the new objective function that we differentiate with respect to $\overrightarrow{w}$

$$\frac{\partial \mathcal{L}}{\partial \overrightarrow{w}} = 0 = \frac{\partial f}{\partial \overrightarrow{w}} - \lambda \frac{\partial g}{\overrightarrow{w}}. \tag{6.8}$$

Substituting values

$$\frac{\partial \mathcal{L}}{\partial \overrightarrow{w}} = 0 = 2\mathbf{V}\mathbf{w} - \mathbf{2}\lambda\mathbf{w} \tag{6.9}$$

$$\mathbf{V}\mathbf{w} = \lambda\mathbf{w}. \tag{6.10}$$

The above expression is recognised as the eigenvector equation where $\mathbf{w}$ is the eigenvector and λ the corresponding eigenvalue. The desired vector $\mathbf{w}$ is the eigenvectors of the covariance matrix $\mathbf{V}$ and the maximising vector $\mathbf{w_k}$ will be the one associated to the largest eigenvalue λ_k. The remaining eigenvectors with their associated eigenvalues are the other principal components of the data. It is worth noting that, eigenvectors are orthogonal to each other, therefore fulfilling the requirement of orthogonality of principal components.

Data Compression Using PCA

Now that a method for computing principal components is outlined, we show how they can be used for data compression. Consider Equation (6.1), rewritten below for convenience

$$\mathbf{T} = \mathbf{P}\mathbf{w}$$

that transforms a data matrix of m variables to a new orthogonal subspace of uncorrelated values. However, not all principal components need to be kept. Retaining only the first L principal components, gives the truncated projection

$$\mathbf{T}_L = \mathbf{P}\mathbf{w}_L \tag{6.11}$$

where matrix $\mathbf{T}_L$ has n rows but only L columns instead of a the full $n \times m$ matrix. This way we have reduced the size of the transformation, though retaining much of the variance $\sigma^2_{\mathbf{w}}$. To retrieve the original data we do a matrix manipulation

$$\mathbf{P} = \mathbf{w}_L^{-1}\mathbf{T}_L \tag{6.12}$$

or

$$\mathbf{P} = \mathbf{w}_L^{T}\mathbf{T}_L \tag{6.13}$$

since $\mathbf{w}$ is a positive definite matrix.

Using the reduced transformation $\mathbf{T}_L$ it is possible to reduce a multivariable data analysis problem to a smaller case. Another important and relevant advantage of PCA is the ability to retrieve the original data from the reduced transformation with much of the original information intact. This is used when transferring data over a limited connection. We can send over a smaller packet of data and retrieve by matrix manipulations the original data, or most of the original data, at the other end.

We support this claim by help of an example on image compression. Consider Figure 6.2. This image is of size 240×320 in a double precision representation. Transferring this image over a 250kBps transfer rate takes 2.45 seconds. Figure 6.3 shows the retrieved image after applying data reduction by PCA to the original image using Equation (6.13). We show this for 4 different numbers of principal components. It is clear how the number of principal components affects how much of the original data is recovered. Obviously, using the entire set of principal components will recover all of the original data, but this is not required.

Refer to the bottom right hand side image in Figure 6.3 retrieved using 50 principal components from a total of 320. Visually, it is similar to the original image with

Figure 6.2: *Example image for compression.*

comparatively low noise. Using just 50 principal components we retain 97% of the variance of the image data. In terms of communication costs, by applying PCA we get two matrices, $\mathbf{T}_L$ and $\mathbf{w}_L$ and a vector of row means required to make the data zero mean. Transferring this over the same transfer rate takes only 1.09 seconds, which is over 50% less compared to transferring the original data. At the other end, retrieving the original data from the reduced data only requires a matrix transpose and multiplication.

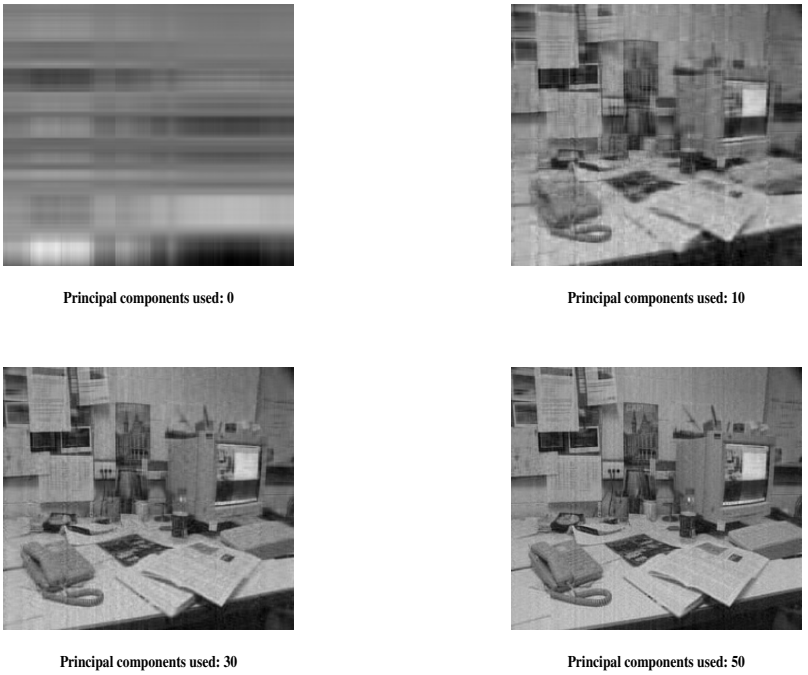


Figure 6.3: *Image retrieval of Figure 6.2 for varying number of principal components.*

6.2.2 Application to Image Features Descriptors

Inspired by how PCA can be applied to image compression and how the image can be consequently retrieved, we implement it on image features "descriptors". Considering a cooperative scenario where feature data is to be transferred between autonomous platforms reconstructing a scene, this can greatly reduce load on the communication bandwidth. Yet, we still need the retrieved features to be distinct enough to produce accurate correspondences between scenes.

We compare matching precision of descriptors between overlapping images before and after PCA and compare data communication costs of transferring them between platforms.

Features are extracted and described using Harris feature detection and 1SD description (Chapter 3). The technique is applicable to any type of descriptor. The descriptors are then compressed with PCA with a limited number of components and then retrieved. We develop an iterative technique to automatically decide on how many principal components to use based on a variance ratio.

Determining Number of Principal Components

Consider a matrix $\mathbf{P}$ where columns are components of a descriptor and the rows are a number image features $\mathbf{x}$. We are using 1SD, so the descriptor will have 128 elements for each feature. This means a matrix of size $\mathbf{x} \times 128$. Applying PCA will result in a 128×128 matrix of eigenvectors $\mathbf{w}$ and 128 eigenvalues λ. Arranging λ in descending order, the top L number will capture most of the variance of the descriptor data for the image, resulting in L principle components.

We propose an iterative technique that determines L automatically to capture 95% of the variance. Algorithm 6.1 gives the necessary steps to compute the required

number of principle components.

Algorithm 6.1: Automatic algorithm to determine number of principal components needed to capture 95% of the variance of descriptor data for given features of an image.

objective: Determine number of principal components required to capture 95% of the variance of descriptor data for given features of an image;

input : A matrix of descriptors $\mathbf{P}$ where the columns are components of a descriptor and the rows are image features $\mathbf{x}$;

algorithm:

Compute row mean $\tilde{\mathbf{p}}$ of $\mathbf{P}$;

Solve $\mathbf{P_0} = \mathbf{P}$ - $\tilde{\mathbf{p}}$;

Compute covariance matrix $\mathbf{V}$ of $\mathbf{P_0}$;

Determine eigenspace of $\mathbf{V}$ by solving $\mathbf{Vw} = \lambda\mathbf{w}$;

Order λ in descending order and sum over λ giving λ_s;

Initialise $\lambda_a = 0$;

for *L = 1:length(λ)* **do**

$\lambda_a = \lambda_a + \lambda\,(L)$

$\sigma^2_{ratio} = \frac{\lambda_a}{\lambda_s}$

if $\sigma^2_{ratio} \geq 0.95$ **then**

Break;

Using the above algorithm, we retain only L principal components of data $\mathbf{P}$ from Equation (6.11). This results in a compression of the descriptor data that we can send over a communication channel with less cost than the original data.

Experimental analysis Here, we compare performance of original image feature descriptors to retrieved descriptors after PCA using Equation (6.13). We test performance over a number of image sets given in Figure 6.4, comparing matching precision and memory requirements. The size of descriptors from both images of an image set are summed representing memory requirements before PCA. The memory requirements after PCA are also summed over the set but include $\mathbf{w}_L$, $\mathbf{T}_L$ and $\tilde{\mathbf{p}}$ which is the row norm of $\mathbf{P}$. The results are given in Table 6.1.

Figure 6.4: *Image set used to compare matching precision of descriptors before and after data compression with PCA.*

Seq No.	Precision before PCA	Precision after PCA	Size (kB) before PCA	Size (kB) after PCA
1	0.55	0.55	388.02	285.4
2	0.89	0.86	1266.46	846.80
3	0.52	0.39	449.53	382.71
4	0.77	0.71	474.11	318.81
5	0.91	0.88	326.65	259.59
6	0.96	0.94	1051.64	765.04
7	0.77	0.70	433.15	394.45
8	0.90	0.82	306.17	201.67
9	0.35	0.34	1920.00	1069.27
10	0.90	0.84	333.8	244.00

Table 6.1: *Results from matching precision and data compression in kB of descriptor data of image sequences given in Figure 6.4.*

From tabulated data in Table 6.1, we can see that applying PCA to descriptors does not greatly affect their matching precision. From tested images, an average decrease in precision is just over 6%. On average, we get data compression of more than 31% after PCA. This equates to 31% less time to transfer the data. Pitted against decrease in precision, the benefits of PCA in terms of communication costs makes it an effective solution. Interesting to note, transferring all the feature data of the test image set before PCA takes 27 seconds over a 256 kBps connection, whereas it takes only 18 seconds transferring data from PCA over the same connection.

6.3 Merging Reconstructions from 2 Platforms

As we are considering three platforms surveying and reconstructing a scene, we need to devise a method to merge the reconstructions together. We propose a solution for reconstructing three images at a time, one from each platform and then merging them together. For this purpose, we introduce the trifocal tensor which is similar to the fundamental matrix, but is instead suitable for three overlapping views. Before introducing the tensor we give reasons for its use.

Why Use the Trifocal Tensor ?

