PROJECTIVE RECONSTRUCTION AND METRIC MODELS FROM UNCALIBRATED VIDEO SEQUENCES

A thesis submitted to the University of Manchester
for the degree of Doctor of Philosophy
in the Faculty of Science and Engineering

August 2001

By
Daniel T. Oram
Department of Computer Science

ProQuest LLC.
789 East Eisenhower Parkway
P.O. Box 1346
Ann Arbor, MI 48106 – 1346

Contents

List of Figures

Abstract

Over recent years, the possibility of reconstructing three dimensional models from images has led to a very active research field. This has recently culminated in systems that are capable of largely automated reconstruction even when nothing at all is known about the cameras or scene except the images themselves. To achieve this has required the combination of computer vision techniques and statistical methods with extensive development of the geometry of multiple views.

Whilst much of the theory underlying the process is very well understood, there still remain numerous problems associated with automated reconstruction. This thesis attempts to improve the state of the art systems by refining and adding to the existing techniques.

The approach involved work on many different aspects of the problem. Firstly, to resolve problems with point matching and accuracy, a new matching approach utilising video sequences was developed. This includes methods to enable successful tracking of features despite occlusion or degenerate motions and a method of selecting frames from video sequences so as to avoid degeneracy and maximise the accuracy with which geometry can be determined.

This new tracking scheme is also integrated with a new and more general projective reconstruction algorithm. This includes new robust methods for projective reconstruction which result in sometimes dramatic increases to speed, accuracy and flexibility of the reconstruction process. These new algorithms are extensively compared to existing projective reconstruction algorithms using both real and synthetic data.

Finally, all the techniques are combined with existing state of the art methods and other more minor new algorithms to create a completely automated reconstruction system that produces texture mapped models of scenes from video sequences and involves absolutely no user intervention or calibration at all.

Declaration

No portion of the work referred to in this thesis has been submitted in support of an application for another degree or qualification of this or any other university or other institution of learning.

Copyright

Notation

Throughout this text, capital letters will refer to matrices e.g. P, and bold letters will refer to column vectors e.g. $\mathbf{x}$ or $\mathbf{X}$. Unless otherwise indicated subscripts on vectors such as $\mathbf{a}_n$ indicate the nth item in that vector.

- $\wedge$ Is the usual euclidean intersection operator, for example $\mathbf{x} \wedge \mathbf{x}'$ indicates the intersection of the lines $\mathbf{x}$ and $\mathbf{x}'$
- $\mathcal{R}^n$ Represents Euclidean n-space.
- $\mathcal{P}^n$ Symbolises projective n-space.
- M^* Indicates the matrix of co-factors of M.
- $\simeq$ Used to indicate equivalence subject to a non-zero scale factor.
- $[\mathbf{t}]_\times$ The antisymmetric matrix defined by $\mathbf{t}$ such that $[\mathbf{t}]_\times \mathbf{x} = \mathbf{t} \times \mathbf{x}$.
- $d_e(\mathbf{x}, \mathbf{y})$ Is defined as the euclidean distance between the two points $\mathbf{x}$ and $\mathbf{y}$. To be more exact, for n dimensional points $\mathbf{x}$ and $\mathbf{y}$ given in homogenous notation as $n+1$ vectors this gives
$$d_e(\mathbf{x}, \mathbf{y}) = \sum_{k=1}^{n} \left(\frac{X_k}{X_{n+1}} - \frac{Y_k}{Y_{n+1}} \right)^2$$
- $\mathbf{x}_j^i$ In the context of multiple views this will indicate item j in image i.
- F_{ij} Indicates the fundamental matrix from image 1 to image 2. This is defined such that $\mathbf{x}_2^T F_{12} \mathbf{x}_1 = 0$, for image 1 point $\mathbf{x}_1$ and image 2 point $\mathbf{x}_2$.
- $\mathbf{e}_{ij}$ In the case of $\mathbf{e}$ representing an epipole this indicates the image of the centre of camera j in camera i.
- T_{ijk} For T representing the trifocal tensor.

- A^+ Is the pseudo inverse of the $n \times m$ matrix, defined such that $AA^+ = I_{n \times n}$. See section 3.1.1 on page 44 for more details.
- $\|\mathbf{a}\|$ Indicates the euclidean norm of the vector $\mathbf{a}$. More specificaly if $\mathbf{a}$ has n items then $\|\mathbf{a}\| = \sqrt{\sum_{i=1}^{n} |a_i|^2}$

Chapter 1

Introduction

Understanding the three dimensional structure in a scene from just images of that scene has been a long-standing area of research in both computer vision and geometry. The very first directly related work was the specification of the perspective relationship between images and the scene that they capture. This was first specified as far back as the beginning of the 15th century by the Italian architect Filippo Brunelleschi and then formalised in the same year by Leon Battista Alberti. It was immediately applied by the painters of the Italian renaissance. Over time, the initial description was developed into a complete branch of geometry - projective geometry. This geometry has been specifically designed to model what information a projection (for example, an image) captures.

In the 20th century, measurement from images has proved to be of great interest. Initial efforts focused on industrial inspection and robotics type applications where accuracy is of paramount importance. This very restricted application domain resulted in systems using expensive devices to take very accurate measurements in tightly controlled conditions and of tightly controlled scenes.

More recently, the rise of the personal computer has led to a huge expansion in the numbers of three dimensional computer graphics and multimedia applications. There has been a correspondingly large rise in the need to quickly, easily and cheaply produce realistic 3D models for these applications. Subsequent research into three dimensional reconstruction has focused more on solving these problems and has led to efforts toward automated reconstruction using nothing more than conventional video cameras.

Initial attempts at this automated reconstruction used careful calibration to determine the parameters of the camera such as focal length and centre. Since this was restrictive, later researchers have applied the tools of projective geometry to produce reconstructions

subject to an arbitrary projective transformation instead of a more conventional Euclidean one. A projective transformation is a much more general transformation than a Euclidean one, and so the calibration details of the camera such as its focal length and centre can be absorbed into the reconstruction.

With the emergence of self-calibration techniques that could convert these reconstructions into a more conventional Euclidean form, the techniques were in place to completely automate reconstruction from images. Whilst the basic techniques and theory have existed since the mid 1990's, the problem is still far from being practically solved for the general case. Self-calibration is still unreliable except when some details about camera motion or scene structure are used, and the problem of automated feature tracking, although solved for video sequences, is always problematic. There is also room for improvement throughout all stages of the process, where existing techniques are giving good (but rarely ideal) results.

The work in this thesis sets out to overcome some of the remaining problems and to refine existing solutions. In particular, the main problems addressed and refined are feature tracking, projective reconstruction and rectification, with the ultimate aim of taking totally automated reconstruction out of the lab and making it practical.

1.1 3D Structure Capture: Existing Methods

As would be expected, making measurements of a three dimensional scene can be achieved by other means than interpreting images. To illustrate why image based measurement is desirable over these methods, a number of alternative measurement methods with overlapping application domains will be reviewed, along with their pros and cons.

1.1.1 Structured Light

One means of structure capture is to project certain easily detectable patterns of light onto an object, and based on the deformation of these patterns infer surface structure. Such systems can produce very high accuracy, even when the objects being measured possess little or no texture. It has for example been used to capture facial expressions, and small objects on turn tables.

The big disadvantage of structured light is a lack of flexibility. As would be expected, it requires tightly controlled lighting conditions and scene positioning. On top of this it is complicated by reflections off more than just one object in the scene, and is thus unable to

handle reflective objects or general environments well.

1.1.2 Ultrasonic

It is also possible to take measurements by bouncing sounds (usually ultrasonic) off an object to be measured and, based on the time the sound takes to return, determine the distance to the object. Initial and well known systems of this form were used to determine the depth of the sea directly under a ship. Modern development has led to very accurate and relatively cheap ultrasonic devices that have been successfully used in applications such as medical imaging and robot navigation.

One advantage and disadvantage of ultrasound is that it can pass through regions that light cannot (e.g. water or skin). However, the speed of sound tends to mean that it can be fairly inaccurate. It is also complicated by unpredictable phenomena such as multiple reflection of sound off different objects in the scene, or reflection and deflection caused by passing through different substances.

1.1.3 Laser Range Finders

Laser range finders work on a very similar basis to ultrasound devices, but use light instead of sound. Distance is then measured by the phase of the light or by the echo time, and tends to produce highly accurate results. Laser range finders have been successfully applied to capture of objects and scenes in tightly controlled conditions as well as to other computer vision applications such as robot navigation.

However, laser range finders suffer from many drawbacks. In particularly the use of phase mean they usually have a limited depth resolution and need to be tuned to a certain depth. They are also easily disrupted by small scene movements such as might occur in a natural scene, they are very expensive, and they can suffer from the same reflection problems as structured light and ultrasonic devices. On the whole they are most useful for controlled conditions where high accuracy is required, e.g. applications such as measurement on production lines.

1.1.4 Passive Devices

All the previous examples have been of active measurement devices. Unlike active devices, passive devices do not send signals into the environment, but instead interpret existing

signals being reflected or emitted from the environment (e.g. light or heat). The advantages are fairly self evident - the device will require only a receiver and not a transmitter thereby making possible greater flexibility and cheaper and simpler hardware.

Cameras are a good example of a passive device. Light reflected off the scene is imaged by a camera, converting the reflections and emissions of light from three dimensional structure into a planar representation. This conversion inevitably results in the loss of some information such as angles or lengths and makes subsequent interpretation of the scene far more difficult. The other major problem is that, because cameras use light, interpretation of images will be subject to suitable lighting being available and will be complicated by many types of lighting effect.

On the other hand, the camera-based approach has many advantages. Firstly, the freedom offered by a camera and the quick image formation time make it very flexible as well as requiring less control of the scene. Cameras are also widely available, with cameras already in extensive use for numerous applications. Cameras are therefore cheap, accurate, easy to use, readily available and widely understood.

Another advantage of camera-based measurement is that objects can be captured regardless of their distance and size. Images are also ideal for taking measurements that will have application to visualisation problems since they record the same information as the eye. This also means that, unlike active devices more information can be recovered - most notable of which is colour. In theory, even lighting and reflective properties of the scene could be determined. Although techniques for lighting capture are not yet well developed, they are fast becoming practical [GHH01].

1.2 3D Reconstruction From Images

This work sets out to further practical reconstruction for more general low cost applications where expensive devices and extremely high accuracy are not appropriate or necessary. As such, it will make use of only a hand-held video sequence to produce three dimensional models using entirely automated means.

This is no small task. As mentioned above the information loss resulting from the imaging process makes reconstruction from just images a difficult task. Subsequently, this section will attempt to present all the different problems that must be solved in order to produce a reconstruction from images, as well as a brief review of the existing solutions and literature. The wide ranging nature of the problem precludes a single literature review, and so more

detailed reviews of existing literature will be given as they become relevant.

1.2.1 Feature Matching

Before any reconstruction can be attempted some method is required for detecting suitably 'interesting' points and tracking them between images. At the heart of this is the most basic element of point tracking: given a 3D feature, where does it appear in two different images? This can be a very difficult problem to solve. If nothing is known about the scene structure and camera positions it is necessary to make certain assumptions about the differences between images that may not always hold true.

Many of the simplified models that are used to match a point in one image to another will hold true only if there is little variation between the images due to camera position, lighting and scene movement. Fortunately, this will usually be the case for a video sequence, and so effective point matching is almost always possible provided the scene itself is largely rigid (i.e. very little movement.). For this reason, this project attempts reconstruction from video sequences rather than arbitrary images.

Two main methods of point tracking for image sequences have been proposed. The first makes use of geometric constraints to help guide a correlation based matching scheme across image pairs or triplets (see [ZDFL95, ZDFL94, BTZ96, FZ98b, Pol99]). The alternative approach (see [TS94, BGK98, TFTR98]) uses a very simple model of image motion and attempts to track features using this alone.

In fact, regardless of the scheme, feature matching can be very closely integrated with the whole reconstruction process. After a reconstruction has been produced, structure can be projected through the hypothetical cameras and used to guide further matching. This matching is significantly simplified, and can occur across larger gaps in the sequence than consecutive image pairs.

1.2.2 Reconstruction of Cameras and Structure

Once some tracked features are available, it becomes possible to attempt to reconstruct the cameras and the three dimensional structure associated with those features. This problem, often referred to as the structure and motion problem, has proved to be quite a formidable task. In essence, the problem involves minimising the error from re-projecting the hypothesised three dimensional structure into images using the hypothesised cameras subject to

some assumed noise model. This function is highly nonlinear and involves projecting unknown structure with unknown cameras. Since such a function cannot be solved directly for a general camera, it can only be used to refine an existing reconstruction (e.g. using the well known Bundle Adjustment [TMHF00, Sla80]).

Initial attempts at solving the structure and motion problem assumed that cameras were calibrated, i.e. that their internal parameters such as focal length and centre were known. The first solutions focused on the two view problem [LH81], but soon progressed to the more difficult problem of longer sequences. Some representative works can be found in [CWC90, SA90, Jac97]. Of note is the work of [TK92] which provides a closed form solution to the reconstruction problem provided the cameras can be approximated with a simpler, less general affine model.

For calibrated systems to work, the calibration of the camera must be very accurate, involving both a cumbersome and difficult calibration stage. Calibration can also be significantly affected by temperature or mechanical shock and so must be repeated for each scene. To avoid these problems, later researchers started to produce reconstructions subject to an unknown projective transformation rather than a Euclidean transformation.

As for calibrated reconstruction, the difficulty of uncalibrated reconstruction meant that early work focused on very small collections of images. For such image collections, similar simple relationships exist for the uncalibrated projective case as were found for the calibrated Euclidean case [Fau92, Har92, Har97]. These so called 'multilinear forms' are much more easily determined from real data and so are very useful for purposes of robust reconstruction (either calibrated or uncalibrated) and for boot strapping larger reconstructions. The development of the theory and practice of multilinear forms is now fairly complete - see [ZDFL95, LF96b, TZ97, Har97, PF98, Har98, TZ00, TZM98] for a sample of some of the most relevant work.

The multilinear forms, whilst effective could only be used to produce cameras and structure for very small collections of images [HS94, BTZ96]. Extra techniques were therefore developed for longer sequences. So-called factorisation techniques [ST96, HBS99], which were originally invented for calibrated reconstruction, take a different approach by attempting to solve for all cameras and structure at the same time, but as a result only produce an approximation. Alternative solutions involve building a reconstruction up bit by bit, starting from multilinear forms and using existing reconstructed data to add new cameras and structure to the reconstructions as well as refining existing ones - see [BZM97, BTZ96, FZ98b] for a sample of such systems.

One final approach of note is that of [AP95, JAP99] which, starting from the assumption that the scene is effectively planar, attempts to produce a full Euclidean reconstruction by a nonlinear minimisation. However, since the initial guess is totally arbitrary it tends to be very unreliable.

1.2.3 Self-Calibration

Although projective structure is easier to determine, and considerably more flexible, it is also harder to work with and less suited to most modelling tasks. Subsequently, much work has also been done in attempting to upgrade reconstructions from the projective framework into a Euclidean framework. This can be achieved very reliably by using known scene details such as vanishing points, angles or shapes (see [FLR+95, HZ00]). Recently, it has also become possible to attempt this using entirely automatic means (see [PKG97]), especially if the reconstruction is good.

Totally automatic calibration has also received much attention over the years. The first general approach was proposed in [MF92] and was based on the so called Kruppa Equations. These equations express a constraint due to the absolute conic - a special conic which remains fixed for the group of Euclidean transformations. Because it is fixed, it can be used to identify Euclidean transformations as a sub-group of projective transformations. Through a large amount of work, this led to fairly stable algorithms for self-calibration (for example [PKG97]), but these require a good reconstruction and are prone to degeneracy [Stu97, Kah99, Stu99].

Other approaches to self-calibration sometimes take advantage of restricted motions such as pure translations [MGDP94] or pure rotations [Har94c]. Similar success has been achieved with stereo rigs [ZBR95] resulting in what appear to be highly effective algorithms. However, unless scene details or restricted motions and cameras are used, self-calibration can still not be regarded as a reliably solvable problem.

1.2.4 Dense Correspondence

The solution to the reconstruction problem and self-calibration only makes use of a sparse set of interesting points. Interesting points are usually selected for the ease and accuracy with which they can be matched between images rather than to allow a model of the structure in the scene to be produced. Once cameras are available though, the knowledge of their positions can be used to greatly simplify the matching problem. It is therefore possible to

return to the images and attempt to match enough features so that a model of the scene can be produced.

Some commercial systems [pbR, pbESI] exist that allow this sort of modelling, provided that cameras and calibration are available. These invariably use some form of user-guided interaction to produce shapes and objects. An alternative approach is to attempt to match very large numbers of points between images (dense correspondence) and then fit surfaces or models to the huge set of points that result.