Here we motivate the use of the tensor. Considering three views at a time, there exist in general two ways to compute the reconstruction of a scene. Both require determining the camera matrices for each view which are determined from the fundamental matrix F. The first way is to take two images at a time and compute F between each image and consequently the camera matrices using camera resectioning [24, 55, 101]. This technique however requires that the three fundamental matrices are compatible, i.e., the camera matrices are non-collinear. The condition for three fundamental matrices F_1, F_2 and F_3 is given as

$$\mathbf{e}_{23}^{\mathrm{T}}\mathrm{F}_{21}\mathbf{e}_{13} = \mathbf{e}_{31}^{\mathrm{T}}\mathrm{F}_{32}\mathbf{e}_{21} = \mathbf{e}_{32}^{\mathrm{T}}\mathrm{F}_{31}\mathbf{e}_{12}. \tag{6.14}$$

where $\mathbf{e}$ are the epipoles and T is a transpose.

In case the above condition is not met, a least-squares solution will be required, which is only suitable for compatible camera matrices [24]. This, however, is not the case for the tensor. It has no issues with collinearity of the camera matrices and can determine a unique solution for them from three overlapping views. Since we are considering freely moving platforms, cameras occupying the same plane is a possibility and therefore the tensor is the best solution to handle image triplets.

6.3.1 Trifocal Tensor based Reconstruction

The trifocal tensor ($\mathcal{T}$) encapsulates geometric relations between three views that are independent of scene structure [24]. It can relate points and lines in two views to corresponding points and lines in a third view. The tensor depends on motion and internal parameters of the camera but can also be calculated using feature correspondences as will be shown [149].

Let the camera matrices for three views, as shown in Figure 6.5, be $\mathbf{P} = [I|0]$, $\mathbf{P}^{'} = [\mathbf{A}|\mathbf{a}_4]$ and $\mathbf{P}^{'} = [\mathbf{B}|\mathbf{b}_4]$, where $\mathbf{A}$ and $\mathbf{B}$ are 3×3 matrices and the vectors $\mathbf{a}_i$ and $\mathbf{b_i}$ are the i-th columns of the respective camera matrices for $i = 1...4$. The trifocal tensor is then given by

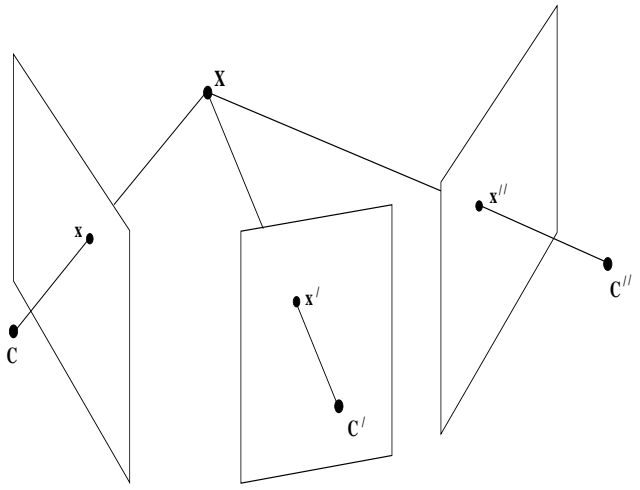


Figure 6.5: *Projection of point* $\mathbf{X}$ *in three views with centre* $\mathbf{C}$, $\mathbf{C}^{'}$ *and* $\mathbf{C}^{''}$.

$$\mathcal{T}_i = \mathbf{a}_i \mathbf{b}_4^{\mathrm{T}} - \mathbf{a}_4 \mathbf{b}_i^{\mathrm{T}}. \tag{6.15}$$

Without knowing the camera's internal parameters, we can determine $\mathcal{T}$ by exploiting 1 of 5 trifocal incidence relations [24]. These incidence relations, also known as the triliniearities, are conditions which a tensor must satisfy. They include correspondences between points and lines, points and points and lines and lines. Here, we exploit the relationship between points and points in three views, which in matrix notation is given as

$$\left[\mathbf{x}^{'}\right]_{\times} \left(\sum_i x^i \mathcal{T}_i\right) \left[\mathbf{x}^{''}\right]_{\times} = 0_{3\times 3} \tag{6.16}$$

where $\mathbf{x}$ is a feature in one image, $\mathbf{x}^{'}$ is a feature in a second image and $\mathbf{x}^{''}$ is a feature in the third image. How to estimate $\mathcal{T}$ using this expression is given shortly.

Retrieving the Camera Matrices

We can retrieve the camera matrices from the trifocal tensor up to a projective ambiguity, as in the case for the fundamental matrix. The first camera is chosen as $\mathbf{P} = [I|0]$. The second and third camera matrices are expressed as

$$\mathbf{P}^{'} = \left[[\mathcal{T}_1, \mathcal{T}_2, \mathcal{T}_3]\, \mathbf{e}^{''} | \mathbf{e}^{'} \right] \tag{6.17}$$

and

$$\mathbf{P}^{''} = \left[\left(\mathbf{e}^{''} \mathbf{e}^{''T} - \mathrm{I} \right) [\mathcal{T}_1, \mathcal{T}_2, \mathcal{T}_3]\, \mathbf{e}^{'} | \mathbf{e}^{''} \right] \tag{6.18}$$

where $\mathbf{e}^{'}$ and $\mathbf{e}^{''}$ are epipoles in the second and third image. Let $\mathbf{u_i}$ and $\mathbf{v_i}$ be the left and right null-vectors of $\mathcal{T}_i$. Then the epipoles are obtained as the null-vectors of the following

$$\mathbf{e}^{'} = [\mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3] = 0 \tag{6.19}$$

and

$$\mathbf{e}^{''} = [\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3] = 0 \tag{6.20}$$

Computating $\mathcal{T}$

Since the trifocal tensor has 3 indices, it is conveniently represented in tensor notation instead of standard matrix notation. Please refer to [24] for a quick tutorial on tensor notation.

Writing Equation (6.16) in tensor notation leads to

$$x^i x^{'j} x^{''k} \epsilon_{jqs} \epsilon_{krt} T_i^{qr} = 0 \tag{6.21}$$

where T_i is i-th tensor and ϵ is a tensor representing vector product. This expression is of the form $C\mathbf{t}$, where $\mathbf{t}$ is a 27 element vector of entries of the tensor. Since T has 27 entries, a minimum of 26 equations are required to solve for $\mathbf{t}$ up to scale.

In constructing $C\mathbf{t} = \mathbf{0}$ from Equation (6.21), it is not necessary to use the complete set of equations for each correspondence, since not all are linearly independent. All choices of s and t in Equation (6.21) lead to a set of 9 equations, of

which only 4 are linearly independent. These maybe obtained by choosing two values each for s and t.

For a given choice of s and t, Equation (6.21) can be expanded as (an example expansion is given in Appendix B)

$$x^k \left(x'^i x''^m T_k^{jl} - x'^j x''^m T_k^{il} - x'^i x''^l T_k^{jm} + x'^j x''^l T_k^{im} \right) = 0^{ijlm} \tag{6.22}$$

when $i, j \neq s$ and $l, m \neq t$. Setting $j = m = 3$ and letting $i, l = 1, 2$ gives four different equations in terms of the observed image feature coordinates.

We require at least 7 feature correspondences across three views to estimate the tensor. It is also important that the geometric constraint $T_I^{jk} = a_i^j e''^k - e'^j b_i^k$ is enforced. Algorithm 6.2 gives the general algorithm to compute a geometrically valid tensor.

Algorithm 6.2: Computing the trifocal tensor minimising algebraic error.

objective: Given a set of feature correspondence in three views, compute the trifocal tensor;

input : Feature correspondences $\left(\mathbf{x}, \mathbf{x}, \mathbf{x}''\right)$;

algorithm:

From set of feature correspondences determine set of equations of the form $A\mathbf{t} = \mathbf{0}$;

Solve $A\mathbf{t} = \mathbf{0}$ using SVD;

Find the two epipoles e' and e'' from T_i^{jk};

Construct a 27×18 matrix E such that $\mathbf{t} = \mathrm{E}\mathbf{a}$ where $\mathbf{t}$ consists of the entries of T_i^{jk}, $\mathbf{a}$ is a vector representing entries of $\mathbf{a_i^j}$ and $\mathbf{b_i^k}$, and where E represents $T_I^{jk} = a_i^j e''^k - e'^j b_i^k$;

Solve the minimisation problem: minimise $||A\mathrm{E}\mathbf{a}||$ subject to $||\mathrm{E}\mathbf{a}|| = \mathbf{1}$.

Example results We show an example reconstructing three views using the trifocal tensor. We use Algorithm 6.2 to estimate $\mathcal{T}$ and retrieve the camera matrices using Equations (6.17) and (6.17). Reconstruction is then performed from triangulation as in Chapter 5.

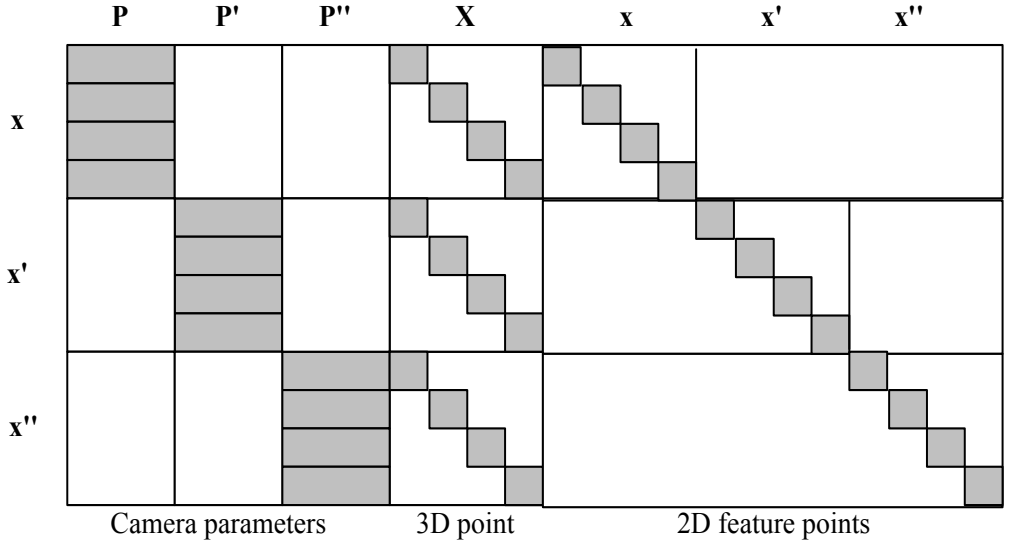


Figure 6.6: *Form of the Jacobian matrix for three camera matrices, as in the case for the trifocal tensor, 4 feature correspondences across three views and 4 3D points.*

Furthermore, we perform bundle adjustment BA to optimise the camera matrices, feature points and reconstructed points using non-linear optimisation. We have previously used BA for optimising estimates of the 2D homography in Chapter 3 but here we employ it in the 3D space. The cost function to be minimised is

$$\sum_k d\left(\mathbf{x}_k, \hat{\mathbf{x}}_k^{'}\right) + d\left(\mathbf{x}_k^{'}, \hat{\mathbf{x}}_k^{'}\right) + d\left(\mathbf{x}_k, \hat{\mathbf{x}}_k^{'}\right) \tag{6.23}$$

where the points $\hat{\mathbf{x}} = \mathrm{P}\mathbf{X}, \hat{\mathbf{x}}^{'} = \mathrm{P}^{'}\mathbf{X}$ and $\hat{\mathbf{x}}^{''} = \mathrm{P}^{''}\mathbf{X}$.

Figure 6.6 shows the structure of the Jacobian matrix J used. The form consists of the 3 cameras, as is the case with the tensor, four feature points from each image and therefore four 3D points. The parameters of the camera matrices are independent of the other camera matrices, the 3D points are independent of each other but depend on the features and the feature points are only dependent on themselves.

Figure 6.7 gives the example image triplet used to estimate the trifocal tensor. It also shows common features between images used to estimate the tensor. They are Harris features matched using 2SD plus RanSaC for outlier removal. Figure 6.8 gives the projection of the common features in 3D space from the camera matrices extracted from the tensor. The red circles are optimised points and black dots is

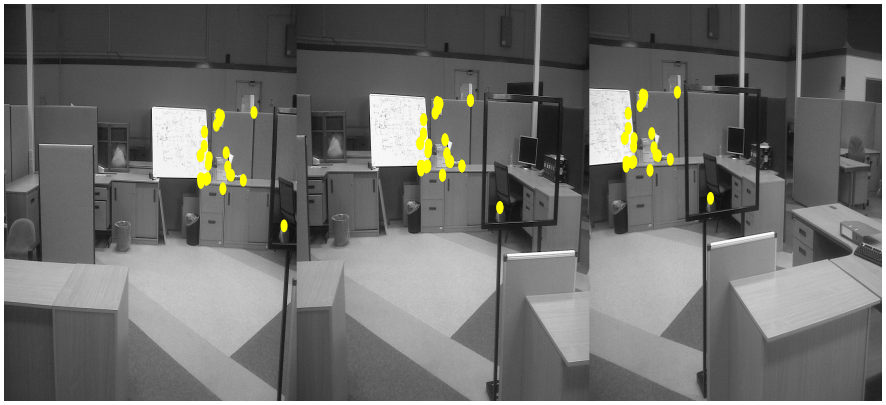


Figure 6.7: *Image set used to reconstruct scene given in Figure 6.8.*

the original unoptimised data. Figure 6.9 gives the average reprojection error from which it can be seen that by doing BA, global error is reduced.

Figure 6.10 gives another example of scene reconstruction from three views given in Figure 6.11. Figure 6.12 gives the global error which is reduced by 8% after BA.

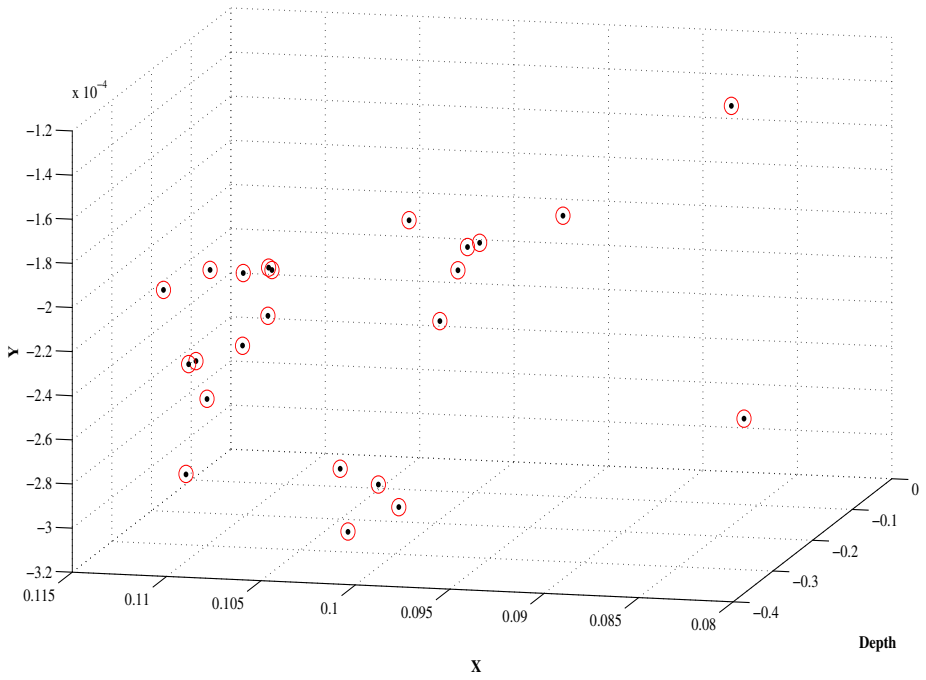


Figure 6.8: *Points reconstructed in 3D from camera matrices obtained from the trifocal tensor applied to images in Figure 6.7. Red circles show optimised points and black points show unoptimised points.*

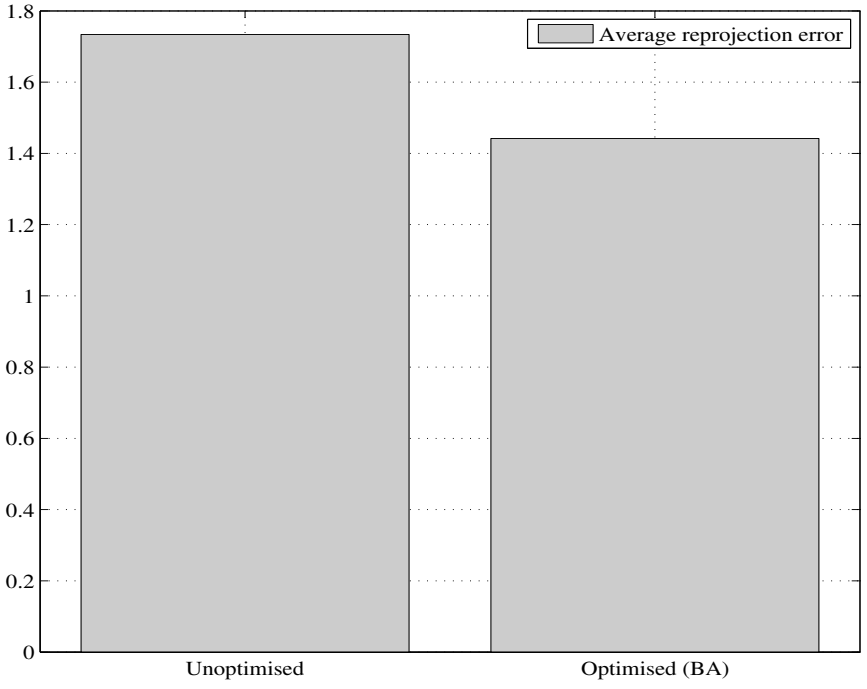


Figure 6.9: *Comparing average reprojection error of reconstruction of images in Figure 6.7. A difference of more than 17% is observed.*

Figure 6.10: *Image set used to reconstruct scene given in Figure 6.11.*

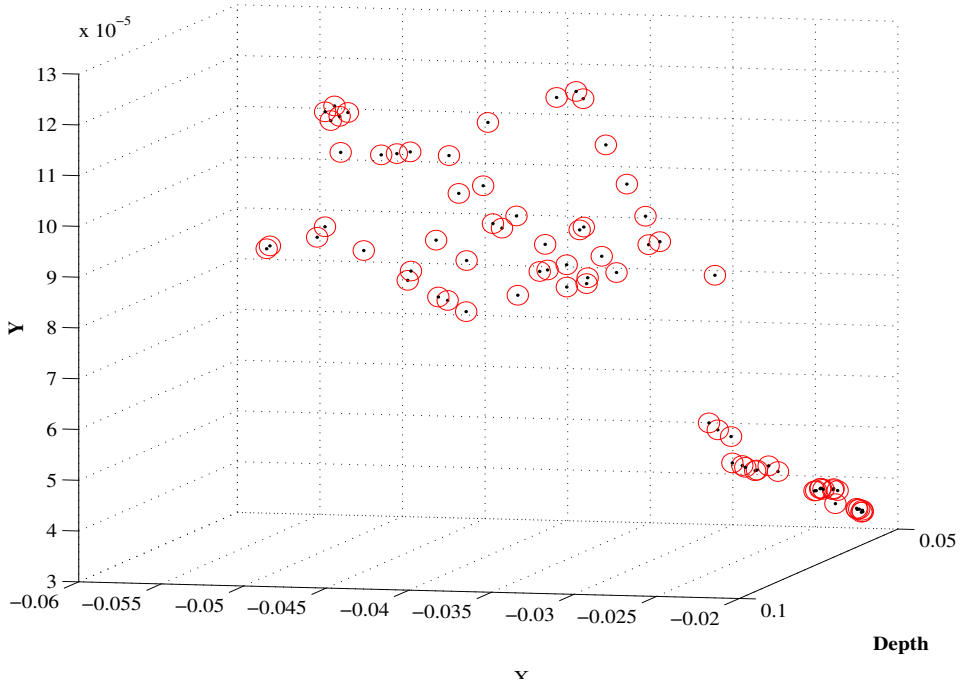


Figure 6.11: *Points reconstructed in 3D from camera matrices obtained from the trifocal tensor applied to images in Figure 6.10. Red circles show optimised points and black points show unoptimised points.*

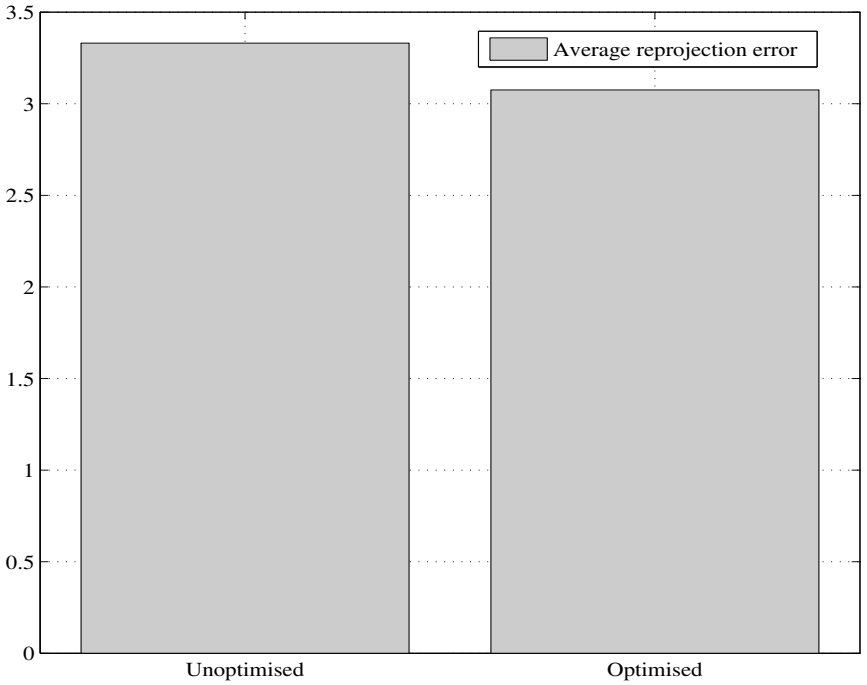


Figure 6.12: *Comparing average reprojection error of reconstruction of images in Figure 6.10. A difference of more than 8% is observed.*

6.3.2 Merging 2 Trifocal Reconstructions

After getting two scene reconstructions from a tensor each, we need a way to merge them. We do this by determining the projective transformation between the two reconstructions in 3-space. This transformation is similar to a 2D homography but is instead applied to 3D points. As such, it is estimated using corresponding points between the two 3D scenes. Therefore, before merging the scenes together, we need to determine the corresponding points between the scenes.