Performing this dense correspondence is a very difficult problem since many regions of an image exhibit very little texture variation and so are hard to match. To deal with this, many approaches enforce extra constraints to make the resulting points fit neatly to certain natural assumptions, such as surface smoothness and uniqueness of matches.

There are many approaches to dense correspondence. Some are based on feature matching, such as [PMF85, OK85, MP79], and use features to split the images into regions. Others are based on correlating small regions of images, such as [Fal94, Fal97, CHRM96, RC98, Kos93, Sun97, BT98], and then propagate this information so that natural constraints are imposed.

1.2.5 Model Creation

Once a dense model is available, the final stage is to attempt to fit planes and surfaces onto the model. This problem is beyond the scope of this project and has not been extensively approached in this work. However, it does seem fairly clear that existing techniques can probably be adapted to perform model building to a largely automated extent. In [Pol99] thin plate splines were used to determine surfaces and then existing triangulation techniques were used to produce a set of triangles suitable for efficient rendering using existing hardware.

Alternative automatic techniques involve plane fitting [FZ98a], or space carving [KS98], but tend to be suitable only for specific types of scene. However, automation remains a lofty goal, and the most effective solutions are doubtlessly the CAD based ones. These tend to use quite large amounts of user interaction to fit features and objects to the images (for example [Str94]). Knowledge of camera geometry can prove invaluable to this process and can help guide user interaction as well as refining accuracy to greatly improve the quality and speed of reconstruction over a conventional CAD tool. However, complex scenes can still prove difficult to reconstruct.

1.3 Existing Systems

As can be seen from the brief overview in the previous sections automated model production requires solving many different problems and is a very complex task. Whilst total automation is an ideal, for quite a long time systems have been available that allowed partially automated reconstruction using all or some of the techniques from previous sections.

Initial systems such as [TM91, CB88] were focused on the calibrated approach (calibration here refers to known or roughly approximated camera internal parameters and not necessarily relative motion) with sparse collections of images. Calibration is either provided by an accurate calibration stage prior to geometry determination as in [CB88] or is solved for by using rough approximations to the internal parameters with Euclidean motion equations [TM91] (a method which seems to be as effective as more modern work, although possibly less general).

These calibrated approaches that avoid the use of projective reconstruction are now widely commercialised - for example, the well known PhotoModeller [pbESI]. These assume calibrated cameras, and build models using large amounts of user-interaction, in controlling image acquisition, camera calibration, feature matching and modelling. They can produce very effective results, but the large amount of user interaction makes complex models very slow and difficult to produce. In effect, most commercial systems of this type offer CAD tools more than they do automated reconstruction.

More recent systems have greatly improved this approach whilst maintaining the flexibility of using very few photographs. The system of Debevec, Taylor and Malik [DTM96a, DTM96b] is capable of producing reconstructions from a singe calibrated image. It does, however, rely on user interaction to semi-automatically fit a three dimensional model to the image. This was combined to good effect with image based techniques that enabled very realistic texture mapping and display of 3D models.

A similar but multiple view approach to modelling (but not reconstruction) was taken by the DIPAD project [Str94], which uses user interaction to fit a model to calibrated images. Multiple view information was then used to refine the accuracy, and guide the user to further matching in other images. However, camera positioning is determined by user interaction (aligning the existing model with a projected model), and so the whole process, whilst effective, is fairly labour intensive.

To the author's knowledge one of the first systems to offer an uncalibrated approach, where calibration was determined after a projective reconstruction, was the Realise system

[FLR+95]. This system made use of user-specified information, such as vanishing points and parallelism, to calibrate the cameras. Such information is not always readily available, but since this system was designed for reconstruction of urban scenes, such features could easily be obtained.

Such uncalibrated approaches are just beginning to be commercialised. Of particular note is the image processing factory suite of software [pbR], which combines self-calibration techniques with similar user guided model construction. However, to remain practical it was found necessary to allow the user to provide the calibration or extra information to aid in the calibration in order to keep the system general.

Less specifically targeted and non commercial systems have focused more on the automated approach. The first system to make use of video sequences to aid point matching was based on the Tomasi and Kanade factorisation approach [TK92], but was limited by the need for points to be matched across all images and by the use of a less general calibrated affine camera model.

More recent work by Beardsley, Torr and Zisserman [BTZ96] used a full perspective camera model for reconstruction, and added in a greatly improved matching scheme guided by geometry calculation. However, the system still assumed that some means of camera calibration was available if Euclidean structure was required, and did not attempt to build models. More recent work along the same lines [FZ98a] has added actual semi-automatic model building and self-calibration tools to this process.

A similar system [KPG98, Pol99] using the same underlying techniques attempted to use dense correspondence for model building in an attempt to increase automation. It was also the first to introduce totally automatic and fairly general self-calibration algorithms after the projective reconstruction process. Note however, that Euclidean (i.e. calibrated) reconstructions were being more reliably produced using little or effectively no calibration by many earlier systems such as [TM91, CB88].

1.4 Contributions

Before giving an overview of the contents of this thesis, an overview will be given of the main contributions made by this work. This is especially relevant since automated reconstruction from image sequences is far from being a new research field, and so contributions are built on top of numerous existing methods.

- A new projective reconstruction algorithm was developed. This made use of a merging approach to reconstruction whereby larger reconstructions are produced by merging smaller ones. This technique is shown to be much more flexible than existing methods. A number of new techniques are then invented for merging two reconstructions, as well as a number of robust methods. Extensive comparisons with existing projective reconstruction methods show a very significant improvement both in speed and accuracy, especially to robust reconstruction. When accumulated over large sequences, these improvements can be very significant, and allow reconstruction from much less accurate and much longer sequences.

- Feature matching, and subsequently reconstruction, was improved by providing a means of selecting images from a video sequence with which to start the reconstruction. This scheme selects the best pairings so as to maximise the accuracy with which the geometry can be calculated. Without this selection process, reconstruction can be very poor, and also very slow. It also provides a means of avoiding and coping with degeneracy as well as a sensible means of matching with existing structure when using merging reconstruction. Overall, this enables the input sequences to be very large, much larger than could be handled before, as well as greatly increasing the scope of sequences the system could handle.

- A new approach to image rectification was proposed. This makes use of matched points between images to produce a rectification for an image pair which minimises image distortion. The method simplifies previous general methods as well as resolving some minor problems.

- A complete system capable of reconstruction from video sequences was presented that combined all the previously mentioned advancements. This was then used to illustrate the effectiveness of the proposed methods on real data.

1.5 Thesis Overview

Because the system built in this work makes use of many well established theories and techniques, chapters 2 to 6 describe the necessary background. In chapter 2, projective geometry will be briefly described. Projective geometry is important to this work, because it provides a natural mathematical framework with which to describe the image formation

process. This chapter will attempt to communicate the basic structure, transformations and standard notation for this geometry, as well as the close relationship it has with the affine and Euclidean geometries.

Chapter 3 uses the tools of projective geometry to describe the image formation process by a projective camera. This is extended to model the image formation process of more than one view and some relationships arising from multiple views are discussed.

Chapter 4 gives a brief overview of existing techniques for robustly estimating the geometry of multiple views using the multiple view relations discussed in chapter 3. This sets the scene for chapter 5 which then covers reconstruction of cameras and structure either using these relations or using known structure.

With the underlying theory and techniques of multiple view reconstruction laid down by earlier chapters, chapter 6 extensively reviews existing state of the art methods for projective reconstruction of longer sequences. This chapter attempts to describe all the methods as well as show their applicability and limitations.

Chapter 7 then describes the new projective reconstruction algorithm. First, this involves explaining the merging approach to reconstruction. The merging approach is then shown to offer much greater flexibility than existing methods, by designing a number of algorithms for projective reconstruction. Each of these algorithms is designed to take a different form of image data (e.g. image collections or video sequences), but uses the same core techniques. The robust merging algorithm is then described and shown to produce considerably improved results on both synthetic data and real data.

The problem of feature matching both across image pairs and longer sequences is discussed next, in chapter 8. Feature matching is left until this point in the work, because it is so heavily integrated into the reconstruction scheme. The chapter first reviews a method for matching between image pairs, then introduces a method to select suitable image pairs for reconstruction. Finally, a method is given for matching during the merging process used for reconstruction.

With a complete reconstruction system described, chapter 9 describes the rectification process. This process makes matching between images very simple by aligning the cameras so that the two image geometric constraint between images is of a particularly simple form. This simplification is absolutely essential in order to make the dense correspondence algorithms used to produce models tractable.

The penultimate chapter 10 puts all the components described together to create a complete system. It then gives some sample results using real video sequences and discusses

the limitations and possible extensions of the complete system. Finally, chapter 11 briefly presents a general summary and some conclusions from the work.

Chapter 2

Basic Projective Geometry

2.1 Introduction

Geometry will prove an invaluable tool in addressing the reconstruction problem. It provides a well established and well tested means of describing a scene, its projection and the process by which it is projected. Whilst the familiar Euclidean geometry will prove highly useful to this end, there also exists another more relevant geometry - namely projective geometry. Projective geometry was developed in the 17th century by Desargues, specifically to model what is left after the projection of a Euclidean space and so is already tailored to reasoning about images.

Knowledge of projective geometry is not absolutely essential to make use of structure and motion algorithms, but it is not easily possible to understand the theory underlying the algorithms unless at least some projective geometry is known. Indeed, the close relationship between Euclidean and projective geometries means that projective geometry is already in widespread use in computer graphics where the conventional notation is used to express projective concepts (e.g. projection) in a linear manner. Readers who are further interested in the subject of projective geometry may like to consult projective geometry texts, such as the classic [SK51] or the simpler [Sam88], and the directly relevant [MT96].

This chapter will start by attempting to provide a conceptual understanding of projective geometry, and its relation to the other natural geometries (Euclidean and affine geometries). A concise definition of projective geometry will then be given with the introduction of some general rules and notation for n dimensional projective spaces. These are then used to examine in detail the two and three dimensional projective spaces that are most commonly used in computer vision, in this case to represent an image and a world space respectively.

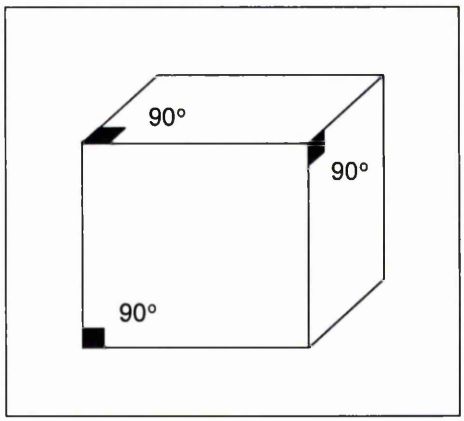

Figure 2.1: Example projection of a cube to illustrate perspective effects. Note how the right angles no longer appear as right angles in the projection

Finally, the notion of a hierarchy of geometries consisting of Euclidean, affine and projective geometries is refined and the relationship between them discussed in detail.

2.2 The Hierarchy of Geometries

Usually, we think in terms of the world around us as being Euclidean in nature, and as such model it using Euclidean geometry. However, when we actually observe the world using our eyes (effectively by a process of projection) we do not see the world in a Euclidean form. For example, consider figure 2.1 illustrating a projected cube. In this case, because it is a cube, we know all the corner angles are 90 degrees, but in the projection they do not all appear to be 90 degrees. This is an example of a perspective effect, and is caused by the loss of some information (in this case angles) when a Euclidean space is projected.

This loss of Euclidean concepts through projection also includes notions such as distance, continuity and betweenness. It is from this realisation that projective geometry arises; it is a geometry based purely on those properties of Euclidean geometry which remain invariant to projection. As would be expected, since projective geometry is based on a looser interpretation of the same underlying world, there is a close relationship between Euclidean and projective geometries. Indeed, this is the case. However, before continuing to a more concise description of this relationship, it will first be necessary to consider in more detail how it is possible to actually define a geometry.

The famous Erlangen program of Klein (Klein's inaugural address to the University of Erlangen in 1872) takes the approach of studying all geometries as a space of points and the

group of transformations that leave the structure of that space unchanged. Theorems are then just invariant properties of the group of transformations. It follows that a geometry can be described completely either by the set of axioms (fundamental and absolute principles, e.g. definition of points, lines, parallelism, etc.) giving the structure of the space, or alternatively by the set of transformations that leave this space unchanged. Either the axioms or the transformations can be implicitly derived from each other.

Now that a method of understanding geometries has been given, it is possible to address the standard hierarchy of 'natural' geometries in detail. These geometries are termed natural here, because they have been created to model the natural world:

$$\text{Projective} \subset \text{Oriented Projective} \subset \text{Affine} \subset \text{Metric} \subset \text{Euclidean}$$

Starting from Euclidean geometry and moving up the hierarchy, each geometry is based on the previous geometry, but with fewer axioms (derived from the same set). This means each successive geometry has a less rigid space and hence also more and more transformations that leave that space invariant. Section 2.3 below will aim to show how the most general of these geometries - projective geometry - can be defined and expressed. The concepts introduced will then be used to describe in more detail the hierarchy of geometries mentioned here.

2.3 Projective Geometry

Now that an intuitive interpretation of projective geometry has been given, it is appropriate to provide a much more rigorous definition including the conventional notation normally used to represent projective spaces. Note that from this point on, no distinction will be made between the algebraic treatment of projective geometry expressed using the given notation, and the 'coordinate free' geometric viewpoint.

For an n dimensional Euclidean space R^n, it is normal to express points using n numbers, each giving a coordinate position in each dimension. However, in an n dimensional real projective space $\mathcal{P}^n$, a point is described by an $n+1$ vector of coordinates $\mathbf{x} = [x_1, \ldots, x_{n+1}]^T$ where at least one of the x_i is non-zero. The vector representation for a point in $\mathcal{P}^n$ is often referred to as the homogeneous or projective coordinate representation of the point. As stated at the beginning of this section, in this text both a point and the coordinate vector representation will be indicated using the same symbol.

There is a further constraint on this coordinate vector representation that needs to be defined before it will exclusively represent the space of all projective points. Two coordinate

vectors $\mathbf{x} = (x_1, \ldots, x_{n+1})^T$ and $\mathbf{y} = (y_1, \ldots, y_{n+1})^T$ are defined as being equivalent if there is some non zero scalar λ such that $\lambda x_i = y_i \ \ \forall i \in \{1, \ldots, n+1\}$ resulting in the following important condition:

$$\forall \mathbf{x} \in \mathcal{P}^n, \lambda \in R, \lambda \neq 0 \Rightarrow \lambda \mathbf{x} = \mathbf{x} \tag{2.1}$$

This so called scale factor constraint is of great significance, since, as will be seen throughout this text, the lack of a one to one relationship between points and coordinate vectors makes the application of linear algebra to projective geometry slightly more difficult. To aid clarity, throughout this text, the scale factor constraint will be indicated by using the symbol $\simeq$ and so equation 2.1 can be rewritten as $\mathbf{x} \simeq \mathbf{x}$.

2.3.1 Projective Transformations

A collineation is just a linear transformation of a projective space which preserves collinearity (i.e. collinear points are mapped to collinear points). For an n dimensional projective space a collineation from $\mathcal{P}^n$ into itself is any $(n+1) \times (n+1)$ matrix $\mathbf{A}$ where $det(\mathbf{A}) \neq 0$ (i.e. it is invertible). An important observation due to equation 2.1 above is that any matrix A associated with a collineation is defined subject to a nonzero scale factor λ and so $A = \lambda A$.

In the more general case, an arbitrary projective transformation can define a mapping from $\mathcal{P}^m$ into $\mathcal{P}^n$ and can be represented by any $(m+1) \times (n+1)$ matrix which need not be invertible. Sometimes, and if they are invertible, projective transformations are referred to as projectivities, collineations or homographies. Any projective transformation A can be applied to a point $\mathbf{x}$ linearly as $\mathbf{x}' \simeq A\mathbf{x}$.

2.3.2 The Projective Basis

Just as in Euclidean geometry, it is possible to consider projective spaces in terms of an arbitrary basis. A projective basis in $\mathcal{P}^n$ is described by a set of $n+2$ points of $\mathcal{P}^n$ such that no $n+1$ of these points are linearly dependent (not on the same line). Any point in $\mathcal{P}^n$ can then be described in terms of any $n+1$ of these basis points. This description of a point omits one of the basis points, but this surplus point is still needed to constrain the arbitrary scale factor for all the other points in the basis.

The standard projective basis is given by $\mathbf{e}_i = (0, \ldots, 1, \ldots, 0), \ \forall i \in \{1, \ldots, n+1\}$ where the 1 is at the ith position in the vector and $\mathbf{e}_{n+2} = (1, \ldots, 1)$. A projective point

$\mathbf{x} = (x_1, \ldots, x_n)$ of $\mathcal{P}^n$ can be described using any $n+1$ points of the standard basis e.g.:

$$\mathbf{x} = \sum_{i=1}^{n+1} x_i e_i$$

This form of projective basis is often known as the canonical basis and will be important in simplifying representations in later chapters (see section 3.2.8, page 52 on canonical form for camera matrices). An important result associated with bases given in [Fau93] states that any projective basis of $\mathcal{P}^n$,$\mathbf{x}_1, \ldots, \mathbf{x}_{n+2}$ can be transformed via a uniquely determined projectivity T to the canonical basis as $\mathbf{e}_i \simeq T\mathbf{x}_i$. Similarly, and in general, any projective basis can be transformed into any other projective basis by a unique projectivity in the same manner.