3D Scene Correspondence

We achieve this correspondence in the 2D space, i.e., the image planes between the image sets. Consider that we have two sets of three images each as given in Figure 6.13. Here features in between sets are matched. We facilitate this matching by using colour coding of features introduced in Chapter 3: only features with the same colour code between the sets is considered for a match. This increases speed

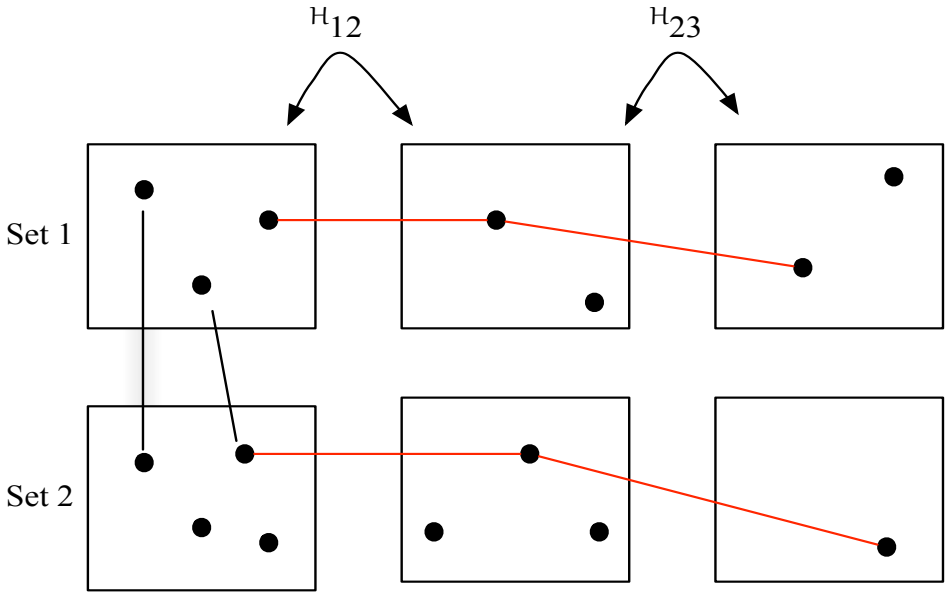


Figure 6.13: *Harris feature matching using inter-image homography. The inter-image homographies are used to project features into the other views to get matches.*

and reduces chances of false matches as was shown. Furthermore, it helps in easy management of features.

After establishing inter set matches, we determine which image between the sets has the most matches with the corresponding image in the other set. In the illustration in Figure 6.13 it is the first image of each set. Since we apply RanSaC for outlier removal, we have the inter image homographies $\mathcal{H}$ between the images of each individual set, e.g. $\mathcal{H}_{12}$. Using this we can propagate the inter set matches into the remaining images of each set. This is more clear from Figure 6.13 where matches given in red are established using inter image homographies. We need these matches to relate similar points in the 3-space after triangulation to determine the 3D homographic transformation between the two scene reconstructions.

Experimental analysis Figure 6.14 shows two image sets of image triples which are matched using the above described technique. In this example, the first images of each set produce the most matches, so we will use features from these images and propagate them into the remaining two images. The total number of image features used is shown in Figure 6.15.

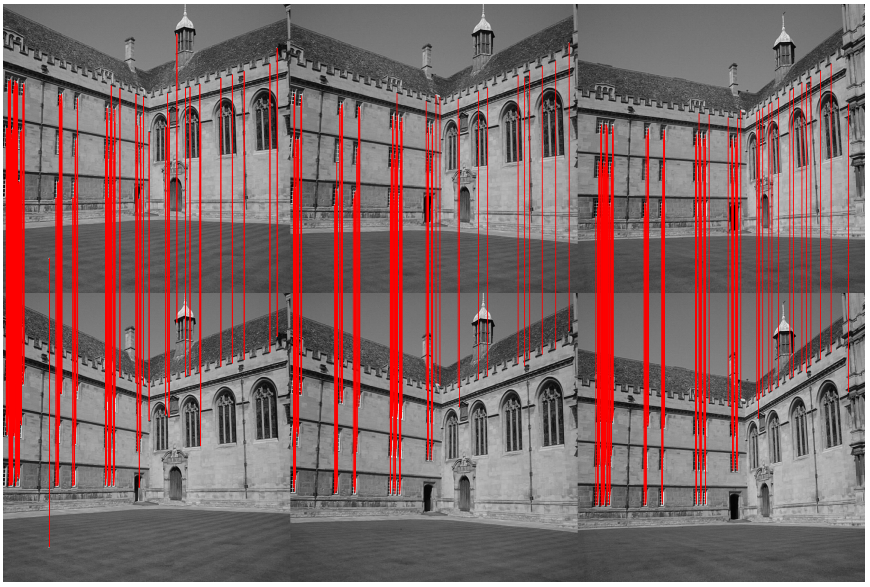


Figure 6.14: *Example matches between two sets with three images each. Between sets matching is done with help from feature clustering based on colour codes.*

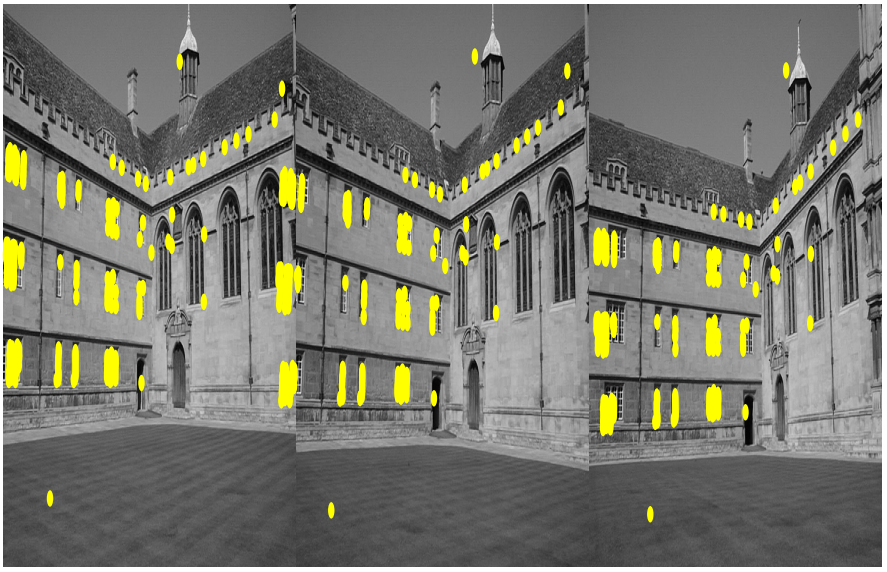
Figure 6.16 shows the reconstruction of the scene given in Figure 6.14 from the two sets of three images each solved using the tensor. Notice the similarity of the reconstruction but the obvious change in axis. We wish to merge these reconstructions together. This is accomplished using a transformation

$$\mathbf{X}_2 = V\mathbf{X}_1 \tag{6.24}$$

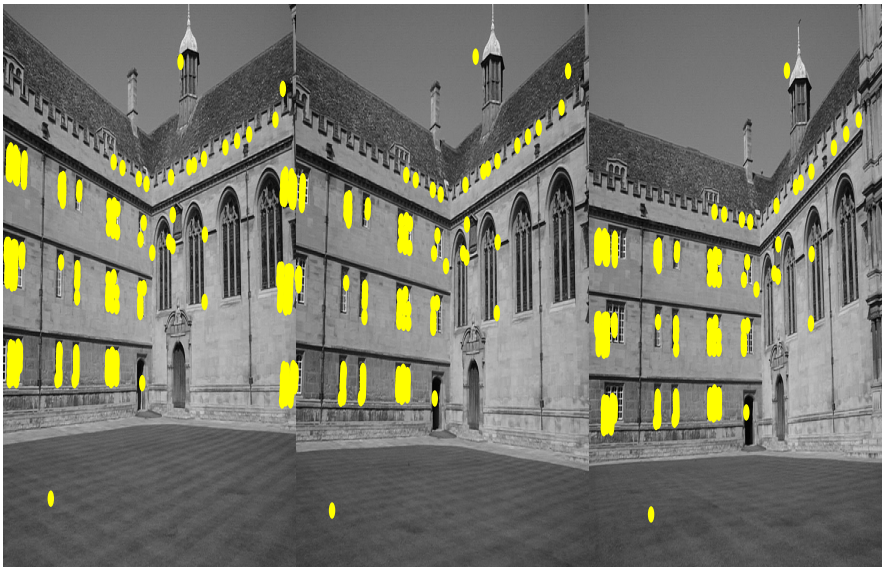
where V is a 4×4 matrix of projective transformation and $\mathbf{X_2}$ and $\mathbf{X_1}$ are 3D points in homogenous coordinates for set 1 and 2 respectively.

Since we know the matching points in 2D between the sets from Figure 6.15, this transformation can be estimated. Re-arranging (6.24) and assembling in design matrix of the form $C\mathbf{v} = \mathbf{0}$ where vector $\mathbf{v}$ contains parameters of V we can solve for it using SVD.

From V, we can merge the two scenes together shown in Figure 6.17. Notice the axis are similar and relative distances are low. A mean value can be taken between corresponding points to get a single 3D point.



(a)



(b)

Figure 6.15: *Showing matched features between images of each set (a) and (b).*

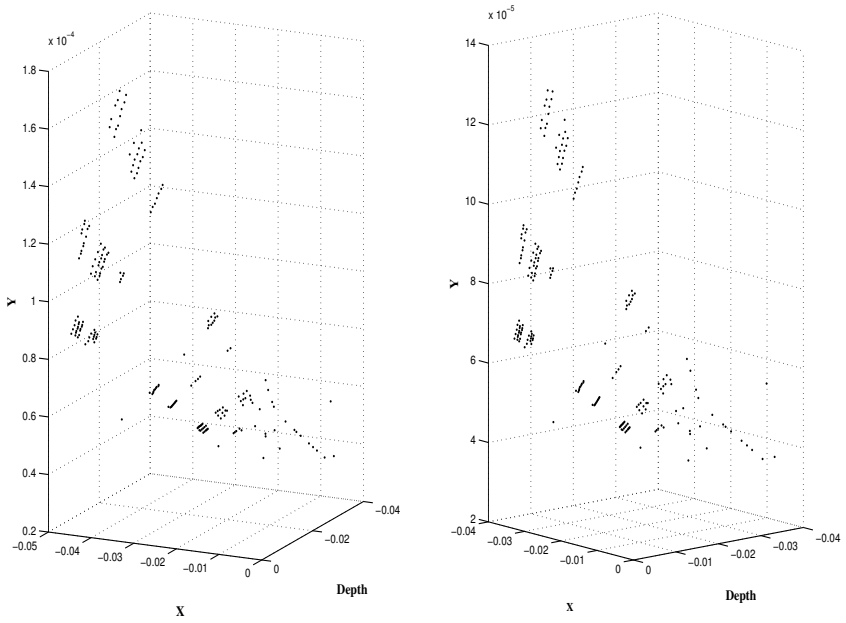


Figure 6.16: *Scene reconstruction from each set.*

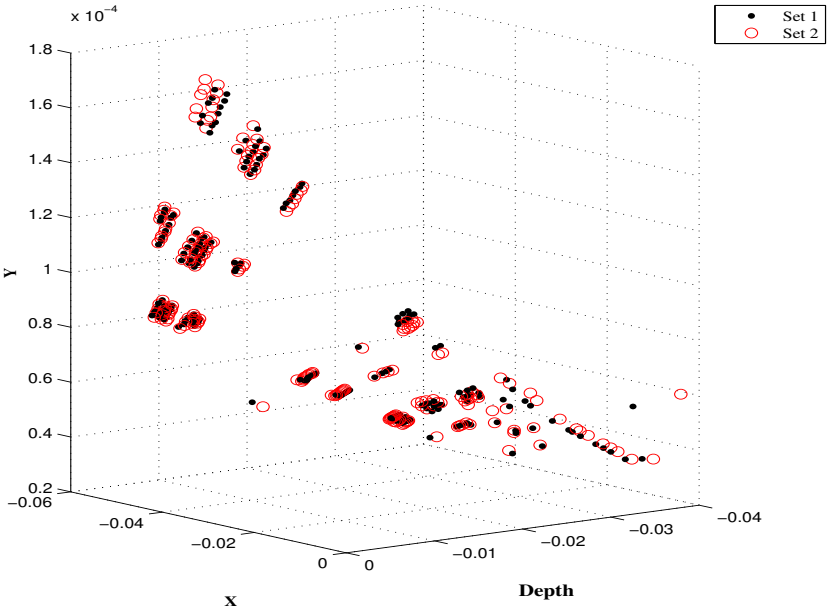


Figure 6.17: *Merged scene from two trifocal tensor based reconstructions. Red is set 1, black is set 2. Notice same axis.*

6.4 Chapter Conclusion

In this chapter, we have presented a couple of tools to assist in cooperative 3D scene reconstruction. Firstly, we have introduced a feature compression technique based on PCA which compresses descriptors to be transferred over a communication link. These can then be retrieved up to a required variance and matched. We have shown that for up to 95 % variance retention the descriptors perform similarly to the original descriptors but with an average 31 % lesser communication cost. We have also developed an iterative technique to determine the number of principal components required to compress the data.

Secondly, we have introduced a technique to merge two 3D scene reconstructions together. This is based on the trifocal tensor which reconstructs 3 images at a time. Considering three platforms taking one image at a time, we develop a technique to match between two time stamps of images, reconstruct them and then finally merge them together.

These methods show promising results but require further investigation. Moreover, the techniques presented here form a fascinating avenue for future work.

7
Conclusions & Future Work

7.1 Summary

This thesis has investigated robust and fast methods for 2D and 3D image mosaicing to increase information content of images. Mosaicing is underlined by the process of image registration which relates one image to another overlapping image. A significant portion of the contributions of this thesis are devoted to solving the image registration problem from image feature based methods. We contribute to two main aspects of image registration, namely feature description and matching and robust estimation of inter-image transformations. Another avenue of work included in the thesis, is i n the 3D domain. This includes, an IMU assisted feature matching technique and semi-dense projective 3D scene reconstruction using homographic lines. Additionally, we have introduced novel methods for cooperative image mosaicing. These, we set as a subject for extension of the work of this thesis.

Feature Description and Matching - Chapter 3

After identifying image feature based methods as an ideal solution for image registration, we have contributed to improve feature description and matching. Firstly we have examined two feature description techniques, namely intensity vectors and distribution based descriptors. From these, the distribution based descriptor performs the best. They are more robust to changes noise, though less distinctive. To overcome this issue, we have developed a novel two signature descriptor. It

combines the robustness of histograms with the distinctive power of colour. Experiments show an average of 20% increase in precision compared to the SIFT like descriptor.

We have also invented a novel feature clustering technique based on six colour codes that allows precise and fast feature matching by reducing the search region for locating candidate matches. Example tests on real world images show a 5 times increase in matching speed.

Robust Homography Estimation - Chapter 4

To estimate inter-image homography $\mathcal{H}$ between the images, we have introduced robust filter based techniques that take into account feature location uncertainty. They also allow us to track changing parameters of $\mathcal{H}$. These filters can be used in real time cooperative image mosaicing, where optimal transformation between mobile platforms is likely to change due to motion.

We have developed a novel way to reduce feature location uncertainty by applying information fusion on features from three channels of an RGB image. This is done using covariance intersection and is shown to reduce error covariances of features and give better results for the inter-image homography of real world overlapping images.

The error location information is recast into a new H_∞ filter exploiting the L_∞ norm that takes into account system modelling uncertainty to estimate $\mathcal{H}$. Results from numerous examples show that this filter is capable of accurately estimating geometric and photometric transformations between images. Furthermore, it outperforms, in terms of results and computational cost, the expensive non-linear optimisation techniques, i.e., Levenberg-Marquardt.

Rapid 3D Scene Reconstruction - Chapter 5

An IMU assisted feature matching technique is developed that isolates the area of overlap of camera views and projects them on to the image plane. This reduces the search region in which to look for candidate matches resulting in less false matches

and faster matching of features. This technique is a good tool in cooperative mosaicing but requires IMU data. Since most autonomous devices are installed with IMUs for navigation, this is not a hindrance.

A semi-dense 3D scene reconstruction algorithm is presented that uses homographic lines to establish pixel correspondences. This is instead of rectifying the images and gives good results. We have also applied Delaunay triangulation on the reconstruction to fill in any empty spaces in the visualisation.

Cooperative Mosaicing Methods - Chapter 6

New methods for cooperative mosaicing are introduced. These include feature description compression using PCA that allows to reduce the communication costs of transferring descriptors from one platform to another yet still maintaining comparable matching precision. Additionally, we have introduced a scenario of cooperative mosaicing in 3-space that registers image between three platforms for three images at a time. To this end, we used the trifocal tensor capable of handling three images at a time giving a broader view of the scene. These methods are to to be investigated further, topics for which are introduced in the next section.

7.2 Suggestions for Future Work

Whilst most of the objectives set out in the beginning of this thesis have been tackled, there are still avenues for future work that can extend the topic of single and cooperative 2D/3D image mosaicing.

We have already introduced a couple of methods in Chapter 6 to be investigated further. An extension of the work in this chapter would be to test trifocal based scene reconstruction including feature communication on ground platforms in real time. Additionally, it will be interesting to see how feature location uncertainty, as tackled for homography estimation in Chapter 4, affects the results.

The homographic line based reconstruction in Chapter 5 can be extended to work with more than two images, therefore giving a bigger reconstruction. Additionally, other meshing techniques based on square or hexagonal meshes can be tested.

It will be interesting to use image mosaics separated in time for change detection. This will include identifying changes in the scene and additionally can be extended to categorise the changes into objects of interests.

Using image mosaics, not only for surveillance, but also as cues for navigation is another interesting extension of this work. When building mosaics, real world positions of objects or the platform can be retained which can be later used to guide a robot surveying the same environment. As an example, this can be done to look for objects of interests.

Frameworks for cooperative behaviour of platforms can also be looked at. For example, how platforms will share information and how it all will be managed. This can be done with either a centralised framework or a decentralised approach where each platform is given limited autonomy.

Although a few of the algorithms included in the thesis are written in C/C++ code it would be ideal to implement all algorithms developed in thesis to this programming language for real time capability.

Bibliography

[1] D. Capel. *Image mosaicing and super-resolution*. PhD thesis, University of Oxford, 2001.

[2] http://www.johnparfrey.co.uk/2013/03/fisheye-photography/.

[3] K. Schutte and A. Vossepoel. Accurate mosaicing of scanned maps, or how to generate a virtual A0 scanner. In *In Annual Conference for the Advance School of Computing*, pages 353–359, 1995.

[4] P. Cheeseman, B. Kanefsky, R. Kraft, and J. Stutz. Super-resolved surface reconstruction from multiple images. Technical report, NASA, 1994.

[5] R. Kumar, P .Anandan, M. Irani, J. Bergen, and K. Hanna. Representation of scenes from collection of images. In *ICCV Workshop on the Representation of Visual Scenes*, pages 10–17, June 1995.

[6] S. Mann and R.W. Picard. Video orbits of the projective group: A new prespective on image mosaicing. Technical report, MIT, 1996.

[7] J.B.A. Maintz and M.A. Viergever. A survey of medical image registration. *Medical Image Analysis*, 2(1):1–36, 1998.

[8] L.G. Brown. A survey of image registration techniques. *ACM Computing Surveys*, 24(4), December 1992.

[9] U. Bhosle, S. Chaudhuri, and S.D. Roy. A fast method for image mosaicing using geometric hashing. *IETE Journal of Research, Special Issue on Visual Media Processing*, pages 317–324, 2002.

[10] M. Heikkila and M. Pietkainen. An image mosaicing module for wife area surviellance. In 3^{rd} *International Workshop on Video Surveillance and Senros Networks*, pages 11–18, 2005.

[11] R. Hartley. Self-calibration from multiple views with a rotating camera. In *European Conference on Computer Vision*, pages 471–478, 1994.

[12] P.H.S. Torr and D.W. Murray. Outlier detection and motion segmentation. Technical report, University of Oxford, 1993.

[13] E. Turkbeyler, C. Harris, and R. Evans. Building aerial mosaics for visual MTI. In *Unmanned/Unattended Senros and Sensor Network*, October 2008.

[14] K.J. Hanna. Direct multi-resolution estimation of ego-motion and structure from motion. In *IEEE Workshop on Visual Motion*, pages 156–162, 1991.

[15] Qifa Ke and T. Kanade. Transforming camera geometry to a virtual downward-looking camera: robust ego-motion estimation and ground-layer detection. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 390–397, 2003.