2.3.3 The Projective Plane

The projective plane is simply the projective space $\mathcal{P}^2$. This is the simplest projective space that will be of direct practical interest, because it is ideal for modelling the image plane of a camera (a plane in $\mathcal{R}^2$). This is possible because $\mathcal{R}^n$ is embedded within $\mathcal{P}^n$. More details of this embedding will be given later.

As described in section 2.3 we can represent a point in $\mathcal{P}^2$ as a coordinate vector $\mathbf{x} \simeq (x_1, x_2, x_3)^T$. Similarly, a line is represented by a 3-vector $\mathbf{l} \simeq (l_1, l_2, l_3)^T$, and consists of all points satisfying the equation $\mathbf{l}^T\mathbf{x} = 0$. Both lines and points can be swapped in this equation without altering it. This results in the aptly named principle of duality - that to any theorem of 2-dimensional projective geometry there is a dual theorem obtained by reversing the roles of lines and points.

As a consequence of this duality it is possible to think of any coordinate vector $\mathbf{x}$ in the projective plane as representing either:

- The set of lines which pass through the point that the coordinate vector $\mathbf{x}$ defines; that is all lines described by the coordinate vector $\mathbf{l}$ for which $\mathbf{l}^T\mathbf{x} = 0$
- Or, alternatively, the coordinate vector $\mathbf{l}$ can be thought of as the set of points represented by the line equation it defines. This means all points $\mathbf{x}$, for which $\mathbf{l}^T\mathbf{x} = 0$.

It is important to note that a line passing through two points can be found as the cross product of the two points. Also, if a matrix A represents a mapping of homogeneous coordinates then A^* (the matrix of cofactors) is the corresponding mapping applicable to

lines. In future discussions, it is worth noting that for invertible matrices the line map can easily be obtained as $A^* \approx (A^T)^{-1}$.

2.3.4 Projective 3-space

The next important projective space to be looked at is $\mathcal{P}^3$. Just as in Euclidean space $\mathcal{R}^3$, in $\mathcal{P}^3$ there are points, lines and planes. From the definition of projective spaces, points in $\mathcal{P}^3$ can be modelled as a tuple of 4 numbers - for example the point $\mathbf{x}$ can be represented by the coordinate vector $(x_1, x_2, x_3, x_4)^T$. Similarly, planes are also represented by a tuple of four numbers - for example the equation of the plane (u_1, u_2, u_3, u_4) is:

$$u_1x_1 + u_2x_2 + u_3x_3 + u_4x_4 = 0$$

The fact that points and planes have the same representation leads us back to the principle of duality. In $\mathcal{P}^2$ there were identical representations for points and lines, and similarly in $\mathcal{P}^3$ there are identical representations for points and planes. As before, this means there are two ways to think about a coordinate vector $\mathbf{y}$ in the projective space:

- The set of all planes which pass through the point that the coordinate vector $\mathbf{y}$ defines; that is to say all planes described by the coordinate vector $\mathbf{x}$ for which $\mathbf{x}^T\mathbf{y} = 0$
- Or, alternatively the coordinate vector $\mathbf{x}$ can be thought of as the set of points represented by the plane $\mathbf{x}$ defines. This means all points $\mathbf{y}$ for which $\mathbf{y}^T\mathbf{x} = 0$

A line in $\mathcal{P}^3$ now has to be represented either as the linear combination of two points, i.e. $\lambda_1\mathbf{x}_1 + \lambda_2\mathbf{x}_2$ or as the intersection of two planes.

2.4 The Hierarchy of Geometries Revisited

Now that projective geometry has been described, it will be possible to use the notation and concepts from projective geometry, to show how it sits in the hierarchy of 'natural' geometries first mentioned in section 2.2:

$$\text{Projective} \subset \text{Oriented Projective} \subset \text{Affine} \subset \text{metric} \subset \text{Euclidean}$$

To reiterate, starting from Euclidean geometry and moving up the hierarchy, each geometry is based on the previous geometry, but with less rigid structure and more transformations.

In fact, the space of all points in each geometry is a subset of the space of all points in the geometry above it in the hierarchy. Similarly, all transformations in a lower geometry form a subset of all the transformations in a higher geometry.

The rest of this section will aim to show in more detail how these different geometries fit together and how they restrict transformations and structure for the particular case of 3 dimensional space. It will also show how homogeneous (projective) notation, being the most general notation, can easily be used to express the transformations and structure of all these geometries.

2.4.1 Projective Geometry

The weakest geometry is projective geometry. It preserves only those properties of Euclidean structure that remain invariant to projection, and hence allows the largest group of transformations. As seen earlier in this chapter, a general transformation of $\mathcal{P}^3$ can be represented using homogeneous notation by any 4×4 matrix:

$$T_P \simeq \begin{bmatrix} p11 & p12 & p13 & p14 \\ p21 & p22 & p23 & p24 \\ p31 & p32 & p33 & p34 \\ p41 & p42 & p43 & p44 \end{bmatrix} \tag{2.2}$$

defined subject to a nonzero scale factor (it need not be invertible). Consequently, there are 15 degrees of freedom in a projective transformation. It is important to note that if homogeneous notation can express a transformation in a linear manner it must be a projective transformation.

Since the corresponding structure is restricted by so few invariants, there are not many relations that remain protectively invariant. Primarily, notions of incidence, collinearity and tangency are projective invariant as well as the cross ratio and cross ratio derivatives. However, compared to the richness of Euclidean geometry, there is relatively little structure. Note that for the practical problems projective geometry is to be applied to in this work, these relations will never be exact and will always be subject to the noise present in the images (e.g. the intersection of two lines will never be exact).

2.4.2 Oriented Projective Geometry

Oriented projective geometry is a relatively new concept, only recently being proposed by Stolfi [Sto91]. The direct usefulness of the concepts underlying oriented projective geometry to the structure and motion problem was first pointed out in [Har93], and refined in [Lav96, LF96a] (where the link to oriented projective geometry was made).

The new geometry is obtained by only a minor modification to normal projective geometry. Returning to equation 2.1, it has been seen that all projective points can be expressed as an equivalence class of vectors $\mathbf{x}$:

$$\forall \mathbf{x} \in \mathcal{P}^n, \lambda \in R, \lambda \neq 0 \Rightarrow \lambda \mathbf{x} = \mathbf{x}$$

In other words, if $\mathbf{x}$ is a vector representing a point in $\mathcal{P}^n$ then so is $\lambda\mathbf{x}$. It may be useful to consider each non zero vector as defining a line through the origin. Under this consideration, two vectors can be considered equivalent if they define the same line.

In oriented projective geometry, the equivalence relation is changed slightly so that if $\mathbf{x}$ is a vector representing a point in oriented n space $\mathcal{O}^n$, then so is $\lambda\mathbf{x}$ provided that $\lambda > 0$, i.e.

$$\forall \mathbf{x} \in \mathcal{O}^n, \lambda \in R, \lambda > 0 \Rightarrow \lambda \mathbf{x} = \mathbf{x}$$

Considering each non zero vector as lines again, equivalence now becomes between half-lines defined by vectors. This allows the definition of a coherent orientation over the whole of O^n so that it becomes possible to introduce the notion of betweenness, i.e. that one point lies in front of or behind a plane. Transformations of oriented projective geometry have the same notation as for projective geometry (see equation 2.2) but are now also defined subject to a non zero positive scale factor. Note that the extra constraint on the scale factors means that for every projective transformation T_p there exist two oriented projective transformations T_p and $-T_p$.

In the context of the structure and motion problem this notion of orientation will prove highly useful to distinguish between points in front of and behind a camera (not possible with normal projective geometry). This can be achieved using the constraint that all points known to be visible in an image must be in front of the camera. It is expressed in the reconstruction of the 3D points by enforcing the convention that points in front of the camera have a positive transformation T_p and those behind have a negative transformation $-T_p$.

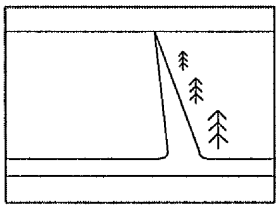

Figure 2.2: Illustration of infinite points: Parallel lines such as the road in this figure converge in an image at a point infinitely far away. Similarly, the plane on which the road lies intersects the set of infinite points at a line - the horizon line.

2.4.3 Affine Geometry

Affine geometry can be seen to fit between projective and Euclidean geometries. It allows for the inclusion of important missing concepts such as parallelism and the notion of betweenness i.e. that one point can be between two other points. These properties are added to projective geometry to create affine geometry, by the identification of so called infinite points. For example, imagine the image of a pair of parallel lines; the two lines converge to a point infinitely far away on the horizon line as shown in figure 2.2. This convergence at infinite distance can serve as a definition of parallelism. So, by the identification of the infinite points forming a line in an image (or a plane at infinity for 3D space), the geometry may be considered affine.

This definition allows us to consider affine space as a projective space, but with the points at infinity identified. Consequently, an affine transformation must be a projective transformation that maps infinite points to infinite points. Given that infinite points must remain within the plane at infinity, there is no need for the capacity to express them explicitly when working with affine points. Consequently, points in $\mathcal{A}^3$ can be defined using only 3 coordinates instead of the 4 used for homogeneous notation. Transformations of affine space then take the following form:

$$\begin{bmatrix} X' \\ Y' \\ Z' \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} + \begin{bmatrix} a_{14} \\ a_{24} \\ a_{34} \end{bmatrix}$$

It has already been established that the group of affine transformations will be a subgroup of all projective transformations, and so, subsequently, it makes sense that affine structure and transformations can be represented using homogeneous notation. In order to do this, a special plane must be fixed in P^3 to act as the infinite plane. For simplicity in notation mappings, by convention infinite points are usually fixed as all points in $\mathcal{P}^3$ with a final

coordinate of 0. i.e. for a point in $\mathcal{P}^3$, (X, Y, Z, W) the infinite plane is all points of the world with the form $(X, Y, Z, 0)$. Consequently, the plane at infinity can be defined as $\Pi_\infty = (0, 0, 0, 1)$ in $\mathcal{P}^3$.

Given this convention for the location of the plane at infinity, a one to one mapping can be obtained from affine coordinates to projective coordinates as $\mathcal{A}^3 \rightarrow \mathcal{P}^3 : (X, Y, Z) \rightarrow (X, Y, Z, 1)$, i.e. take the affine coordinate vector and add a 1. Similarly, if the infinite points are known and fixed as just defined, a mapping from projective space to affine space can be determined as:

$$\mathcal{P}^3 \rightarrow \mathcal{A}^3 : (X, Y, Z, W) \rightarrow \left(\frac{X}{W}, \frac{Y}{W}, \frac{Z}{W}\right)$$

This mapping is naturally only defined if $W \neq 0$. This is convenient since such points are by our convention the infinite points and do not need a representation in affine notation anyway.

Since this relation allows affine structure to be expressed with homogeneous notation, it makes sense to provide a representation for affine transformations in homogeneous notation. These take the form:

$$T_A = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ 0 & 0 & 0 & \lambda \end{bmatrix}$$

Here, λ is introduced to account for the arbitrary scaling in homogeneous notation. As can be seen from this, an affine transformation has 12 degrees of freedom and it can easily be verified that the homogeneous form leaves the plane at infinity $\Pi_\infty = (0, 0, 0, 1)$ invariant i.e. $T_a \Pi_\infty = \Pi_\infty$. Note that this does not mean that the position of points in the plane at infinity remain unchanged - they only remain within Π_∞.

2.4.4 Euclidean Geometry

Euclidean space, in this case, means Euclidean space under the group of similarity transforms, i.e. we also allow uniform changes of scale as well as rigid displacements. In some treatments of Euclidean geometry, this is considered as extended Euclidean, similarity or metric geometry. For the case in hand of reconstructing 3D information from photographs, absolute length cannot be determined anyway without taking some actual scene measurements, so any measurements taken from photographs are usually subject to an arbitrary scale.

Overall Euclidean geometry represents a very significant strengthening of structure and a subsequent decrease in the number of invariant transformations. The only remaining transformations are similarity transforms, that is transformations which preserve angles, but not necessarily length. With this restriction comes two new important invariant properties, angles and relative lengths.

In order for transformations to leave this rigid structure invariant, it is necessary to further specialise the group of affine transformations by requiring they leave a particular conic invariant as well as the infinite points. In $\mathcal{R}^3$ this conic is known as the absolute conic Ω and is located on the plane at infinity. The notion of the absolute conic is a little more abstract than the notion of the plane at infinity, but it can be understood by reconsidering the horizon line discussed previously in the context of affine geometry. When observing the world it can be seen that parallel lines can converge to any point on a sphere at infinite distance, depending on the orientation of the lines in 3-space (defined by their angle). It follows that the angle two lines make at their intersection can be determined by where they intersect with the infinite sphere.

With the geometry defined, the notation can now be defined. As with affine geometry, points in R^3 can be represented by using 3 coordinates. This means there is a one to one bidirectional mapping between points in R^n and points in A^n and so the same notation is valid for the structure of both affine and Euclidean geometries. The big difference is that, in Euclidean geometry, transformations must take the further restricted form:

$$\begin{bmatrix} X' \\ Y' \\ Z' \end{bmatrix} = \lambda \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix}$$

where r_{ij} are coefficients of an orthonormal matrix R, t_i is a translation and λ represents an arbitrary scaling. λ should be removed when dealing with a truly Euclidean transformation instead of a metric one. The above transformation can be seen to have 7 independent degrees of freedom, 3 for the rotation, 3 for the translation and an arbitrary scaling.

Note that if an $n \times n$ matrix is orthonormal then it must be subject to $\sum_{i=1}^{n} i$ constraints as $\sum_{k=1}^{n} r_{ik} r_{jk} = \delta_{ij}, (1 \leq i \leq j; 1 \leq j \leq 3)$ with δ_{ij} as the Kronecker delta:

$$\delta_{ij} = 1 \text{ for } i = j$$

$$\delta_{ij} = 0 \text{ for } i \neq j$$

This corresponds to the matrix relation that $R^T R = R R^T = I$,hence $R^{-1} = R^T$, and $\det R = 1$.

In the same way that affine transformations can be expressed in homogeneous notation, Euclidean transformations can too. In this case, a general transformation takes the form

$$T_M \simeq \begin{bmatrix} \lambda r_{11} & \lambda r_{12} & \lambda r_{13} & t_x \\ \lambda r_{21} & \lambda r_{22} & \lambda r_{23} & t_y \\ \lambda r_{31} & \lambda r_{32} & \lambda r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

where again the r_{ij} form an orthogonal matrix. Since the absolute conic is invariant to Euclidean geometry, and hence not a Euclidean concept, Euclidean notation does not need the capacity to describe it. However, since a one to one mapping between Euclidean points to projective points has been established (via affine geometry), it follows that we can use projective geometry to describe the absolute conic. The conic is given by the intersection of the quadric of equation $X^2+Y^2+Z^2+W^2=0$ with Π_∞. This places the restriction $W=0$, so Ω is given by:

$$X^2 + Y^2 + Z^2 = 0 \text{ with } W = 0$$

Note that this entity requires two equations to describe it and so is somewhat cumbersome. It is however possible to use the dual of the absolute conic, the absolute dual quadric Ω^*. This is referring back to the principal of duality, and so whilst the absolute conic is defined as an equation on points, the absolute dual quadric is defined as an equation operating on lines (in $\mathcal{P}^2$) or planes (in $\mathcal{P}^3$). This allows the absolute dual quadric in $\mathcal{P}^3$ to be expressed in homogeneous notation much more simply as (in canonical form):

$$\Omega^* \simeq \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Note that the plane at infinity $\Pi_\infty = (0,0,0,1)$ is both the left and right null space of Ω^*. Being a dual quadric, a transformation T is applied to it as $T\Omega T^T$. Using this definition, it is easy to verify that Euclidean transformations leave this conic invariant.