[16] S.A. Krim, O.A.A. Elaziz, and R. Chatila. A computationally efficient EKF-vSLAM. In 16^{th} *Mediterranean Conference on Control and Automation*, pages 511–516, June 2008.

[17] N. Karlsson, E. di Bernardo, J. Ostrowski, L. Goncalves, P. Pirjanian, and M.E. Munich. The vSLAM algorithm for robust localization and mapping. In *IEEE Conference on Robotics and Automation*, pages 24–29, April, 2005.

[18] D. Krys and H. Najjaran. Development of visual simultaneous localization and mapping (VSLAM) for a pipe inspection robot. In *International Symposium on Computational Intelligence and Robotics*, pages 344–349, June 2007.

[19] J.K. Kamarainen and P. Paalanen. Experimental study on fast 2D homography estimation from a few point correspondences. Technical report, Lappeenranta University of Technology, 2009.

[20] R. Zhao, L. Ma, and D. Yang. Image registration based on convex hull RanSac like. In *International Symposium on Computer Network and Multimedia Technology*, pages 1–4, 2009.

[21] F. Caballero, L. Merino, J. Ferruz, and A. Ollero. Homography based kalman filter for mosaic bulding. applications to UAV position estimation. *IEEE International Conference on Robotics and Automation*, pages 2004–2009, April 2007.

[22] C.R. Viala and A. Salmeron. An iterative kalman filter approach to camera callibration. *Advanved Concepts for Intelligent Vision Systems*, 5259:137–146, 2008.

[23] C. Morimoto and R. Chellappa. Fast 3d stabilisation and mosaic construction. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 660–665, June 1997.

[24] R. Hartley and A. Zisserman. *Multiple view geometry*. Cambridge University Press, 2^{nd} edition, 2003.

[25] D.J. Holtkamp ad A.A. Goshtaby. Precision registration and mosaicing of multi-camera images. *IEEE Transactions on Geoscience and Remote Sensing*, 47(10):3446–3455, 2009.

[26] M. Mizotin, G. Krivovyaz, A. Velizhev, A. Chernyavskiy, and A. Sechin. Robust matching of aerial images with low overlap. In *Conference on Photogrammetric Computer Vision and Image Analysis*, pages 3–18, September 2010.

[27] Y. Caspi and M. Irani. Alignment of non-overlapping sequences. In 8^{th} *IEEE International Conference on Computer Vision*, volume 2, pages 76–83, June-July 2001.

[28] B.E. Pires and P. Aguiar. Registration of images with small overlap. In 6^{th} *Workshop on Multimedia Signal Processing*, pages 255–258, October 2004.

[29] P.E. Anuta. Spatial registration of multispectral and multitemporal digital imagery using fast Fourier transform. *IEEE Transactions on Geoscience Electronics*, 8(4):353–367, 1970.

[30] O. Samritjiarapon and O. Chitsobhuk. An FFT-based technique and best-first search for image registration. In *International Symposium on Communications and Information Technologies*, pages 364–367, 2008.

[31] B.S. Reddy and B.N. Chatterji. An fft-based technique for translation, rotation and scale-invariant image registration. *IEEE Transactions on Image Processing*, 5(8):1266–1271, 1996.

[32] J. Davis. Mosaics of scenes with moving objects. In *Computer Vision and Pattern Recognition*, pages 354–360, June 1998.

[33] A. Geiger. Monocular road mosaicing for urban environments. In *IEEE Intelligent Vehicles Symposium*, pages 140–145, June 2009.

[34] T. Botterill, S. Mills, and R. Green. Speeded-up bag of words algorithm for robot localisaiton through scene recognition. In *IEEE Image and Vision Computing*, pages 1–6, 2008.

[35] Y. Ning, R. Chen, and P. Xu. Wide baseline image mosaicing by integrating mser and hessian-affine. In 4^{th} *International Congress on Image and Signal Processing*, pages 2034–2037, 2011.

[36] C.C.D. Santos, S.A. Stoeter, P.E. Rybski, and N.P. Papanikolopoulos. Mosaicing images: Panoramic imaging for miniature robots. *IEEE Robotics and Automation Magazine*, pages 62–68, 2004.

[37] M. Saptharishi, C.S. Oliver, C.P. Diehl, K.S. Bhat, J.M. Dolan, A.T. Ollennu, and P.K. Khosla. Distributed surveillance and reconaissance using multiple autonoumous ATVs: Cyberscout. *IEEE Robotics and Automation Magazine*, 18(5):825–836, 2002.

[38] N. Gracias, S. Zwaan, A. Bernardino, and J.S. Victor. Mosaic based navigation for autonomous underwater vehicles. *IEEE Journal of Ocean Engineering*, 28(4):609–624, 2003.

[39] Y. Ping, M. Zheng, G. Anjie, and Q. Feng. Video image mosaics in real-time based on SIFT. In *Conference on Pervasive Computing Signal Processing and Applications*, pages 879–882, September 2010.

[40] M. Kourogi, T. Kurata, J. Hoshino, and Y. Muraoka. Real-time image mosaicing from a video sequence. In *International Conference on Image Processing*, pages 133–137, October 1999.

[41] Z. Zhu and A.R. Hanson. Mosaic based 3D scene representation and rendering. In *IEEE International Conference on Image Processing*, pages 633–636, September 2005.

[42] H.Y. Shum and M. Han. Interactive construction of 3D models from panoramic mosaics. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 427–433, June 1998.

[43] F. Li, J. Barabas, and A.L. Santos. Information processing for live photo mosaic with a group of wireless image sensors. In 9^{th} *ACM/IEEE International Conference on Information Processing in Sensor Networks*, pages 58–69, 2010.

[44] B. Zhou, H. Zhou, and J. Zheng. High-tree image mosaicing and seam processing for precision pesticide application. In *New Technology of Agricultural Engineering*, pages 86–90, 2011.

[45] S.E. Chen. Quicktime VR - an image based approach to virtual environment navigation. In *Conference on Computer Graphics*, pages 29–38, 1995.

[46] S.N. Sinha, M. Polleyfeys, and S.K. Kim. Highresolution and multiscale panoramic mosaics from pan-tiltzoom cameras. In 4^{th} *Indian Conference on Computer Vision, Graphics and Image Processing*, 2004.

[47] S. Gumustekin. Mosaic image generation on a flattened Gaussian sphere. In *IEEE Workshop on Applications of Computer Vision*, pages 50–55, December 1996.

[48] A. Roman and H. Lensch. Automatic multispectral images. In *Eurographics Symposium on Rendering*, 2006.

[49] E. Zhang, R. Raguram, and P.F. Georgel. Efficient generation of multiperspective panoramas. In *International Conference on 3D Imaging, Modelling, Processing, Visualisation and Transmission*, pages 86–92, 2011.

[50] D.N. Wood, A. Finkelstein, J.F. Hughes, C.E. Thayer, and D.H. Salesin. Multiperspective panoramas for Cel animation. In 24^{th} *Annual Conference on Computer Graphcis and Interactive Techniques*, pages 243–250, 1997.

[51] U. Bhosle, S. Dutta, and S. Chaudhuri. Multispectral panoramic mosaicing. *Pattern Recognition Letters*, 26(4):471–482, 2005.

[52] R.T. Collins, A.J. Lipton, H. Fujiyoshi, and T. Kanade. Algorithms for cooperaitve multisensor surveillance. *Proceedings of the IEEE*, 89(10):1456–1477, 2001.

[53] D. Rodriguez. *Single and multiple stereo view navigation for planetary rovers*. PhD thesis, Cranfield University, 2013.

[54] D.A. Forsyth and J. Ponce. *Computer vision a modern approach*. Prentice Hall PTR, 1^{st} edition, 2003.

[55] Richard Szeliski. *Computer Vision: Algorithms and Applications*. Springer, 2011.

[56] B. Zitova and J. Flusser. Image registration methods: a survey. *Image and Vision Computing*, 21:977–1000, 2003.

[57] D.G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[58] M.V. Wyawahare, P.M. Dr. Patil, and H.K. Abhyankar. Image registration techniques: An overview. *International Journal of Signal Processing, Image Processing and Pattern Recognition*, 2(3), September 2009.

[59] C. Harris and M. Stephens. A combined corner and edge detector. In *Alvey Vision Conference*, volume 15, pages 147–151. Manchester, UK, 1988.

[60] Z. Zheng, H. Wang, and E.K. Teoh. Analysis of gray level corner detection. *Pattern Recognition Letters*, 20:149–162, 1999.

[61] H. Bay, A. Ess, T. Tuytelaars, and L.V. Gool. Speeded up robust features. *Computer Vision and Image Understanding*, 110(3):346–359, 2008.

[62] T. Lindeberg. Feature detection with automatic scale selection. *International Journal of Computer Vision*, 3:79–116, 1998.

[63] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, and T. Kadir. A comparison of affine region detectors. *International Journal of Computer Vision*, 65(1-2):43–72, 2006.

[64] E. Rosten and T. Drummond. Machine learning for high-speed corner detection. In *European Conference on Computer Vision*, volume 1, pages 430–443, 2006.

[65] D. Rodriguez and N. Aouf. Robust Harris-Surf features for robotic vision based navigation. In *International Conference on Intelligent Transportation*, pages 1160–1165, 2010.

[66] H. Yan and J.G. Lio. Robust phase correlation based feature matching for image co-registration and DEM generation. In 5^{th} *Conference on Systems Engineering for Autonomous Systems Defence Technology Centre*, 2008.

[67] G. Carneiro and A.D. Jepson. Phase-based local features. In 7^{th} *European Conference on Computer Vision*, pages 282–296, 2002.

[68] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *Pattern Analysis and Machine Intelligence*, 27(10):1615–1630, 2005.

[69] R. Zabih and J. Woodfill. Non-parameteric local transforms for computing visual correspondence. In 3^{rd} *European Conference on Computer Vision*, pages 151–158, 1994.

[70] M.J. Black and P.Anandan. The robust estimation of multiple motions. *Computer Vision and Image Understanding*, 63(1):75–104, January 1996.

[71] C. Schmid, R. Mohr, and C. Bauckhage. Evaluation of interest point detectors. *International Journal of Computer Vision*, 37:151–172, 2000.

[72] H. Moravec. Obstacle avoidance and navigation in the real wold by a seeing robot rover. In *tech. report CMU-RI-TR-80-03, Robotics Institute, Carnegie Mellon University and doctoral dissertation, Stanford Universitym*, September 1980.

[73] Visual geometry group. www.robots.ox.ac.uk/ vgg/data/data-mview.html.

[74] Z. Zhang, R. Deriche, O. Faugeras, and Q.T. Luoung. A robust technique for matching two uncalibrated images through the recovery of the unkown epipolar geometry. *Artificial Intelligence*, 78:87–119, 1995.

[75] F. Tombari, S. Salti, and L. Di Stefano. Unique signatures of histograms for local surface description. In 11^{th} *European Conference on Computer Vision (ECCV)*, pages 356–369, September 2010.

[76] F. Tombari, S. Salti, and L. Di Stefano. A combined texture-shape descriptor for enhanced 3D feature matching. In 18^{th} *IEEE International Conference on Image Processing*, pages 825–828, 2011.

[77] L. Hiatao. Enhanced point descriptors. In *International Conference on Audio Language and Image Processing*, pages 677–681, 2010.

[78] M. Mignotte. Segmentation by fusion of histogram based K-means clusters in different colour spaces. *IEEE Transactions on Image Processing*, 17(5): 780–787, 2008.

[79] G.J. Burghouts and J.M. Geusebroek. Performance evaluation of local colour invariants. *Journal of Computer Vision and Image Understanding*, 113(1): 48–62, 2009.

[80] D. Nister and H. Stewenius. Scalable recogniton with a vocabulary tree. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, pages 2161–2168, 2006.

[81] T. Botterill, S. Mills, and R. Green. Real-time aerial image mosaicing. In 25^{th} *International Conference of Image and Vision Computing*, pages 1–8, 2010.

[82] L. Ljung. *System identification: Theory for the user*. Prentice Hall PTR, 2^{nd} edition edition, 1999.

[83] D. Simon. *Optimal state estimation: Kalman, H_∞ and nonlinear approaches*. John Wiley and Sons, 2006.

[84] R. Hartley. In defence of the 8-point algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(6):580–593, 1997.

[85] Y. Kanazawa and K. Kantani. Do we really have to consider covariance matrices for image features? In 8^{th} *IEEE Internatial Conference on Computer Vision*, volume 2, pages 301–306, 2001.

[86] M.J. Brooks, W. Chojnacki, D. Gawley, and A. van den Hengel. What value covariance information in establishing vision parameters? In *IEEE International Conference on Computer Vision*, July 2001.

[87] B. Zeisl, P. Georgel, F. Schweiger, E. Steinbach, and N. Navab. Estimation of location uncertainty for scale invariant feature points. In *British Machine Vision Conference*, pages 1–12, September 2009.

[88] W. Niehsen. Information fusion based on fast covariance intersection filtering. In 5^{th} *International Conference on Information Fusion*, volume 2, pages 901–902, July 2002.

[89] S.J. Julier and J.K. Uhlmann. *Handbook of Data Fusion*. CRC Press, Boca Raton, FL, USA, 2001.

[90] H. Lee, Y. SEO, and R. Hartley. Homography estimation with L_∞ norm minimisation method. In 14^{th} *Korea-Japan Joint Workshop on Frontiers of Computer Vision (FCV)*, pages 87–91, 2008.

[91] B. Micusik and R. Pflugfelder. Localizing non-overlapping surveillance cameras under the L_∞ norm. In *Conference on Computer Vision and Pattern Recognition*, pages 2895–2901, 2010.

[92] K. Astrom, O. Enquist, C. Olsson, and F. Kahl. An L_∞ approach to structure and motion problems in 1-D motion. In *International Conference on Computer Vision*, pages 676–683, October 2007.

[93] F. Yang, Z. Wang, and Y.S. Hung. Robust Kalman filtering for discrete time varying uncertain systems with multiplicative noises. *IEEE Transactions on Automatic Control*, 47:1179–1183, 2002.

[94] Z. Wang, F. Yang, D.W.C. Ho, and X. Liu. Robust finite horizon filtering for stochastic systems with missing measurements. *IEEE Signal Processing Letters*, 12:437–440, 2005.

[95] S.A. Imran and N. Aouf. Robust L_∞ homography estimation using reduced image feature covariances from an rgb image. *Journal of Electronic Imaging*, 21(4), 2012.

[96] H.W. Sorenson. Least-squares estimation: from Gauss to Kalman. *IEEE Spectrum*, 7:63–68, July 1970.

[97] Abdelkrim Nemra. *Robust airborne 3D visual simultaneous localisation and mapping*. PhD thesis, Cranfield University, 2010.

[98] U. Shaked and Y. Theodor. H_∞ optimial estimation: a tutorial. In 31^{st} *IEEE Conference on Decision and Control*, volume 2, pages 2278–2286, 1992.

[99] S. Agarwal, N. Snavely, S. Seitz, and R. Szeliski. Bundle adjustment in the large. In *European Conference on Computer Vision*, pages 29–42, September 2010.

[100] K. Konolige. Sparse sparse bundle adjustment. In *British Machine Vision Conference*, 2010.

[101] S. Agarwal, Y. Furukawa, N. Snavely, I. Simon, B. Curless, S. Seitz, and R. Szeliski. Building rome in a day. In *IEEE Conference on Computer Vision*, pages 72–79, October 2009.

[102] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? the KITTI vision benchmark suite. In *Computer Vision and Pattern Recognition*, pages 3354–3361, June 2012.

[103] A. Agarwal and R. Chellappa. Ego-motion estimation and 3D model refinement in scenes with varying illumination. In *IEEE Workshop on Motion and Video Computing*, pages 140–146, January 2005.

[104] T. Oskiper, R. Kumar, J. Fields, and S. Samarasekera. Vehicle 3D pose tracking using distributed aperture sensors. In *SPIE Proceedings of Unmanned Systems Technology VIII*, volume 6320, April 2006.

[105] P. Alcantarilla, L. Bergasa, and F. Dellaert. Visual odometry priors for robust EKF-SLAM. In *IEEE Conference on Robotics and Automation*, pages 3501–3506, 2010.

[106] Y. Furukawa and J. Ponce. Dense 3D motion capture for human faces. In *Computer Vision and Pattern Recognition*, pages 1674–1681, June 2009.

[107] L.L. Sik and S. Pattanaik. Fusing geo-referenced images for urban scene. In 15^{th} *International Conference on Information Fusion*, pages 9–16, July 2012.

[108] J.T. Hwang, J.S. Wang, and Y.T. Tsai. 3D modelling and accuracy assessment- a case study of photosynth. In 20^{th} *International Conference on Geoinformatics*, pages 1–6, June, 2012.

[109] W. Xu, L. Deng, Q. Zheng, and S.Frezza. Using stereo vision to construct 3D surface models. *Potentials IEEE*, 31(2):31–37, 2012.

[110] N. Snavely, S. Seitz, and R. Szeliski. Skeletal graphs for efficient structure from motion. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2008.

[111] A. Geiger, J. Ziegler, and C. Stiller. Stereoscan: Dense 3D reconstruction in real-time. In *IEEE Intelligent Vehicles Symposium*, pages 963–968, 2011.

[112] A. Geiger, M. Roser, and R. Urtasun. Efficient large-scale stereo matching. In 10^{th} *Asian Conference on Computer Vision*, pages 25–38, 2010.

[113] P. Gao, X. Guo, D. Yang, M. Shen, Y. Zhao, Z. Xu, Z. Fan, J. Yu, and Y. Ma. 3D model reconstruction from video recorded with a compact camera. In 9_{th} *International Conference on Fuzzy Systems and Knowledge Discovery*, pages 2528–2532, 2012.

[114] D.T. Kien. A review of 3D reconstruction from video sequences. Technical report, University of Amsterdam, 2005.

[115] C. Beall, B.J. Lawrence, V. Ila, and F. dellaert. 3D reconstruction of underwater structures. In *International Conference on Intelligent Robotics and Systems*, pages 4418–4423, October 2010.

[116] M. Langer and S. BenHimane. An interactive vision-based 3D reconstruction workflow for industrial ar applications. In *IEEE International Symposium on Mixed and Augmented Reality*, 2010.

[117] R. Ziegler, W. Matusik, H. Pfister, and L. McMillan. 3D reconstruction using labeled image regions. In *Eurographics Symposium on Geometry Processing*, pages 1–13, June 2003.

[118] Y. Furukawa, B. Curless, S. Seitz, and R. Szeliski. Towards internet-scale multi-view stereo. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1434–1441, June 2010.

[119] N. Snavely, S. Seitz, and R. Szeliski. Photo tourism: exploring photo collectins in 3D. In *The* 33^{rd} *International Conference and Exhibition on Computer Graphics and Interactive Techniques*, volume 25, pages 835–846, 2006.

[120] R. Ferreira, J.P. Costeira, C. Silvestre, I. Sousa, and J.A. Santos. Using stereo image reconstruction to survey scale models of rubble-mound structures. In 1^{st} *International Conference on the Application of Physical Modelling to Port and Coastal Protection*, pages 1–10, 2006.

[121] M. Gerrits and P. Bekaert. Robust stereo aggregtion with large windows. Technical report, Transnationale Universiteit Limburg, 2006.

[122] M. Lhuillier and L. Quan. A quasi-dense approach to surface reconstructio from uncalibrated images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(3):418–433, March 2005.

[123] F. Lafarge, R. Keriven, M. Bredif, and H.H. Vu. A hybrid multi-view stereo algorithm for modeling urban scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(1):5–17, 2013.

[124] C. Baillard and A. Zisserman. Automatic reconstruction of piece-wise planar models from multiple views. In *Computer Vision and Pattern Recognition*, volume 2, June 1999.

[125] C. Baillard, C. Schmid, A. Zisserman, and A. Fitzgibbon. Automatic line matching and 3D reconstruction of buildings from multiple views. In *Conference on Automatic Extraction of GIS Objects from Digital Imagery*, pages 69–80, September 1999.

[126] Y. Furukawa, B. Curless, S. Seitz, and R. Szeliski. Manhattan world stereo. In *Computer Vision and Pattern Recognition*, pages 1422–1429, June 2009.

[127] Y. Furukawa, B. Curless, S. Seitz, and R. Szeliski. Reconstructing building interiors from images. In 12^{th} *International Conference on Computer Vision*, pages 80–87, September 2009.

[128] H.H. Liao, Y. Lin, and G. Medioni. Aerial 3D reconstruction with line-constrained dynamic programming. In *IEEE Conference on Computer Vision*, pages 1855–1862, 2011.

[129] C. Fruh and A. Zakhor. An automated method for large-scale ground-based city model acquisition. *International Journal of Computer Vision*, 60(1): 5–24, 2004.

[130] A. Banno, T. Masuda, T. Oishi, and K. Ikeuchi. Flying laser range sensor for large-scale site-modeling and its application in bayon digital archival project. *International Journal of Computer Vision*, 78(2-3):207–222, 2008.

[131] N. Ho and R. Jarvis. Large scale 3D environmental modelling for stereoscopic walk-through visualisation. In *3DTV Conference*, pages 1–4, May 2007.

[132] C. Fruh, S. Jain, and A. Zakhor. Data processing algorithms for generating textured 3D building facade meshes from laser scans and camera images. *International Journal of Computer Vision*, 62(2):159–184, 2005.

[133] T. Nicosevici and R. Garcia. Online robust 3D mapping using structure from motion cues. In *IEEE Kobe Techno Ocean*, pages 1–7, April 2008.