So far, a mapping from Euclidean to affine and hence projective notation has been given. The equivalent mapping of projective to Euclidean structure is however more complex, since it requires that the infinite plane and absolute conic be identified. Given knowledge of these two entities they can both be transformed to their canonical forms and, subsequently, structure may be thought of as being Euclidean, but represented in projective space.

geometry	transformations	DOF	invariants
projective	$T_P \simeq \begin{bmatrix} p11 & p12 & p13 & p14 \\ p21 & p22 & p23 & p24 \\ p31 & p32 & p33 & p34 \\ p41 & p42 & p43 & p44 \end{bmatrix}$	15	cross-ratio, collinearity, incidence, tangency
affine	$T_A \simeq \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix}$	12	relative distances along direction, parallelism
metric	$T_M \simeq \begin{bmatrix} \lambda r_{11} & \lambda r_{12} & \lambda r_{13} & t_x \\ \lambda r_{21} & \lambda r_{22} & \lambda r_{23} & t_y \\ \lambda r_{31} & \lambda r_{32} & \lambda r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$	7	relative distances, angles
Euclidean	$T_E \simeq \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$	6	absolute distances

Table 2.1: Properties of the standard hierarchy of geometries. The coefficients r_{ij} form orthonormal matrices.

2.4.5 Summary of the Hierarchy

Table 2.1 gives a brief overview of all the geometries in the hierarchy, the number of degrees of freedom in a transformation of that geometry, and some of the major invariants of the geometry.

2.5 Summary

This chapter has provided an introduction to the nature of, and notation used for, the hierarchy of natural geometries, consisting of Euclidean, metric, affine and projective geometries. These geometries and their notation can now be used to provide a means of describing the camera and scene and so will prove an invaluable tool for the process of reconstruction.

In particular, projective geometry provides a very convenient means of expressing the contents of an image in terms of the information it contains about three dimensional space that was not lost during projection. Furthermore, the notation of projective geometry allows this to be expressed in a linear manner.

Chapter 3

Camera Model and Multiple View Geometry

3.1 Introduction

Since the ultimate goal of this work is to produce a reconstruction from a set of images, an accurate description of the process of image formation will be necessary. To this end, this chapter first introduces an appropriate geometric model of a camera, provides an analytical representation and then generalises this to describe worlds defined using affine or projective geometry. Finally, the second part of the chapter discusses important relationships arising from this model, when multiple views of the same scene are available.

3.1.1 Preliminaries

In later discussion, it will be useful, when given a column vector, $\mathbf{t} = (t_x, t_y, t_z)^T$, to introduce the skew-symmetric matrix $[\mathbf{t}]_\times$ defined as:

$$[\mathbf{t}]_\times = \begin{bmatrix} 0 & -t_z & t_y \\ t_z & 0 & -t_x \\ -t_y & t_x & 0 \end{bmatrix}$$

For any non-zero vector $\mathbf{t}$, the above matrix has rank 2 and is closely related to the cross-product of vectors. Given a vector $\mathbf{s}$ then $\mathbf{s}^T [\mathbf{t}]_\times = \mathbf{s} \times \mathbf{t}$ and $[\mathbf{t}]_\times \mathbf{s} = \mathbf{t} \times \mathbf{s}$. Although this convention and matrix is used throughout this text, it has been restated here because knowledge of the properties held by this skew symmetric matrix is key to some of the

derivations in this chapter.

It is also worthwhile to bear in mind that, throughout this chapter there is only a slight distinction between a metric and a Euclidean world. A metric world is subject to an unknown and arbitrary scale factor. This is different to a truly Euclidean world where the scale factor is fixed (see section 2.4.4 on page 39 for more details). Note that most theoretical discussion is relevant to either form and in other texts the distinction is sometimes not even made, with metric geometry being referred to as Euclidean.

Pseudo Inverse

The pseudo inverse of non square matrices will receive a fair amount of use throughout this work, and so will be described in detail here. The pseudo inverse of a square diagonal matrix D is the diagonal matrix D^+, such that:

$$D^+_{ii} = \begin{cases} 0 & \text{if } D_{ii} = 0 \\ D^{-1}_{ii} & \text{if } D_{ii} \neq 0 \end{cases}$$

This can be extended to an $m \times n$ matrix A where $m \geq n$ by considering the SVD of A as $A = UDV^T$. The pseudo inverse of A is then given by:

$$A^+ = VD^+U^T$$

As is generally the case when performing an SVD if $n < m$ then A can be extended to a square matrix by adding rows of zeros. The pseudo inverse A^+ of an $m \times n$ matrix A has the property that $AA^+ = I_{m \times m}$.

3.2 Camera Model

There are numerous ways to approach the modelling of a camera, depending on the type of camera being used, the accuracy required and the type of information to be considered. In this work, a lens-based camera will be considered leading to a model based on central projection, sometimes referred to as the pinhole or perspective camera model. A geometric illustration of this model is given in figure 3.1. Despite its simplicity, this model is sufficient for representing most commonly used lens based cameras.

In figure 3.1, the two screens R and F are two planes, with the plane labelled R being called the *retinal plane* or *image plane* and the second plane F being called the *focal plane.*

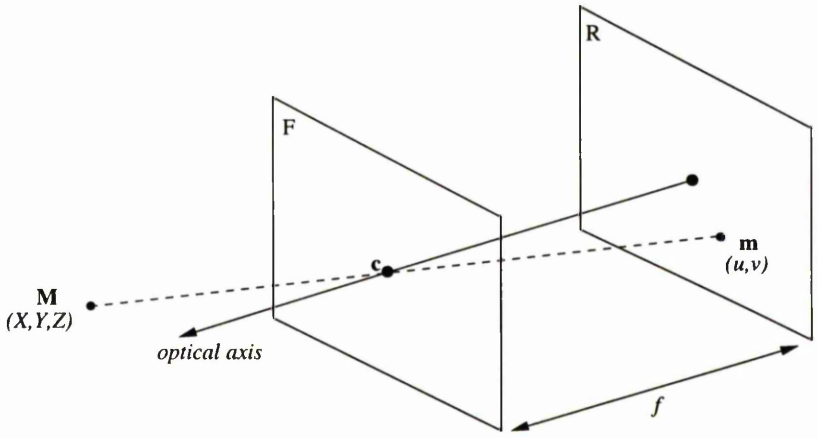

Figure 3.1: The pinhole camera model

The image of any point $\mathbf{X}$ on the retinal plane R is formed by an operation called a *perspective projection* which uses the point $\mathbf{c}$ called the *optical centre* at a distance f (known as the *focal length*) from R, to form an image on R of world structure $\mathbf{X}$ in the scene as the intersection of the line $< \mathbf{c}, \mathbf{X} >$ with the retinal plane R. Note that this projection process is undefined if the point $\mathbf{X}$ is at $\mathbf{c}$. Also important to future discussion is the line passing through the optical centre $\mathbf{c}$ and orthogonal to the retinal plane R, which is known as the *optical axis* or *principal ray.*

In order to make use of this simple geometric model, it is necessary to derive a quantitative interpretation of it. There are two main ways to go about this - either model the physical effects of the camera and create a model based on reflection and emission of light energy, or alternatively make a geometric model which considers reflection and emission of rays of light. Since the goal of this work is to reconstruct geometric and not photometric information, the latter approach will be taken here. Equivalent photometric models can be found in many books on computer graphics or computer vision and will not be discussed further here.

3.2.1 A Simple Model

For the simplest model, the optical centre $\mathbf{c}$ is placed at the origin of the world coordinate system and the retinal plane is taken to be the plane $Z = 1$ (i.e. the optical axis is aligned with the z axis). In this case, the projection of 3D structure (X, Y, Z) in the world coordinate system (object space) into the image coordinate system (u, v) (image space) can be

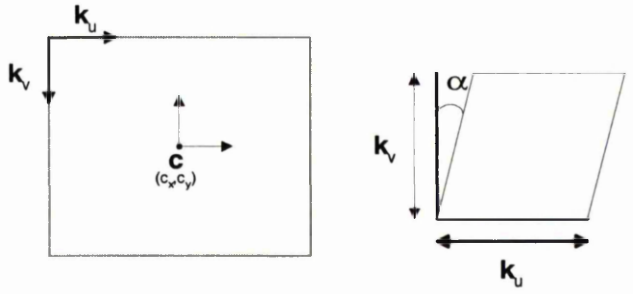

Figure 3.2: Effect of the intrinsic parameters on image formation.

represented as (this can be derived using similar triangles):

$$u = \frac{X}{Z} \quad v = \frac{Y}{Z}$$

However, despite the simplicity of the model this representation is still nonlinear. Since it would be convenient to be able to exploit the power of linear algebra, it is best to express this relationship in a linear manner by using the tools of projective geometry. The projection equation just presented transforms points from a metric object space $\mathcal{R}^3$ to a metric image space $\mathcal{R}^2$. Since metric space $\mathcal{R}^n$ can be embedded within projective space $\mathcal{P}^n$, it is possible to consider projection as a projective transformation and express it linearly using homogeneous notation:

$$\begin{bmatrix} u*s \\ v*s \\ s \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X*T \\ Y*T \\ Z*T \\ T \end{bmatrix} \tag{3.1}$$

where the arbitrary non zero scale factors s and T have been introduced to convert the metric points into homogeneous notation. Also note that if the world point (X, Y, Z) is on F then $Z = 0$ and the mapping just presented is undefined.

3.2.2 Intrinsic Parameters: Camera Calibration

In practice, an actual camera will not have the ideal arrangement assumed in the simple model of the last section, but will distort the image in a number of ways. For a start, the focal length of the camera will not necessarily be 1 and so the image points will be scaled by a factor f. A further scaling occurs independently on each axis k_u, k_v to account for the

difference between the size of the sensors used to sample the image and some arbitrary but fixed unit of length.

Further distortion is also caused when the intersection of the optical axis with the retinal plane is not at the centre of the image. To remove this, a transformation (c_x, c_y) can be applied to take the centre to the origin. Finally, to account for non-orthogonality between the image axis a skew factor also needs to be introduced. All these effects are illustrated in figure 3.2 above, and can be modelled by a transformation applied to the ideal image coordinates $(\hat{u}, \hat{v})$ to get the actually observed image coordinates (u, v):

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} fk_u & (tan(\alpha))\, fk_v & c_x \\ 0 & fk_v & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \hat{u} \\ \hat{v} \\ 1 \end{bmatrix}$$

In practice, k_u and k_v need not be considered individually and are often combined with the focal length to create f_u and f_v. The ratio between these two values, i.e. $\frac{f_v}{f_u}$ is often referred to as the aspect ratio. Similarly, the term representing the skew factor is simply considered as a single independent unknown resulting in the simpler form that will be used throughout this text:

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_u & s & c_x \\ 0 & f_v & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \hat{u} \\ \hat{v} \\ 1 \end{bmatrix} \tag{3.2}$$

The upper triangular matrix in the previous equation is often referred to as the camera calibration matrix and will be represented throughout the rest of this text using the notation K. Often, for convenience, if this transformation is known, it is applied to all image coordinates $\mathbf{x}_i$ to get the normalised image coordinates $\hat{\mathbf{x}}_i$ as:

$$\hat{\mathbf{x}}_i \simeq K^{-1}\mathbf{x}_i$$

3.2.3 Extrinsic Parameters: Camera Position

Up to now it has been assumed the centre of the camera is located at the centre of the world coordinate system and oriented so there is no rotation away from the axis. However, when considering more than one camera or a specific world coordinate system it is necessary to account for the different positions and orientations of the cameras. Since a camera can be placed at the centre of the world by a rigid transformation, it is possible to add a

transformation to the camera description so that it will be transformed to the desired position in space:

$$P = P_{CENTRE} \begin{bmatrix} R & \mathbf{t} \\ \mathbf{0}_3^T & 1 \end{bmatrix}$$

for some rotation R and translation $\mathbf{t}$. Note that because of the gauge freedom, by simply changing the convention it is possible to express the same rotation and translation as an inverted rotation and translation applied to the cameras. Applying the inverse form gives rise to a different expression for the camera:

$$P = P_{CENTRE} \begin{bmatrix} R^T & -R^t\mathbf{t} \\ \mathbf{0}_3^T & 1 \end{bmatrix} \tag{3.3}$$

In general, whichever form provides the most mathematical convenience will be used. Usually, this means the inverse based form is used because it makes the centre of projection $\mathbf{t}$ (centre of projection is given by the null-space of P). It is both intuitively correct and mathematically convenient to have the centre of the camera at the camera's position in space.

3.2.4 The Complete Model and the Projection Matrix

Combining all the equations 3.1, 3.2 and 3.3 gives an expression for projection using a pinhole camera with a given internal calibration as well as position and orientation.:

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_u & s & c_x \\ 0 & f_v & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} R^T & -R^t\mathbf{t} \\ \mathbf{0}_3^T & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{3.4}$$

Rather than expanding out the matrices, this will usually be written compactly in block form as

$$\mathbf{x} \simeq K \left[R^T \middle| -R^T\mathbf{t} \right] \mathbf{X}$$

or even

$$\mathbf{x} \simeq P\mathbf{X}$$

where the 3×4 matrix P is known as the camera projection matrix.

3.2.5 Other Distortions

The model described so far, although workable, is idealised and is often further refined by taking into account the optical distortions of the camera lens. There are both geometrical and physical distortions, but since we are dealing exclusively with the geometric case, physical distortions will be ignored. Distortions are modelled by introducing a new non-linear stage after the projection - for example, to model the radial distortion (generally the most prominent effect):

$$x_c = x + \alpha x(x^2 + y^2), y_c = y + \alpha y(x^2 + y^2)$$

for corrected image coordinates x_c, y_c and sampled coordinates x, y. There are many other forms of correction and the topic has been extensively studied - see for example [TV96, Sla80] - but this work will generally ignore these extra distortions. This is a practical assumption, often applicable to high quality camera lenses. However, it should be noted that even with high quality lenses it can be helpful to account for distortions, particularly radial distortion (see [HZ00] for a good overview of how to do this).

3.2.6 Stratification of the Camera Model: Projection Matrices for Affine and Projective Worlds

Although the most intuitive way of interpreting a scene is to use Euclidean geometry, it has long been known in computer vision and computer graphics that it is actually much simpler, more efficient and generally more practical to consider a metric reconstruction as a special case of a projective reconstruction. One advantage of this has already been seen in this chapter when the homogeneous notation of projective geometry was used to express the projection process in a linear manner. However, in that case, metric world structure was assumed, and it is in fact possible to achieve still further simplifications by allowing affine or even projective structure and cameras.

To understand why this should make the modelling process simpler, consider the definition of projective geometry. As already discussed in chapter 2 it is a very weak geometry, missing many basic concepts present in Euclidean geometry, such as angles, distances or parallelism. Consequently, the group of transformations that can be applied to a projective space is very general, removing the need to consider lots of special cases and special forms of transformation, e.g. orthogonal matrices for rotations.

A further advantage of producing affine or projective reconstructions is that the stronger

geometries, i.e. Euclidean geometry, share all the axioms of the weaker geometries, i.e. affine and projective. This means that any quantitative results from a stronger geometry can be used in a weaker geometry without any additional work (and vice-versa for theoretical results). Similarly, results from a weaker geometry can be used in a stronger geometry with the addition of only a little information. This flexibility is further facilitated by homogeneous notation which provides a linear means of expressing the transformations and structure of any of the three geometries. All this combined allows easy movement up and down the hierarchy of geometries, using exactly the same representation and by the addition or removal of very little information. For example, to convert a projective reconstruction to an affine reconstruction a transformation is applied to all cameras and structure that will take the plane at infinity to the form $(0, 0, 0, 1)$.

As a perfect and very important example of the advantages of the use of projective geometry, consider the case of an unknown camera calibration matrix. The calibration matrix represents a non metric transformation and so, if we were working in a metric coordinate system it would have to be kept separate from the camera and structure. On the other hand, if the reconstruction is affine or projective, the camera calibration matrix is a transformation that leaves the properties of the geometry invariant. It can simply be absorbed into the projection matrix and subsequently the unknown gauge freedom of the structure space. Indeed, when considering projective reconstructions, it is rare to consider the calibration matrices separately, but when working with metric reconstructions it is generally impossible to do otherwise.

3.2.7 Gauge Freedom

If the camera matrices are allowed to be arbitrary, that is if the world coordinate system is not fixed, then it is well known [Fau92, HGC92] that the scene may not be reconstructed more precisely than up to an arbitrary transformation of the world space. For example, with a projective world, an arbitrary projectivity T can be applied to both the structure X and cameras P as:

$$\hat{P} \simeq P T^{-1}$$

$$\hat{\mathbf{X}} \simeq T \mathbf{X}$$

If the modified structure and cameras are then used for projection, the transformation T can be seen to cancel out:

$$\mathbf{x} \simeq PT^{-1}T\mathbf{X}$$

meaning that we can apply any projectivity T we desire without affecting the reconstruction. Hence, any Euclidean, metric, affine or projective reconstruction can be altered by a transformation of the relevant geometry without altering the projected properties.

Throughout the rest of this text, this gauge freedom will be assumed unless otherwise stated. This is sensible for any conceivable situation, because even if the world coordinate system has been fixed, it is still reasonable to alter the world coordinate system, and then transfer it back later.

3.2.8 Alternative Decompositions of the Camera Matrix

It should now be clear that, because the matrix based camera representation just introduced is expressed using homogeneous notation, it can be used to represent projection of Euclidean, metric, affine or projective worlds. This high degree of versatility means there are many ways to decompose the camera matrix, other than the simple one just given in equation 3.4 on page 48. A small selection of example decompositions are given here, selected because they will be used throughout the rest of the text. Because some of these decompositions may seem bizarre or useless at first, where appropriate, a brief indication of what they will be used for will be given.

Block Form

In general, when working with a camera matrix, particularly in the projective case a complete decomposition of the camera matrix is not used and it is simply treated as 11 unknowns and an arbitrary scale factor. Sometimes, for mathematical convenience, the camera matrix is considered in block form as a 3x3 matrix and a vector:

$$P = [A_{3\times 3}|\mathbf{a}]$$

This decomposition will often be used when attempting to consider in detail equations involving P because it allows such equations to be broken into parts. In fact, this decomposition will be used for this purpose in the next sub-section.

Decomposition in Terms of Camera Centre

Using the property that the mapping of a camera is not defined at its centre of projection (there are no longer two world points to define a ray with), then given the cameras centre as $\mathbf{T} = (\mathbf{t}, 1)^T$ it follows that $P\mathbf{T} = 0$. Writing P in block form as $P = [A|\mathbf{a}]$ and applying that the centre of projection is the null-space gives $A\mathbf{t} + \mathbf{a} = 0$, and so $\mathbf{a} = -A\mathbf{t}$ meaning we can write P in the form:

$$P = [A| - A\mathbf{t}]$$

Note that this enforces the camera centre to be finite (i.e. it is not on the plane at infinity). What it means if the camera is infinite will be discussed in section 3.2.9 below. The main advantage of this decomposition is that it makes the cameras centre explicit and hence easy to find. This will find almost immediate use in deriving and considering the matrix used to express relations between two views. It should also be noted that this decomposition is suitable for Euclidean, metric, affine or projective cameras, and indeed, in section 3.2.3 above a similar centre based decomposition was introduced; it has been generalised here.

Canonical Form

Originally due to [LV94], in this parameterisation the gauge freedom of the reconstruction is used to simplify the form of the first camera matrix, by aligning the first camera matrix with the canonical projective basis. Assuming either a projective or affine reconstruction, or in the metric case that all image points have been normalised to remove the calibration matrix, then the projective basis can be aligned with the first camera matrix P_1 to give a set of camera matrices of the form:

$$P_1 \simeq [I_{3\times3}|\mathbf{0}_3]$$
$$P_n \simeq [A_n|\mathbf{a}_n]$$

Using this form has the benefit of removing some or all of the gauge freedom in a reconstruction depending on which geometry is used to represent the world. However, it does come with a practical drawback when cameras can only be determined subject to a certain degree of error. In that case, this parameterisation will favour the first image, because by using convention to align the centre of the world with the first camera, it will be located without any error.

In Euclidean, metric or affine cases, transforming a camera so that it is in canonical form has the effect of entirely removing the gauge freedom. However, in the projective case, a

camera represents only 11 parameters yet the gauge freedom is 15 parameters. This counting argument means that 4 parameters will not be accounted for. These four parameters in fact represent a change of the plane at infinity as well as an arbitrary scaling. The extra freedom can be written in matrix form as (after reducing P_1 to $[I|\mathbf{0}]$):

$$P_1 \simeq [I|\mathbf{0}] \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ a1 & a2 & a3 & \rho \end{bmatrix}$$

where $a1, a2, a3$ represent the position of the plane at infinity and ρ the scaling needed because the affine transformation is being represented in projective notation.

The metric case also presents an anomaly: because the calibration matrix K is not a metric transformation, it cannot be absorbed by the gauge freedom. Consequently, K is made explicit and the canonical form is written as:

$$P_1 \simeq K_1 [I_{3\times3}|\mathbf{0}_3]$$
$$P_n \simeq K_n [R_n| - R_n\mathbf{t}_n]$$

The canonical form will often be used throughout this text for many different reasons, usually for purposes of simplification. It can be achieved very simply in practice by multiplying all camera matrices with the pseudo inverse of the first camera matrix, i.e. $P_1P^+ = [I_{3\times3}|\mathbf{0}_3]$. Note that this assumes a camera matrix as being a 4x4 matrix, with an extra row of the form $(0, 0, 0, 1)$ added. This is just considering the image space of a camera matrix as being embedded within a higher dimension (i.e. a plane in $\mathcal{P}^3$), and fixing the extra freedom (i.e. which plane) for mathematical convenience.

Stratification

As already discussed, the 12 parameters of a projection matrix can be used to represent a projective, affine, metric or Euclidean camera. However, to represent an affine, metric or Euclidean camera using projective notation, certain restrictions have to be imposed on the form of the transformations that can be applied to the camera matrix and structure. One way of doing this is to consider each camera as being a metric camera matrix decomposed as in equation 3.4, page 48 that has been subjected to a general transformation (change of basis) of the relevant geometry.

For a projection matrix working with a fully Euclidean world, this change of basis is manifest as an arbitrary rotation and translation (and a scaling for a metric world). This means that given a reconstruction, the camera matrices can be altered by a transformation composed of a rigid rotation $\hat{R}$ and translation $\hat{\mathbf{t}}$ without altering the reconstruction (provided the structure is also updated by the inverse transformation):

$$P_E = K\,[R|\mathbf{t}] \begin{bmatrix} \hat{R}_{3\times3} & \hat{\mathbf{t}} \\ \mathbf{0}_3^T & \rho \end{bmatrix}$$

On the other hand if the produced reconstruction were of an affine world, then the absolute conic would no longer be fixed and the rotation and translation of the above equation would no longer be anything more than an arbitrary 3×4 matrix, that in this case will be decomposed as $[H_{3\times3}|\mathbf{h}]$. For affine transformations, the plane at infinity must remain fixed, enforcing zeros in the bottom row of the transformation. As such, the complete transformation will take the form (ρ is the scale factor):

$$P_A = K\,[R|\mathbf{t}] \begin{bmatrix} H_{3\times3} & \mathbf{h} \\ \mathbf{0}_3^T & 1 \end{bmatrix}$$

In its weakest form, the projection matrix P_P works with a fully projective world. A change of basis in a projective world can be any 4×4 projectivity H. This means that

$$P_P = K\,[R|\mathbf{t}] \begin{bmatrix} H_{3\times3} & \mathbf{h} \\ \mathbf{a}^T & \rho \end{bmatrix}$$

where $\mathbf{a} = (a_1, a_2, a_3)$ represents a transformation of the plane at infinity and ρ an arbitrary scaling.

Although this representation is somewhat cumbersome since it includes the gauge freedom explicitly, it will prove to be invaluable for approaching the problem of merging reconstructions differing by a gauge freedom. It also proves invaluable for considering self-calibration, however self-calibration will not be addressed in detail in this work.

3.2.9 Other Camera Models

There are many variations on the central projection camera model, most of which represent more restricted forms of the full perspective model just given. In general, these forms of camera model can be separated into finite and infinite cameras where an infinite camera has a camera centre on the plane at infinity and a finite camera does not. Throughout this work,

full perspective cameras will usually be assumed, although occasionally a finite camera will be required, particularly when using certain camera decompositions.

For a camera to be infinite, the centre of the camera must lie on the plane at infinity. For this to be true, the camera must have a centre of the form $\mathbf{C} = (\mathbf{c}, 0)$. Consequently, given $P = [A|\mathbf{a}]$, the centre is given by the null-space as $A\mathbf{c} = 0$ and so, for this equation to have a solution, A must be singular. This means that for a camera to be finite A must be non-singular (i.e. $\det A = 0$).

Infinite cameras can be further classified into two types - affine and non-affine cameras. Affine cameras have a third row of the form $(0, 0, 0, 1)$ giving a camera matrix of the following form:

$$P \simeq \begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

which maps infinite points to infinite points (hence the name affine camera). The big advantage of this simplification is that all scale factors λ in the projection equation $\lambda\mathbf{x} = P\mathbf{X}$ become 1, greatly simplifying the projection equation. However for the affine model to be a good approximation to the projection process, the distance of each point from the optical axis must be small (i.e. a small field of view) and there must be little depth variation in each image.

Although alternative models are not used in this work, they can be extremely useful in some cases, particularly for purposes of simplifying the reconstruction problem at the cost of generality. For further details of many different affine and non-affine infinite cameras, the reader is referred to other works, such as [HZ00].

3.2.10 Summary

This section has presented the full perspective camera model. This model maps a region of $\mathcal{R}^3$ to $\mathcal{R}^2$, which for mathematical and practical convenience has been extended in this case to be between $\mathcal{P}^3$ and $\mathcal{P}^2$. This mapping is undefined at the camera's centre of projection.

The projection process is represented by a 3x4 matrix P of rank 3, known as the projection matrix. This matrix transforms object space points (4 vectors) to image space points (3 vectors) by the equation $\mathbf{x} \simeq P\mathbf{X}$. The projection matrix is only defined subject to a non-zero scale factor, and so it has only 11 independent entries. These entries allow for the modelling of the camera's relative position and orientation in object space, and, in the

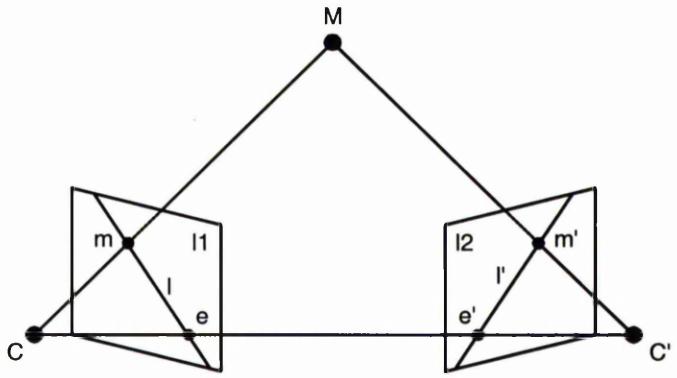

Figure 3.3: Geometry of two views - the epipolar geometry

affine or projective case, the intrinsic effects of the camera. In general, any set of cameras is considered to be subject to a gauge freedom. If an affine or projective world is being considered, then internal camera parameters as described in section 3.2.2 can be included in the gauge freedom of the reconstruction and not determined explicitly.

3.3 Two View Geometry: The Epipolar Geometry

Now that the model for a single image of a scene has been addressed, it is possible to move on and consider the case when there are two views of the same scene. The addition of an extra view creates new relationships between image points which are extremely important to most of the processes involved in multiple view image analysis. The geometry of two views is often referred to as the epipolar geometry.

Before discussing the geometry of two views, it is worth taking note of a new convention in notation. When working with two view geometry, priming will be used to indicate quantities in the second image. For example, camera 1 would be denoted P and camera 2 P'. When working with more than two views, priming will be dropped in favour of explicit numbering, e.g. P_n. When there is a need to express both a point number j and an image number i a point will be written m^i_j.

Figure 3.3 gives a pictorial representation of the central projection of a point M onto two image regions I1 and I2 by two cameras with centres of projection C and C'. Given only I2, all that can be determined about M from its image $\mathbf{x}'$ is that M must lie on the infinite line defined by O' and $\mathbf{x}'$ (the back projection of $\mathbf{x}'$). As a direct consequence of this, given I1

as well, it is known that the matching projection of M (i.e. $\mathbf{x}$) must lie on the projection of the line $< O', M >$ in I1.

The projection of $< O', M >$ in image 1 is also known as an epipolar line and it can be seen that all points m'_k in $I2$ will generate a pencil of epipolar lines in image 1 all containing the point $\mathbf{e}$. The point $\mathbf{e}$ is known as the epipole and is the image of the point O' (the 2nd cameras centre), the only point common to all back projected lines in image 2 (it must be common to all points because it is the centre of projection). It is natural to consider the plane defined by the centre of both cameras and a scene point. Any point on this plane, often known as the epipolar plane, will share the same pair of epipolar lines in both images with any other point on the plane. As such, the epipolar plane illustrates the ambiguity present in the geometric information of two images when only one projection of a point is known.

This mapping of points in one image to lines in the other image is known as the epipolar constraint and can be represented algebraically by a 3x3 rank 2 matrix F known as the fundamental matrix (due to [Fau92, Har92]). If calibrated image points are used, then the equivalent matrix is known as the essential matrix (due to Longuet-Higgins [LH81]). If $\mathbf{x}_i \longleftrightarrow \mathbf{x}'_i$ are a set of matching points lying on lines $\mathbf{l}_i$ and $\mathbf{l}'_i$ respectively, then:

$$F\mathbf{x}_i \simeq \mathbf{l}'_i \tag{3.5}$$

$$F^T\mathbf{x}'_i \simeq \mathbf{l}_i \tag{3.6}$$

If it is then imposed that the corresponding point must lie on the transferred line, it is possible to write that $\mathbf{x}'^T_i F^T \mathbf{x}_i = 0$. This relation shows that it is possible to express the relationship between a match across two images in a linear manner. An equivalent linear expression can also be found for three or four images [Har98, HZ00, Har97], and such constraints are often referred to as the multilinear constraints. These multilinear constraints will prove to be very useful for the practical application of multiple view image analysis.

3.3.1 The Essential Matrix

Since it is most intuitive to consider the cameras in terms of rotations and translations, a derivation will be given for the essential matrix E first. The essential matrix assumes a normalised coordinate system as discussed in section 3.2.2. In this coordinate system, the

projection of the 3D structure $\mathbf{X} = (\mathbf{x}, 1)^T$ into the two images as $\mathbf{x}_1, \mathbf{x}_2$ can be written as:

$$\lambda_1 \mathbf{x} = P\mathbf{X} = \left[R^T | -R^T\mathbf{t}\right] \mathbf{X} \tag{3.7}$$

$$\lambda_2 \mathbf{x}' = P'\mathbf{X} = [I_{3\times3}|\mathbf{0}]\,\mathbf{X} \tag{3.8}$$

when the coordinate system is aligned with the second camera P'. Equation 3.8 reduces to $\mathbf{X} = \lambda_2\mathbf{x}'$, which substituted into equation 3.7 gives

$$\lambda_1 \mathbf{x} = R^T\lambda_2\mathbf{x}' - R^T\mathbf{t}$$

Multiplying by R and combining the two scale factors into one (λ) gives:

$$\lambda R\mathbf{x} = \mathbf{x}' - \mathbf{t}$$

Performing a cross product with $\mathbf{t}$ to eliminate $-\mathbf{t}$ on the right:

$$\lambda\mathbf{t} \times (R\mathbf{x}) = \mathbf{t} \times \mathbf{x}' \tag{3.9}$$

It should be noted here that $\mathbf{t}$ is in fact equivalent to the epipole in image 2. This is because from the parameterisation of P the camera centre is $(\mathbf{t}, 1)$, which projects into image 2 as $[I_{3\times3}|0]\,[\mathbf{t}, 1] = \mathbf{t}$ hence $\mathbf{t}$ is the epipole in image 2. Using this, the above equation can be written as:

$$\underbrace{[\mathbf{t}]_\times R}_{E}\mathbf{x} \simeq \underbrace{\mathbf{t} \times \mathbf{x}'}_{\mathbf{l}'}$$

This can be seen to be equivalent to equation 3.5. The equivalent of equation 3.6 can be found by performing the same derivation but swapping P1 and P2. It can be seen that E depends on only 5 parameters: 3 for a rotation and 3 for a translation less 1 because the scale is arbitrary. An important point of note is that the structure has been eliminated, leaving a minimal representation, and meaning the gauge freedom in object space no longer needs to be considered. Equivalent derivations can be found throughout the literature on epipolar geometry (for example [LH81, LF96b]).

3.3.2 The Fundamental Matrix

The essential matrix E is associated with a normalised image coordinate system where the internal parameters of the imaging system such as focal length and camera centre have already been accounted for and their effects removed from the images. In practice, the transformation normalising the image coordinate system can simply be added into E to give

the fundamental matrix F. Given two cameras with calibration matrices K_1 and K_2, the relationship between E and F can be expressed as:

$$F \simeq K_2^{-1} E K_1^{-1}$$

Since the essential matrix is a fundamental matrix, with an identity mapping for the calibration matrices, it follows that the essential matrix can easily replace the fundamental matrix in any of the following discussions.

In keeping with the notion that the cameras can work with Euclidean, metric, affine or projective spaces, the fundamental matrix will now also be derived using projective cameras (following [Har95c]). Note that this proof is much more general, and applicable to a world described using any of the natural geometries. More precisely, it is going to be shown that, given two projective cameras represented by the matrices $P = (A| - A\mathbf{t})$ and $P' = (A'| - A'\mathbf{t}')$, the corresponding fundamental matrix is given by:

$$F = \left[A'\left(\mathbf{t} - \mathbf{t}'\right)\right]_\times \left(A'A^{-1}\right) \tag{3.10}$$

This can be proved by considering that the epipolar line $\mathbf{l}'$ will be the projection of the line $< O, m >$ in image 2 (as in figure 3.3 above). The image of this line can be determined by projecting two points on $< O, m >$.