[134] I. Esteban, J. Dijk, and F. Groen. Automatic 3D modeling of the urban landscape. In *International Congress on Ultra Modern Telecommunications and Control Systems and Workshops*, pages 421–428, 2010.

[135] A. Varol, A. Shaji, M. Salzmann, and P. Fua. Monocular 3D reconstruction of locally textured surfaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(6):1118–1130, 2012.

[136] A. Laurentini. The visual hull concept for visual-based image understanding. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16:150–162, 1994.

[137] E.H. Zhang and Y.W. Zhao. A novel 3D human action reconstruction method. In *International Conference on Machine Learning and Cybernetics*, pages 1633–1636, July 2011.

[138] K. Kolev and D. Cremers. Integration of multiview stereo and silhouettes via convex functionals on convex domains. In *Lecture Notes in computer Science*, volume 5302, pages 752–765, 2008.

[139] H. Kim, J.Y. Guillermaut, T. Takai, and M. Sarim. Outdoor dynamic 3D scene reconstruction. *IEEE Transactions on Circuits and Systems for Video Technology*, 22(11):1611–1622, November 2012.

[140] T. Duckworth and D.J. Roberts. Camera image synchronisation in multiple camera real-time 3D reconstruction of moving humans. In 15^{th} *IEEE/ACM International Symposium on Distributed Simulation and Real Time Applications*, pages 138–144, September 2011.

[141] O.J. Woodman. An introduction to inertial navigation. Technical Report 696, Cambridge University, August 2007.

[142] P. Cignoni, C. Montani, and R. Scopigno. Dewall: A fast devide and conquer delaunay triangulation algorithm in ed. *Computer-Aided Design*, 30(5):333–341, April 1998.

[143] S. Oudot. Delaunay triangulation. Technical report, Stanford University, Palo Alto, 2011.

[144] D. Chen, A. Ravindran, and P. Vishwabrahamanasaraf. Reverse engineering closely-spaced free-form shapes for a fabric-over-body model. *Engineering*, 3 (10):1022–1029, 2011.

[145] F. Preparata and M. Shamos. *Computational geometry: an introduction*. Springer Verlag, 1985.

[146] F. Aurenhammer. Voronoi diagrams-a survey of fundamental data structure. *ACM Computing Surveys*, 23(3):345–405, September 1991.

[147] I. Jolliffe. *Principal component analysis*. John Wiley and Sons, 2005.

[148] S. Wold, K. Esbensen, and P. Geladi. Principal component analysis. *Chemometrics and Intelligent Laboratory Systems*, 2:37–52, 1987.

[149] J. Li. The trifocal tensor and its application in augmented reality. Master's thesis, University of Ottawa, September 2005.

[150] M. I A Lourakis and A.A. Argyros. Is Levenberg-Marquardt the most efficient optimization algorithm for implementing bundle adjustment? In 10^{th} *IEEE International Conference on Computer Vission. ICCV 2005.*, volume 2, pages 1526–1531, October 2005.

[151] D. Marquardt. An algorithm for the least-squares estimation of nonlinear parameters. *SIAM Journal of Applied Mathematics*, 11(2):431–441, 1963.

[152] H. Nielsen. Damping parameter in marquardt's method. Technical report, Technical University of Denmark, 1999.

A
Non-Linear Optimisation

A.1 Levenberg-Marquardt Optimisation

Levenberg-Marquardt (LM) is an iterative technique that finds the local minima of a function that is expressed as the sum of squares of several non-linear, real-valued functions. It is now a standard technique for nonlinear least-squares problems, widely adopted for dealing with data fitting applications. LM can be thought of as a combination of steepest descent and the Gauss-Newton method. When the solution is far from a local minimum, the algorithm behaves like a steepest descent method: slow, but guaranteed to converge. When the current solution is close to a local minimum, it becomes a Gauss-Newton method and exhibits fast convergence [150, 151].

Let ζ be a function that maps a parameter vector $\tau \, \epsilon \, \mathbb{R}^m$ to a measurement vector $\hat{\pi} = \zeta(\tau)$, $\hat{\pi} \, \epsilon \mathbb{R}^n$. Initial estimates for τ_0 and π are provided and the vector τ^* that best satisfies ζ locally, i.e., minimises the squared dtistance $\varepsilon^T \varepsilon$ with $\varepsilon = \pi - \hat{\pi}$ for all τ. The basis of the LM algorithm is a linear approximation of ζ in the neighbourhood of τ. Denoting by J the Jacobian matrix $\frac{\partial \zeta(\tau)}{\partial \tau}$, a Taylor series expansion for a small $|||\delta_\tau|$ gives,

$$\zeta(\tau + \delta_\tau) \approx \zeta(\tau) + \mathrm{J}\delta_\tau. \tag{A.1}$$

LM is iterative. Initiated at τ_0, it produces a series of vectors that converge to a local minimiser τ^* of ζ. Hence, at each iteration, it is required to find the step δ_τ that minimises the quantity

$$||\pi - \zeta(\tau + \delta_\tau)|| \approx ||\pi - \zeta(\tau) + \mathrm{J}\delta_\tau|| = ||\varepsilon - \mathrm{J}\delta_\tau|| \tag{A.2}$$

The wanted δ_p is the solution to a linear least squares problem: the minimum is attained when $\mathrm{J}\delta_\tau - \varepsilon$ is orthogonal to the column space of J. This yields the δ_τ as the solution of the normal equations:

$$\mathrm{J}^T\mathrm{J}\delta_p = \mathrm{J}^T\varepsilon. \tag{A.3}$$

The LM method actually solves a variation of the above equation,

$$N\delta_\tau = \mathrm{J}^T\varepsilon, with N \equiv \mathrm{J}^T\mathrm{J} + \mu\mathrm{I} \text{ and } \mu > 0, \tag{A.4}$$

where I is the identity matrix. The strategy of altering the diagonal elements of $\mathrm{J}^T\mathrm{J}$ is called damping and μ refers to the damping parameter. It allows LM to alternate between a slow descent approach when it is far from the minimum by increasing μ and a fast, quadratic convergence when being near the minimum's neighbourhood by decreasing it. In each iteration of LM μ is adjusted as to achieve the best possible update δ_τ. An efficient update strategy for updating the damping step is given in [152].

Following we have included a brief description of the LM algorithm, Algorithm A.1.

Algorithm A.1: Pseudo code for Levenberg-Marquardt

input : A vector function $\zeta : \mathbb{R}^m \rightarrow \mathbb{R}^n$ with $n \geq m$, a measurement vector $\pi \epsilon \mathbb{R}^n$ and an initial parameters estimate $\tau \epsilon \mathbb{R}^m$

output: A vector $\tau^* \epsilon \mathbb{R}^m$ minimising $||\pi - \zeta(\tau)||$

algorithm:

i:= 0; v:= 2; τ:= τ_0 ;

A:= $J^T J$; $\varepsilon_\tau := \pi - \zeta(\tau)$; g:= $J^T \varepsilon_\tau$;

stop:= ;

while *not stop and* $i < i_{max}$ **do**

  i:= i+1;

  repeat

    Solve $(A + \mu I)\, \delta_\tau = g$;

    if $||\delta_\tau|| \leq \varepsilon_2 ||\tau||$ **then**

      stop:= true;

    else

      $\tau_{new} := \tau + \delta_{tau}$;

      $\rho := (||\varepsilon_\tau||^2 - ||\pi - \zeta(\tau_{new})||^2) \,/\, \left(\delta_\tau^T (\mu \delta_{p+g})\right)$;

      if ρ *¿ 0;*

      then $\tau = \tau_{new}$;

      A:= $J^T J$; $\varepsilon_\tau := \pi - \zeta(\tau)$; g:= $J^T \varepsilon_\tau$;

      stop:= $(||g||_\infty \leq \varepsilon_1)$;

      $\mu := \mu \times max\left(\frac{1}{3}, 1 - (2\rho - 1)^3\right)$; v:= 2 ;

      ;

      else ;

      ;

      $\mu := \mu \times$ v; v:= 2×v ;

  until *p¿0 or (stop)*;

$\tau^* := \tau$;

B
Exploiting Trilinear Relations

As an example we extract a trilinearity from Equation (6.22) given in Chapter 6, which can be used to estimate the trifocal tensor. Equation (6.22) is reproduced in tensor as follows

$$x^i x^{'j} x^{''k} \epsilon_{jqs} \epsilon_{krt} T_i^{qr} = 0 \tag{B.1}$$

or re-written as

$$x^i \left(x^{'j} \epsilon_{jqs} \right) \left(x^{''k} \epsilon_{krt} \right) T_i^{qr} = 0. \tag{B.2}$$

Please refer to [24] for a tutorial on tensor notation.

The above equation is a trilinear relation. "Tri" since every monomial involves a coordinate from each of the three image elements involved. We get 9 trilinearities from Equation (B.2) from three choices of s and t but only four of these are linearly independent and can be used to estimate the trifocal tensor.

For example, choosing two values each for s and t as 1 and 2, we get 4 trilinear relations. For $s = 1$, expanding $\left(x^{'j} \epsilon_{jqs} \right)$ results in

$$l_q^{'} = x^{'j} \epsilon_{jqs} = \sum_j \sum_q \sum_1 \epsilon_{jqs} x^{'j} = \left(0, -x^{'3}, x^{'2} \right) \tag{B.3}$$

where $j, q = 1, 2$ and 3. This is a horizontal line in the second view through $\mathbf{x}^{'}$. The above expression is obtained from the Levi-Civita permutation [24]. Similarly, choosing $t = 2$ in the third view results in a vertical line through $\mathbf{x}^{''}$

$$l_r^{''} = x^{''k}\epsilon_{krt} = \sum_k \sum_r \sum_2 \epsilon_{jqs} x^{'j} = \left(x^{''3}, 0, -x^{''1}\right) \tag{B.4}$$

.

The trilinear relation Equation (B.2) now can be re-written as

$$x^i l_q^{'} l_r^{''} T_i^{qr} = 0. \tag{B.5}$$

Expanding for all values of q and r, i.e, 1, 2 and 3 leads to

$$x^i \sum_{qr} l_q^{'} l_r^{''} T_i^{qr} = 0 \tag{B.6}$$

where l_a is the a^{th} value of l. Substituting values and rewriting

$$\begin{aligned} x^i \left(0 + 0 + 0 - x^{'3}x^{''3}T_i^{21} + x^{'3}x^{''1}T_i^{23} + x^{'2}x^{''3}T_i^{31} + 0 - x^{'2}x^{''1}T_i^{33}\right) = 0 \\ x^i \left(-x^{'3}x^{''3}T_i^{21} + x^{'3}x^{''1}T_i^{23} + x^{'2}x^{''3}T_i^{31} - x^{'2}x^{''1}T_i^{33}\right) = 0. \end{aligned} \tag{B.7}$$

This expression is a trilinearity, one of nine possible trilinearities. Using four such expressions, we can estimate the trifocal tensor.