The first point we can use is the camera centre which, because of the parameterisation of the camera, is $\mathbf{c} = (\mathbf{t}^T, 1)^T$. Projecting this through camera 2, we get the epipole:

$$\mathbf{e}' = (A'| - A'\mathbf{t}') \begin{bmatrix} \mathbf{t} \\ 1 \end{bmatrix} = A'\mathbf{t} - A'\mathbf{t}' \tag{3.11}$$

The point at infinity of the line $< O, M >$ can also be projected:

$$(A'| - A'\mathbf{t}') \begin{bmatrix} A^{-1}\mathbf{x} \\ 0 \end{bmatrix} = A'A^{-1}\mathbf{x}$$

Taking the cross product of these two points gives the projected line:

$$l' = (A'\mathbf{t} - A'\mathbf{t}') \times \left(A'A^{-1}\mathbf{x}\right) = \underbrace{\left[A'\left(\mathbf{t} - \mathbf{t}'\right)\right]_\times \left(A'A^{-1}\right)}_{F} \mathbf{x} \tag{3.12}$$

It can easily be seen that the resultant matrix F will be of rank 2 because it contains a skew symmetric matrix guaranteed by its form to be of rank 2.

Again, the equivalent fundamental matrix for transfer from image 2 to image 1 can be found by performing the previous derivation, but swapping P and P'. To the best of the author's knowledge, credit for this elegant derivation should go to Faugeras and can be found in [Har95c]. Naturally, there are many alternative ways to derive the fundamental matrix - see for example [LF96b, Fau92, Har92, ZX97] for a small selection.

3.3.3 Summary

When two images of the same scene are available, it becomes possible to define a 3×3 matrix F known as the fundamental matrix - that transfers points in one image to lines (epipolar lines) in the other. This process is linear in terms of F and does not involve any world structure. The fundamental matrix has the following properties:

- The fundamental matrix is subject to the constraint that it has rank 2 and when defined in matrix terms is subject to an arbitrary scale factor. This means that, despite 9 matrix entries, it has only 7 independent parameters.

- F can be used to transfer points in both images $\mathbf{x}$,$\mathbf{x}'$ to lines in the other image $\mathbf{l}$,$\mathbf{l}'$ as:

$$F\mathbf{x} \simeq \mathbf{l}'$$
$$F^T\mathbf{x}' \simeq \mathbf{l}$$

- An epipole is defined as the image of one camera in another. In the two image case, there are two epipoles (one in each image) which can be recovered from the left and right null-spaces of F. So:

$$F\mathbf{e} = 0$$
$$F^T\mathbf{e}' = 0$$

- F can be factored as a product of the epipole and a projectivity M to give:

$$F = [\mathbf{e}']_\times M = M^* [\mathbf{e}]_\times$$

  See equation 3.12. The projectivity M can be used to transfer epipolar lines $\mathbf{l}, \mathbf{l}'$ between images as $\mathbf{l} = \mathbf{l}'M$. This particular factoring of F also has the advantage of enforcing that the rank of F is 2 since the skew symmetric matrix must also be of rank 2.

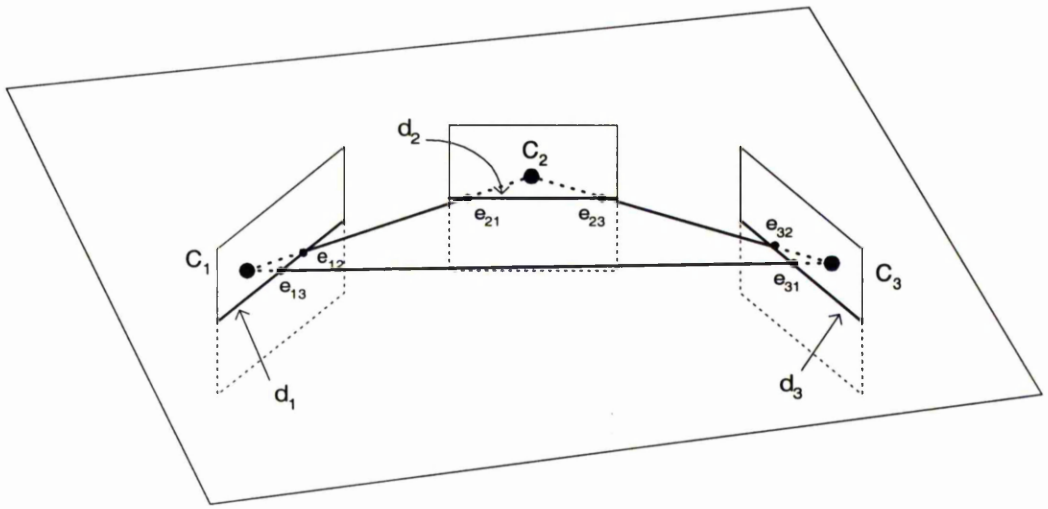

Figure 3.4: The geometry of three views

- The analogous matrix for working with a normalised image coordinate system is the essential matrix E. This depends on 5 parameters and can be decomposed in a similar manner to the fundamental matrix as:

$$E \simeq [\mathbf{t}]_\times R$$

for some rotation R and translation $\mathbf{t}$.

3.4 Three View Geometry: the Trifocal Tensor

The addition of a third view gives rise to a much more complete geometry, as illustrated in figure 3.4. Across three views, there are three cameras $P_i, i = 1, 2, 3$ and three fundamental matrices F_{12}, F_{13}, F_{23}. Of interest is the trifocal plane, defined by the three camera centres C_1, C_2, C_3, which is simultaneously an epipolar plane for each pair of cameras. With three images, there are now two epipoles in each image, one epipole for the image of each of the other camera centres. Since the trifocal plane contains all the camera centres, it follows that, from their definition, all epipoles must lie on the lines of intersection of the trifocal plane with the images d_1, d_2, d_3.

The importance of the addition of a third view is very significant from the point of both robustness and accuracy. Whereas in the two view case, a point in one image could be constrained to lie on a line in the other image, with three views a point in two views is

constrained to an exact point in the third view. It is fairly clear how this transfer can be affected. When using projection matrices, it is possible to reconstruct a point using two views and then project it into the third. When using fundamental matrices, a point can be constrained to lie on a line in the third image by both the other images it appears in. These lines will intersect at a single point provided they are not coincident (the possibility of coincidence is a drawback of using fundamental matrices for transfer in the triplet case).

3.4.1 Euclidean Cameras

The most intuitive way to interpret the three view geometry is to consider it in Euclidean or metric terms. Assuming a normalised image coordinate system, the three camera matrices can be written in the form $P_n = [R_n|\mathbf{t}_n]$, each consisting of a world rotation and translation resulting in a system involving 18 parameters. However, since the world coordinate system is subject to a gauge freedom of 7 parameters (a rotation, translation and scaling) it follows that there are only 11 parameters fully describing the triplet.

This has implications on the form of the three underlying essential matrices. Since between them they represent 15 parameters, they must be subject to further constraints. These constraints are easy to understand in the Euclidean sense, since the definition of an essential matrix is $E \simeq [\mathrm{t}]_\times R$ for some rotation R and translation $\mathbf{t}$. It can be seen that it is possible to remove one essential matrix entirely since it can be determined as the composition of the transformations involved in the other two. However, to do that it is necessary that all the essential matrices have the same relative scaling. Making the relative scale factors explicit gives a definition for the three essential matrices of:

$$E_{12} = \lambda_{12} [\mathbf{t}_{12}]_\times R_{12}$$
$$E_{23} = \lambda_{23} [\mathbf{t}_{23}]_\times R_{23}$$
$$E_{13} = \lambda_{13} [\mathbf{t}_{13}]_\times R_{13}$$

The arbitrary scaling of the world coordinate system is then selected so that $\lambda_{12} = 1$. It follows that E_{12} is described by 5 parameters, and E_{23}, E_{13} are both described by 5 parameters plus 1 extra to give them the same relative scale. Now including the relative scales, it is possible to define one essential matrix in terms of the other two, e.g. for E_{13}:

$$E_{13} = [\mathbf{t}_{12} + \lambda_{23}\mathbf{t}_{23}]_\times R_{12}\lambda_{23}R_{23}$$

leaving only 11 parameters as required.

3.4.2 Projective Cameras

In this case, three camera matrices represent 33 unknowns, and the gauge freedom in the world coordinate system can eliminate 15 unknowns, leaving 18 parameters to fully describe the geometry. If this is related to the fundamental matrices, it can be seen that three fundamental matrices give 21 unknowns leaving 3 surplus parameters to be accounted for.

One way to account for these parameters is to return to the discussion of the trifocal plane defined by all 3 camera centres. It can be seen that the fact all epipoles are the intersection of the trifocal plane with the images implies certain extra constraints. In particular, if the fundamental matrix is used to transfer an epipole to an epipolar line in another image, then that epipolar line should be the line passing through both epipoles in that image. Using the notation from figure 3.4 above, this results in 12 constraints - 4 for each image and 2 each for the transfer of each epipole to the other two images. Of these constraints only three are independent - for example, the following three:

$$F_{12}\mathbf{e}_{13} = \mathbf{d}_2 = \mathbf{e}_{21} \wedge \mathbf{e}_{23}$$
$$F_{31}\mathbf{e}_{32} = \mathbf{d}_1 = \mathbf{e}_{12} \wedge \mathbf{e}_{13}$$
$$F_{23}\mathbf{e}_{21} = \mathbf{d}_3 = \mathbf{e}_{32} \wedge \mathbf{e}_{31}$$

Considering these three constraints and the fundamental matrices, the complete system is described by 18 parameters. A similar and more in depth discussion of the concepts in this section can be found in [Fau92].

3.4.3 The Trifocal Tensor

So far, the geometry of three views has been considered in terms of the epipolar geometry and projection matrices. Both of these representations have drawbacks. When using projection matrices, it is necessary to consider world structure as well as cameras, complicating the whole problem. On the other hand the representation in terms of the epipolar geometry cannot be used to transfer any structure on the epipolar plane. In this case, the epipolar lines for all three images will be coincident (see figure 3.4 above) so they cannot be intersected. In practice, this also means any points that are near the trifocal plane will transfer inaccurately.

As work progressed on the geometry of triplets, linear relations were found that governed the positioning of points in 3 images. These relations are often expressed with a $3 \times 3 \times 3$ homogeneous tensor, called the trifocal tensor. With hindsight, the trifocal tensor was first

discovered in work concerning the reconstruction of lines using calibrated cameras [SA90, WA92], but was not thought of as a tensor until [LV93].

Independently, work on the equivalent system of three images for uncalibrated cameras [Sha95] introduced a set of 27 coefficients for a set of four independent linear conditions relating the coordinates of corresponding points in three views. This was clarified in [Har94a], where the constraints were derived by considering the triplet case using normalised camera matrices. These coefficients were later found to be equivalent to a 3x3x3 homogeneous tensor that will be symbolised T_i^{jk} here. The tensor consists of 27 parameters, but only 18 are independent due to additional nonlinear constraints (see [PF98] for a more in depth discussion). The relationship between points matched in the three images $(\mathbf{x}', \mathbf{x}'', \mathbf{x}''')$ where $\mathbf{x} = (m^1, m^2, m^3)^T$ can be expressed in tensor notation as:

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

Here, the standard convention of sumation over indices, repeated in both upper and lower positions, is used. This equation cancels out on the left for $i = j$ or $l = m$, and swapping i and j or l and m simply changes the sign of the equation.

A similar constraint exists for lines, and a triplet of lines $\mathbf{l}, \mathbf{l}', \mathbf{l}''$ is subject to:

$$\mathbf{l}_i \simeq \mathbf{l}'_j \mathbf{l}''_k T_i^{jk}$$

It has been shown [Har94a, Har94c, Har97] that the tensor is also closely related to a triplet of projection matrices in canonical form, i.e. $P_1 \simeq [I|\mathbf{0}], P_2 \simeq [A|\mathbf{a}], P_3 \simeq [B|\mathbf{b}]$ as:

$$T_i^{jk} \simeq A_i^j \mathbf{b}^k - \mathbf{a}^j B_i^k$$

Although the trifocal tensor will be mentioned throughout this work, it will not receive a great deal of theoretical attention and so will not be discussed further here. It will later be shown that for practical estimation of the geometry of image triplets, the cameras and structure approach is good as if not better than, the trifocal tensor. However, this does not depreciate the value of the trifocal tensor as a tool for understanding the geometry of image triplets. The interested reader is referred to the body of work referenced above, particularly [Har97, PF98, HZ00] for a full description.

3.5 Four View Geometry: The Quadrifocal Tensor

The last multiple view relations of any importance covers four images and is expressed by the quadrifocal tensor. This tensor provides 5 independent constraints per point and marks

the maximum number of images for which linear constraints can be determined. In [Har98], a method for computing this tensor was given, but since the quadrifocal tensor has no direct relevance to the remainder of this work it will not receive further attention here. The interested reader is referred to [SW00] for more details.

3.6 Multiple View Geometry and Inter Image Homographies

The homographies to be discussed in this section are a special form of planar projectivites from $\mathcal{P}^2 \rightarrow \mathcal{P}^2$ which describe the transformation from one plane to another. Of particular interest are inter image homographies that transfer the images of points on a particular world plane from one image to another. Such transformations are useful in their own right, particularly for tasks such as point matching, but a detailed analysis of inter image homographies in relation to cameras and multiple view geometry provides other extremely useful results.

In particular, this analysis will allow projective and affine camera matrices to easily be related to the fundamental matrix. Whilst this discussion can also be generalised to cover fully Euclidean cameras, it is not necessary to do so because in the Euclidean case both the camera matrix and essential matrix can be decomposed in terms of rotations and translations. Consequently, different methods should be applied (see [HZ00]).

The derivations in the following sections are spread widely throughout the literature, but in this case are reworked versions of those found in [HZ00, Pol99, LV94]. In particular, most of the groundwork for the following sections was laid down by Luong and Viéville [LV94].

3.6.1 Inter Image Homographies

To illustrate inter image homographies, it is best to start by considering the projection of points $\mathbf{M}_{\Pi}$ belonging to a plane Π in $\mathcal{P}^3$ into an image i (also a plane in $\mathcal{P}^3$) to give points $\mathbf{m}_{\Pi i}$. This can be considered to be equivalent to transferring points between planes in $\mathcal{P}^3$. Considering the camera projection matrix P_i in block form as $P_i = [A_i|\mathbf{a}_i]$, the projection process can be written as:

$$\mathbf{m}_{\Pi i} \simeq P\mathbf{M}_{\Pi} \simeq [A_i|\mathbf{a}_i]\,\mathbf{M}_{\Pi}$$

However, this does not express the fact that $\mathbf{M}_{\mathbf{\Pi}}$ must belong to the plane $\mathbf{\Pi}$. This can be enforced by using the standard equation for a plane, which states that if a point $\mathbf{M}_{\mathbf{\Pi}} \simeq \left(\mathbf{x}_{\pi}^{T}, 1\right)$ belongs to a plane $\mathbf{\Pi} \simeq \left(\pi^{T}, 1\right)$ then $\mathbf{\Pi}^{T}\mathbf{M}_{\mathbf{\Pi}} = \pi^{T}\mathbf{m}_{\mathbf{\Pi}} + 1 = 0$. Substituting $-\pi^{T}\mathbf{m}_{\mathbf{\Pi}}$ for the 1 in $\mathbf{M}_{\mathbf{\Pi}}$ means the point $\mathbf{M}_{\mathbf{\Pi}}$ must be part of the plane $\mathbf{\Pi}$:

$$\mathbf{M}_{\mathbf{\Pi}} \simeq \begin{bmatrix} \mathbf{m}_{\mathbf{\Pi}} \\ 1 \end{bmatrix} \mathbf{M}_{\mathbf{\Pi}} \simeq \begin{bmatrix} \mathbf{m}_{\mathbf{\Pi}} \\ -\pi^{T}\mathbf{m}_{\mathbf{\Pi}} \end{bmatrix} \simeq \begin{bmatrix} I_{3\times 3} \\ -\pi^{T} \end{bmatrix} \mathbf{m}_{\mathbf{\Pi}}$$

which makes it possible to rewrite the projection equation as:

$$\mathbf{m}_{\mathbf{\Pi}i} \simeq [A_i|\mathbf{a}_i] \begin{bmatrix} I_{3\times 3} \\ -\pi^{T} \end{bmatrix} \mathbf{m}_{\mathbf{\Pi}} \simeq \left[A_i - \mathbf{a}_i\pi^{T}\right] \mathbf{m}_{\mathbf{\Pi}} \tag{3.13}$$

It follows that the homography transferring points on a world plane onto the image plane is given by $H_{\Pi i} \simeq A_i - \mathbf{a}_i\pi^{T}$. Note that, in the specific case of Π being the plane at infinity $\Pi = [0, 0, 0, 1]$, the homography has the simpler form $H_{\Pi i} \simeq A_i$. Also note that this homography transfers points in the coordinate system of the relevant plane (i.e. 3 vectors) and not in the world coordinate system (4 vectors).

Given the above definition, it is simple to obtain the homography that transfers projections of points in the world ($\mathcal{P}^3$) between images. For transfer from image i to j this homography for a particular plane Π can be determined as $H_{ij}^{\Pi} = H_{\Pi j}H_{\Pi i}^{-1}$, i.e. transfer the point from image i to the plane Π in the world and then from this plane into image j. It is worth noting that this is independent of a change of basis in $\mathcal{P}^3$ since if T is a change in basis then H_{ij}^{Π} becomes $H_{\Pi j}TT^{-1}H_{\Pi i}^{-1}$, effectively cancelling T out.

3.6.2 Relation to Camera Projection Matrices

When considering projection from an affine or projective world, the first camera may be aligned with the standard basis to give $P_1 = [I_{3\times 3}|\mathbf{0}_3]$. In this case, any homography for transferring points from a plane Π onto the first image plane will have the form $H_{\Pi 1} = I_{3\times 3}$, and thus $H_{1i}^{\Pi} = H_{\Pi 1}$. Subsequently, the projection matrix for image i can be factored as a homography due to some reference plane REF and the projection of the centre of camera 1 in camera i, $\mathbf{e}_{1i}$:

$$P_i = \left[H_{1i}^{REF}|\mathbf{e}_{1i}\right] \tag{3.14}$$

$\mathbf{e}_{1i}$ is actually the epipole of the fundamental matrix between images 1 and i, i.e. F_{1i}. This is because the epipole is the image of camera i's centre, and, given the special form of P_1, the

camera centre will reconstruct as $(\mathbf{e}_{1i}, 1)$. The new notation H_{1i}^{REF} indicates a homography transferring points on plane REF from image 1 to image i. Note that, for the affine case, the plane REF will be the plane at infinity. Overall, this is really just another way of interpreting the camera matrix.

An important relationship between inter image homographies and projection matrices can be seen by relating the projection matrix in equation 3.14 above and an arbitrary plane $\mathbf{\Pi} \simeq [\pi^T, 1]$ defined as in equation 3.13:

$$H_{1i}^{\Pi} \simeq H_{1i}^{REF} - \mathbf{e}_{1i}\pi^T$$

This relationship shows how the homography used in the camera matrix in equation 3.14 can be made to correspond to any reference plane π. The implications of this for a projective reconstruction are that these three parameters are totally arbitrary. However, for an affine reconstruction, the arbitrary reference plane REF in the projective reconstruction must correspond to the plane at infinity.

3.6.3 Relation to Fundamental Matrix

There also exists a very useful relationship between homographies and fundamental matrices. Given the image of a point on a plane Π in image i, $\mathbf{x}_i^{\Pi}$, and some inter image homography transferring such points to an image j H_{ij}^{Π}, it follows that that the image of the same point in image j will be given by $\mathbf{x}_j \simeq H_{ij}^{\Pi}\mathbf{x}_i$. Plugging this into the epipolar constraint gives:

$$\mathbf{x}_j^T F_{ij}\mathbf{x}_i = \left(H_{ij}^{\Pi}\mathbf{x}_i\right)^T F_{ij}\mathbf{x}_i = 0 \tag{3.15}$$

Since the fundamental matrix maps points to epipolar lines as $F_{ij}\mathbf{x}_i \simeq \mathbf{x}_j \times \mathbf{e}_{ij}$, equation 3.15 is equivalent to $\mathbf{x}_j^T [\mathbf{e}_{ij}]_\times H_{ij}^{\Pi}\mathbf{x}_i = 0$. Identifying this with the original epipolar constraint $\mathbf{x}_j^T F_{ij}\mathbf{x}_i = 0$, and realising that equation 3.15 is valid for any point in image i $\mathbf{x}_i$, F can be defined as:

$$F \simeq [\mathbf{e}_{ij}]_\times H_{ij}^{\Pi}$$

Consider two images i and j, and a plane Π formed by back-projecting a line in image j, $\mathbf{l}_j$. If a point on the plane Π is projected into image i as $\mathbf{x}_i$, then the corresponding point in image j must lie on the corresponding epipolar line, $F_{ij}\mathbf{x}_i$. Since the corresponding point $\mathbf{x}_{\pi j}$ must also lie on the line $\mathbf{l}_j$, it is possible to identify the exact point as $\mathbf{l}_j \times F_{ij}\mathbf{x}_i$, provided that $\mathbf{l}_j$ is not coincident with the epipolar line. This means that the homography H_{ij}^{Π} is equivalent to $\mathbf{l}_j \times F_{ij}$ (i.e. a plane can be used to constrain the fundamental matrix to an

exact equivalence). The problem of coincidence can easily be avoided when $\mathbf{l}_j$ is $\mathbf{e}_{ij}$ since the line $\mathbf{e}_{ij}$ cannot pass through the epipole (i.e. $\mathbf{e}_{ij}^T\mathbf{e}_{ij} \neq 0$). As a consequence of all this:

$$[\mathbf{e}_{ij}]_\times F_{ij}$$

corresponds to the homography of a plane. Identifying this with the previous equations 3.14 and 3.13, it follows that it is possible to write the projection matrices for two views as:

$$P_1 = [I_{3\times 3}|\mathbf{0}_3^T]$$
$$P_2 = [[\mathbf{e}_{12}]_\times F_{12} - \mathbf{e}_{12}\pi^T|\mathbf{e}_{12}]$$

Note that this is a very important result that allows a projective camera setup to be determined directly from the fundamental matrix. Note that it is also subject to four degrees of freedom - 3 for the arbitrary plane π and one extra for the relative scale between F_{12} and e_{12}. Because of these arbitrary values, it is not possible to use this to directly create any more than two cameras in a projective frame, since these arbitrary values must be consistent (i.e. refer to the same plane) for a set of cameras to be in the same coordinate frame.

3.6.4 Summary

This section has provided a small number of results concerning the relationships between camera matrices, the fundamental matrix and inter image homographies. In summary, the main results are:

1. It has been shown in section 3.6.1 that, given a plane Π in $\mathcal{P}^3$ and two cameras P_1 and P_2, there is a homography written H_{12}^{Π} which can be used to transfer points on the plane Π from one image $\mathbf{x}_1$ to the other $\mathbf{x}_2$ as:

$$\mathbf{x}_2 \simeq H_{12}^{\Pi}\mathbf{x}_1$$

2. The projection matrix for an image P_i can be factored as a homography due to some reference plane REF and the projection of the centre of camera 1 in camera i, the epipole e_{1i} provided that P_1 is in canonical form:

$$P_1 = [I_{3\times 3}|\mathbf{0}]$$
$$P_i = [H_{1i}^{REF}|\mathbf{e}_{1i}]$$

This result was derived in section 3.6.2.

3. Given a fundamental matrix between two images F_{12}, with epipoles $\mathbf{e}_{12}$ and $\mathbf{e}_{21}$, the corresponding projection matrices P_1 and P_2 for both images can be written as:

$$P_1 = \left[I_{3\times 3}|\mathbf{0}_3^T\right]$$
$$P_2 = \left[\left[\mathbf{e}_{12}\right]_\times F_{12} - \mathbf{e}_{12}\pi^T|\mathbf{e}_{12}\right]$$

for some 3 vector π giving the position of the plane at infinity. In the projective case, we are free to set the plane at infinity to anything we desire, so π is arbitrary. This important result was derived in section 3.6.3 above.

3.7 Orientation

In this section, a very brief discussion of oriented projective representations for images and multiple view relations will be given. The notion of oriented projective geometry was introduced in section 2.4.2, page 36 and is the refinement of a projective space by the introduction of the notion that some points are in front of a plane and others are behind. This is enforced in homogeneous notation by changing the scale factor constraint, so that all homogeneous quantities are subject to a non zero positive scale factor rather than a non zero scale factor.

In the context of modelling cameras and structure, it proves useful because it can be used to improve a standard projective reconstruction by distinguishing between points in front of and behind a camera. This can be achieved using the constraint that all points known to be visible in an image must be in front of the camera and expressed in the reconstruction of the 3D points by enforcing the convention that points in front of the camera project to a positive scale factor and those behind to a negative (and hence invalid) scale factor. For more information relating to the application of oriented projective geometry to multiple view geometry, see [Har93, Lav96, LF96a].

3.7.1 Oriented Two View Geometry

The notion of oriented projective geometry can also be applied to the epipolar geometry. In this case, a point in one image matches to a half epipolar line in the other image instead of a full epipolar line as in the fully projective case. This is illustrated in figure 3.5 below, where two world points, $\mathbf{X}$ and $\mathbf{Y}$, on the same epipolar plane Π are projected through the camera centres $\mathbf{C}'$ and $\mathbf{C}$ to form matching image points $\mathbf{x}\leftrightarrow\mathbf{x}'$ and $\mathbf{y}\leftrightarrow\mathbf{y}'$. Since by definition the camera centres and epipoles lie on the same line (the baseline), matches are restricted to

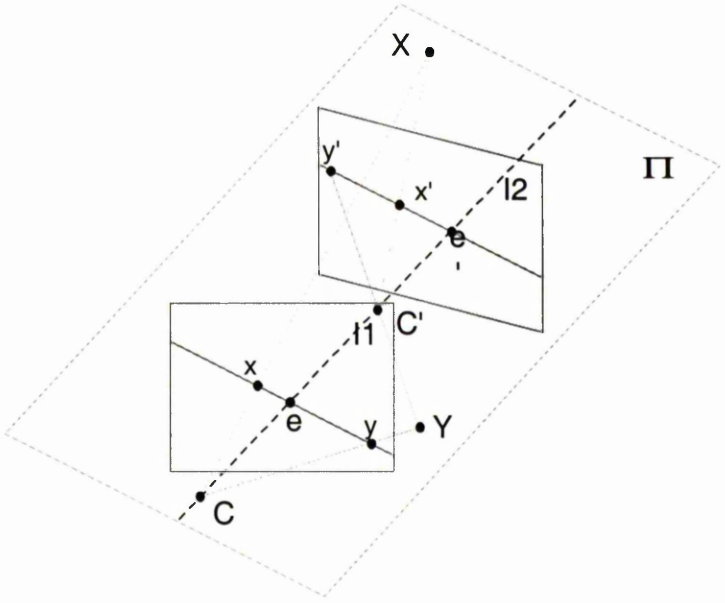

Figure 3.5: Illustration of epipolar geometry with epipoles in the images. Matching for the point M can be seen to be restricted to a half epipolar line. The point N illustrates that if the point matches to the other half of the epipolar line in $I1$ it must be behind $I2$.

being between points on the same side of the base line. If the correspondences are not on the same side of the baseline, the world point will be behind one of the cameras, as illustrated by the point $\mathbf{Y}$.

To orient the epipolar geometry, at least one match is needed. Half lines are represented by restricting all epipolar lines to have a positive scale factor. For example, the epipolar line $\mathbf{l} = (a, -b, c)$ represents only half a line, with the other half represented by the line $(-a, b, -c)$. The orientation is then enforced by using the match $\mathbf{x} \leftrightarrow \mathbf{x}'$ and finding the corresponding epipolar lines $\mathbf{l}$ and $\mathbf{l}'$. Both of these epipolar lines should transfer to exactly the same half epipolar line, and if they do not, the sign of the homography used to transfer them is changed - e.g., if $\mathbf{l}' = (-a, b, -c)$ and the homography H transfers $\mathbf{l}$ to $(a, -b, c)$ then H would be multiplied by -1. It is worth noting that, in order to avoid problems at the boundaries of quadrants, the match pair should first be Hartley-Sturm corrected (see section 5.4.3 on page 102). Alternatively, more than one match can be used. Ideally this method can be used with all the matches and used to reject any matches that are not possible because the corresponding point is behind one camera (i.e. all the points in the minority).

3.7.2 Oriented Cameras and Structure

A projective reconstruction, consisting of cameras P_i and structure $\mathbf{X}_j$, can be upgraded to an oriented projective reconstruction by insisting that the scale factors of all structure projected to the images $\mathbf{x}_j^i$ has a positive scale factor, i.e.

$$\lambda \mathbf{x}_j^i = P_i \mathbf{X}_j \text{ where } \lambda > 0$$

This can be achieved by selecting the signs of the structure and cameras (i.e. multiplying by -1), so this is always the case. An exact method for achieving this with cameras will be given in section 5.5 on page 105.

3.8 Summary

This chapter has provided a brief introduction to the concepts and theory of single and multiple view geometry. A camera notation was introduced using the tools of projective geometry, and it was shown how this notation was capable of representing cameras across the hierarchy of geometries. This camera notation was then used to derive two and three view constraints that were linear in terms of points matched between all the images and some results involving these were discussed. Finally, some very important results concerning homographies and camera matrices were discussed, leading to a very useful method for transferring representation between fundamental matrices and projection matrices without the need to determine any additional information.

3.8.1 Camera Matrices and Structure vs Multilinear forms

This chapter has given two approaches to expressing and interpreting the geometry of multiple images. Both methods have advantages and disadvantages which make them applicable to different problems. A brief summary of some of these advantages and disadvantages is given in the next two sub-sections.

Because of their different characteristics, both representations are useful in producing a practical system. The multilinear forms are very useful for obtaining an initial approximation of camera positions since they express constraints on camera positions without involving world structure. The camera and structure approach can then be used to determine any structure, and also to extend the limited number of images represented by multilinear forms to any desired extent.

Camera Matrices and Structure

Advantages:

- The ability to produce structure using this approach is naturally a benefit, since many applications require some form of structure
- Since the camera matrices are a model of the physical process of projection, it is very easy to use them to produce physically meaningful measurements. This is particularly important when some form of noise is expected to disturb observed points and it is necessary to model this noise.
- There is no limit to the number of images that can be modelled using camera matrices.

Disadvantages:

- Since a reconstruction is subject to a gauge freedom, there are redundant parameters that are fixed arbitrarily. This makes it hard to relate one reconstruction to another, and increases the effort required to produce a reconstruction in the first place.
- The projection equation $\mathbf{x} \simeq P\mathbf{X}$ is nonlinear in unknowns, if neither structure nor cameras are known. Subsequently, given only projections of structure, this makes determining either cameras or structure using the projection process very difficult.

Multilinear Forms

Advantages:

- As their name suggest, the multilinear forms provide a linear relationship between the projection of points in different images. These linear relationships do not have to include structure and so make the multilinear forms much simpler to estimate from real data.
- Multilinear forms do not make use of world structure and so do not include an arbitrary gauge freedom for the world coordinate system. This means they provide a minimal means of representing the geometry of the relevant number of images.

Disadvantages:

- The multilinear forms are limited to modelling collections of 2,3 or 4 images only. Beyond that, the forms cease to be linear and so are not particularly useful.

- It is impossible to produce any easily interpreted information on the 3D structure using just the multilinear forms. The representations that can be produced are only implicit and not as easily worked with.
- The linear relationship between points in different images has no real physical meaning, and so needs to be re-interpreted in a nonlinear manner for many applications.

Chapter 4

Estimating Multiple View Geometry

4.1 Introduction

The multiple view constraints described in chapter 3 are key to much of multiple view image analysis. Because they avoid the need for structure, they have a linear form which provides a very good means of acquiring an accurate description of the geometry of pairs or triplets of views. Consequently, they can prove very useful in the analysis of a pair or triplet of images, or as a stepping stone to build descriptions of larger sequences.

This chapter focuses on a review of methods for determining the geometry of a pair or triplet of images using the multilinear constraints. The difficulty in this process is that any real data will not conform exactly to the idealised model presented so far. This difference will be manifest as errors on the localisation of points in images and will need to be accounted for in a principled manner if good results are to be produced. Consequently, a number of algorithms are presented based on the assumption of an error that conforms to a zero mean Gaussian distribution.

4.2 Preliminaries

Before continuing to the actual problem of determining geometry, it is worth stating a few important assumptions and techniques relating to numerical stability, model fitting and the errors inherent in observing points in real images. These are applicable throughout this work and not just to determining multiple view geometry.

4.2.1 Measurement Errors and Their Distribution

Before any solution can be attempted, it is extremely important to consider how measurement errors will have affected the data. Such errors will be assumed to come from a large number of sources, particularly things such as inaccuracies in sensors used for digitisation or point location and matching algorithms.

Furthermore, it will be assumed that any measurement errors conform to a two dimensional isotropic Gaussian distribution with zero mean and uniform standard deviation. This is in general a well founded assumption with much empirical evidence to support it. In addition, the response characteristics of photometric cells have long been known to produce a Gaussian localisation error. Furthermore, the characteristics of many feature detection and matching schemes are known to produce approximately Gaussian distributed localisation errors, even to the extent that the assumption is the basis for the corner detector, e.g. [HS88]). Based on this, the safety of the Gaussian assumption will be taken as granted. Further justification is omitted due to the very long-standing nature of the assumption and the construction of many successful algorithms and systems based around it.

4.2.2 Model Fitting and Least-Squares

The basic problem to be approached in this chapter is as follows: given a set of data observed with measurement errors, and some model based on adjustable parameters, find the parameters that give the best fit for the data to the model. In other words, it is desirable to find the set of parameters that maximise the probability of the data, given the parameters. This form of parameter estimation is known as maximum likelihood estimation.

As just discussed in section 4.2.1 it is assumed that the errors corrupting observed data are Gaussian in nature. This leads to the use of the well known least-squares methods (due to [Pea01]). Given N observed data points (x_i, y_i) $i = 1 \ldots N$, these are fitted to a model that has M adjustable parameters a_j $j = 1 \ldots M$. The model predicts a relationship between the measured variables independent and dependent on the model:

$$y_i = y\left(x; a_1 \ldots a_M\right) + \epsilon_i \qquad i = 1 \ldots N$$

where ϵ_i is assumed to be some Gaussian distributed error associated with the measured point, y_i. An appropriate Gaussian function can be substituted for the set of ϵ_i and the

resultant equation simplified. Consequently, the maximum likelihood estimate of the parameters a_i that minimises the error ϵ_i can be found by minimising:

$$\sum_{i=1}^{N} [y_i - y(x_i)]^2$$

The reader is referred to appropriate statistical texts for a proof, e.g. [KS83b, DS83]. This minimisation can be achieved using any of numerous methods, such as singular value decomposition (SVD), QR or LQ factorisation, Moores-Penrose inverse or many others.

Whilst least-squares is well documented, it will be worthwhile to point out the less well known orthogonal least-squares [KS83a]. The major drawback with least-squares as just presented above is that it assumes the error exists in only one coordinate - y_i in the example above. This is dependent upon the often justified assumption that the x_i are not subject to errors. For example, if the function were measuring the relationship between the time in an experiment (x_i) and some observed value (y_i), then the time can be found subject to no significant error, but the observation will have error.

However, it may often be the case that there are errors in all coordinates - for example predicting the position of a point in one image given another image of that point. In this case, both points will be measured with error. Orthogonal least-squares assumes that there are errors in all coordinates and sets about solving a different and constrained minimisation to account for this.

For example, consider fitting a hyperplane $\mathbf{h} = (\mathbf{f}, h_p)$ to a set of j points $\mathbf{z}_i = (z_{i1}, z_{i2}, \ldots, z_{in})$ in n space. The maximum likelihood estimate for $\mathbf{f}$ can be found as minimising:

$$\sum_{i=1}^{j} \left(\mathbf{h}^T \mathbf{z}_i\right)^2 \text{ subject to } \mathbf{f}^T\mathbf{f} = 1$$

The scaling constraint ensures that the error measure is invariant to both a Euclidean transformation and a scaling of the coordinate system. This is not the case with normal least-squares. This can be performed as a constrained minimisation (see [GVL89]), and an example solution can be found in section 4.3.1 where it will be applied to fundamental matrix estimation.

4.2.3 Uncertainty Analysis

All the algorithms to be described in this and the following chapters rely on expressing uncertain input quantities in terms of parameterised models, also determined with a degree of

uncertainty (described by a covariance matrix). Consequently, the estimates of the observed independent parameters produced by the model are also subject to a degree of uncertainty. Since Gaussian distributed errors are being assumed, the uncertainty will take the form of a chi-squared distribution and should be propagated through the model to account for the errors in determining the model. Once a covariance matrix has been determined, it is possible to identify all points that have a certain percentage chance of occurring given the model, and visualise this as an ellipsoid centred on the estimated point. The shape and size of this region can be determined by the principal components of the propagated covariance matrix.

Defining such an elliptical region for a particular model estimate has many uses - for example, rejecting points that are outlying to the model (because they are not likely to have occurred), or for guiding matching. However, the exact method is fairly involved; for detailed descriptions of the propagation and determination of uncertainties, the reader is referred to [HZ00] for a more general discussion and to [CZZF96] for a discussion in relation to the fundamental matrix.

For purposes of this work, uncertainty analysis is simplified by not using error propagation, but instead assuming a very simple circular confidence region (as if there were no error in the model, only the data). Having circular confidence regions makes determination of the region and testing for data inside and outside the region very simple and efficient. The size of this circular region can easily be defined by the standard deviation of the data and a value representing the percentage confidence required (easily obtainable from mathematical tables of χ^2 probability). Note that small improvements have been observed (e.g. in [CZZF96]) by not assuming perfect models and using an elliptical confidence region. However, these improvements are not huge and do come at a computational and complexity cost.

4.2.4 Normalising Image Points and Numerical Stability

When using homogeneous points, there is a very important practical problem concerning numerical stability that needs to be addressed. Measurements in an image are usually given in terms of image pixels, so an 800x600 image will have a typical point of homogeneous form $(400, 300, 1)$. Since the first two items in this point are two orders of magnitude larger than the final one, any associated matrices calculated using points of this form will have a very bad condition number (see [Har95a] for a discussion of this in relation to the fundamental matrix).

Ideally, the average point should be of the form $(1, 1, 1)$ so as to minimise both round off errors and condition number problems (even if infinite precision arithmetic is used, poor condition numbers will produce bad results). Similarly, any linear approximations to distance measures (e.g. see equation 4.3) which are not invariant to Euclidean transformations of the points should be conditioned so as to produce more accurate approximations to the actual Euclidean distance. This can all be achieved by transforming the image coordinate frame so that the centroid of the points is at the origin and then scaling isotropically so that the average distance of a point from the origin is $\sqrt{2}$.

The transformation and its inverse taking points to and from these coordinate systems can easily be found and applied to all points prior to computation. Such transformations can also be applied to any calculated quantities to make them work in the original coordinate system, e.g. if matched points $\mathbf{x}_i \leftrightarrow \mathbf{x}'_i$ are normalised such that $\hat{\mathbf{x}}_i = T\mathbf{x}_i$ and $\hat{\mathbf{x}}'_i = T'\mathbf{x}'_i$ and a fundamental matrix $\hat{F}$ is found, then the fundamental matrix for the unnormalised points is found as $F = T'^T \hat{F} T$.

This normalisation of image points is so effective that, unless otherwise stated, it will always be assumed that it has been applied to any technique involving homogeneous quantities such as image or world points. The importance of normalisation to fundamental matrix estimation was first pointed out in [Har95a].

Furthermore, because of the scale factor constraints in homogeneous quantities, sensible decisions are always assumed on the scale of computed numerical quantities. For example, a matrix may be scaled so that it has a Euclidean norm of 1 or so that the largest item in the matrix is 1. This will also be assumed, except where it is relevant to the process in hand, e.g. for a parameterisation.

4.3 Linear Estimation of the Fundamental Matrix

This section will aim to describe the main techniques used for robustly estimating the fundamental matrix F from points matched between a pair of images. It should be recalled from the previous chapter that the fundamental matrix is a 3×3 matrix that can be used to transfer one point from a match pair $\mathbf{x} \leftrightarrow \mathbf{x}'$ to an epipolar line in the other image $\mathbf{l}, \mathbf{l}'$ containing the matching point:

$$F\mathbf{x} \simeq \mathbf{l}' \tag{4.1}$$

$$F^T\mathbf{x}' \simeq \mathbf{l} \tag{4.2}$$

It is also defined subject to an arbitrary scale factor and to the constraint that it must have a rank of 2. These two constraints mean that, although the matrix has 9 entries, only 7 are independent. In practice the problem to be addressed in this section is how to estimate F given a set of n matches $\mathbf{x}_i \leftrightarrow \mathbf{x}'_i \ \forall i \in (1, \ldots, n)$ between features, each of which has been determined subject to Gaussian distributed measurement errors.

4.3.1 The 8 Point Algorithm

The first thing that must be achieved is to find a closed form means of determining the fundamental matrix. Since the expressions for transfer using the fundamental matrix F are linear in terms of F, they form a logical starting point for a linear method to calculate F. By imposing that the transferred line should have the matching point on it, the following constraint is derived:

$$\mathbf{x}'^T F \mathbf{x} = 0 \tag{4.3}$$

Since, as will be shown below, it is possible to impose the scale factor constraint, 8 matches are required to solve for F using equation 4.3. In practice usually far more than 8 matches are available, and these match points will be assumed to be perturbed by Gaussian distributed noise. It is important to note that, in this case, it will be assumed there are no errors due to the placement of features in space, such as might occur from mismatches or moving objects. Such incorrect placements corrupt the Gaussian assumption and will be dealt with by the robust techniques to be reviewed in section 4.5.

Although the algebraic measurement errors from equation 4.3 will be physically meaningless and not Gaussian distributed, it is still close enough to be reasonable with n matches $\mathbf{x}_i \leftrightarrow \mathbf{x}'_i \ \forall i \in (1, \ldots, n)$ to use equation 4.3 and solve for F using existing linear least-squares methods to minimise:

$$\min_F \sum_{i=1}^{n} \left(\mathbf{x}_i'^T F \mathbf{x}\right)^2 \tag{4.4}$$

To make the relationship with each item of F explicit, this can be rewritten as:

$$\min_F \sum_{i=1}^{n} \left(\mathbf{U}_i^T \mathbf{f}\right)^2 \tag{4.5}$$

where $\mathbf{x}_i = (u_i, v_i)$, $\mathbf{x}'_i = (u'_i, v'_i)$ and

$$\mathbf{U}_i = [u_i u'_i, v_i u'_i, u'_i, u_i v'_i, v_i v'_i, v'_i, u_i, v_i, 1]^T, \mathbf{f} = [F_{11}, F_{12}, F_{13}, F_{21}, F_{22}, F_{23}, F_{31}, F_{32}, F_{33}]^T$$

All that remains is to impose the scale factor constraint. This is achieved by using orthogonal least-squares and solving equation 4.5, subject to the constraint that $\|\mathbf{f}\| = 1$. In this situation, no one parameter of $\mathbf{f}$ will prevail over any other and consequently there will be no bias. The resulting constrained problem can be transformed into an unconstrained one by the use of Lagrange multipliers:

$$\min_F \sum_{i=1}^{n} \left(\mathbf{U}_i^T \mathbf{f}\right)^2 + \lambda \left(1 - \|\mathbf{f}\|\right)$$

with λ as the Lagrange multiplier. Requiring the first derivative of this function with respect to $\mathbf{f}$ to be zero gives:

$$\mathbf{U}_n^T \mathbf{U}_n \mathbf{f} = \lambda \mathbf{f}$$

Identifying this to a linear system of equations, the solution $\mathbf{f}$ must be a unit eigenvector of the 9×9 moment matrix $\mathbf{U}_n^T \mathbf{U}_n$ and λ the corresponding eigenvalue. Since we wish to minimise the function, the solution will be the unit eigenvector associated with the smallest eigenvalue. This type of linear algorithm is generally referred to as the 8 point algorithm.

Imposing the Rank 2 Constraint

The advantage of the linear method is that it provides a closed form solution for F, but it does suffer from not imposing the rank 2 constraint. This can be a significant problem because a fundamental matrix that is not of rank 2 will produce epipolar lines that do not intersect at a consistent epipole. Subsequently, the rank 2 constraint is imposed after solving for F, by setting the smallest singular value of F to 0 by decomposing and recomposing F using the singular value decomposition. In effect, this replaces F by F' that minimises the frobenius norm of $\|F - F'\|$ subject to the condition that $det\,(F') = 0$.

Alternatively, it is possible to minimise the algebraic distance given in minimisation 4.5 subject to the rank constraint using an iterative scheme. Briefly, this method enforces that the fundamental matrix be of rank 2 by factoring it as $F = M\,[\mathbf{e}]_\times$ (see sections 3.3.2 and 3.3.3 on page 58 for details of this factorisation). The algorithm proceeds by obtaining an initial estimate of the epipole from the 8 point algorithm. Given this estimate $[\mathbf{e}]_\times$ can be fixed and the remaining parameters in M found in closed form so as to minimise the algebraic error subject to a scale factor constraint. This is used in an iterative gradient descent method such as LM iteration (appendix A) to minimise against the homogeneous epipole. The obvious advantage of this method is that the minimisation is in terms of only 2 values (or 3 if the scale of the epipole is allowed to vary), but it is a fairly complicated

method to implement. In general, it is advisable to use the simple SVD method mentioned first. See [HZ00] for more details of this method.

4.3.2 Minimal Method Using Seven Points

Since the fundamental matrix has 7 entries, it follows that it should be possible to calculate F using only 7 point matches, rather than 8 as required by the previous method. In this case, it is possible to impose 6 constraints on the 9 unknowns of $\mathbf{f}$ which gives an under-determined set of linear equations. SVD of the set of equations can then be used to find two basis vectors $\mathbf{f}_1$ and $\mathbf{f}_2$ which span the solution space. Since the two vectors $\mathbf{f}_1$ and $\mathbf{f}_2$ form a basis of the solution space, they can be combined linearly $\alpha\mathbf{f}_1 + \beta\mathbf{f}_2$ to give all possible solutions. An exact solution can then be found by imposing the rank 2 and scale factor constraints.

Because of the scale factor constraint, the linear combination of $\mathbf{f}_1$ and $\mathbf{f}_2$ must only be in terms of one independent parameter α. This gives:

$$\alpha\mathbf{f}_1 + (1 - \alpha)\,\mathbf{f}_2 \tag{4.6}$$

The rank 2 constraint can then be imposed by enforcing that the determinant of $\mathbf{F}$ must be 0 yielding $\det[\alpha\mathbf{f}_1 + (1 - \alpha)\,\mathbf{f}_2] = 0$. This gives a cubic polynomial in α which can then be solved using the standard formulae to give one or three real solutions for α. The solutions for α can then be substituted back into equation 4.6 to get a complete value for $\mathbf{f}$.

Although this minimal algorithm may seem to be of little immediate use, it will prove invaluable when considering robust methods for the computation of F.

4.3.3 Other Linear Methods

There are some alternative linear methods which are now established do not perform significantly better (see [Zha98]), but worthy of note is a recent method [CGVC00] which solves the minimisation problem in equation 4.5 subject to both the rank and scale factor constraints. However, the method is very complex and does not seem to result in significant improvements.

4.4 Nonlinear Estimation of the Fundamental Matrix

The linear 8 point algorithm for fundamental matrix estimation that was just presented provides a closed form means of determining the fundamental matrix. However, it suffers greatly from a number of drawbacks. In particular:

1. The 'algebraic' distance measure being minimised has no real physical meaning and so does not always produce Gaussian distributed errors. The lack of physical meaning also means the error measure is not invariant to Euclidean transformations of the images, i.e. the error function will produce different values depending on the image coordinate system.

2. The error function also suffers from a lack of normalisation, meaning that a different scale for F will result in different error values.

3. Finally, the constraint that the rank of F is 2 is not enforced during the minimisation.

In the next sections, solutions to these problems will be presented. Solving the first two will require a change in error function, and solving the third will require a special parameterisation of F to enforce all the constraints on it. Both these problems will now be addressed separately.

4.4.1 Error Function

Since the need for the error function to be linear has been dropped, it is appropriate to attempt to solve the first two problems outlined above and look for an error function that is a maximum likelihood estimator (ML) for the given noise model.

To find such a function, first consider a perfectly matched pair of error free points $\hat{\mathbf{m}} \leftrightarrow \hat{\mathbf{m}}'$. Such matches will have to satisfy the two image geometric constraint $\hat{\mathbf{m}}'^T F \hat{\mathbf{m}} = 0$ exactly. However, in practise the observed match points $\mathbf{x} \leftrightarrow \mathbf{x}'$ will be disturbed by measurement errors, in this case assumed to be Gaussian distributed. Under this noise model, the ML estimate can be found by minimising the square of the residuals, hence the following function:

$$d_e(\mathbf{x}, \hat{\mathbf{m}})^2 + d_e(\mathbf{x}', \hat{\mathbf{m}}')^2 \tag{4.7}$$

subject to the underlying geometric constraint that $\hat{\mathbf{m}}'^T F \hat{\mathbf{m}} = 0$. Note that terms for both images are required to prevent one image being unfairly weighted against the other

(the effects of which are discussed in [LDFP93]). This *variational* approach to parameter estimation was first specified in [Tri87] where a solution was found using simplex methods. Unfortunately, in this particular case, the resultant minimisation is not a good candidate for practical use since it is nonlinear, constrained, and for n matches requires calculation of $4n + 7$ unknowns, i.e. $\hat{\mathbf{m}}, \hat{\mathbf{m}}', F$.

Fortunately, the constraint that $\hat{\mathbf{m}}'^T F \hat{\mathbf{m}} = 0$ can be added to the error function quite easily by restating it in terms of the transfer constraints provided by equations 4.1 and 4.2. These allow $d_e(\mathbf{x}, \hat{\mathbf{m}})$ to be rewritten as $d_l\left(\mathbf{x}, F^T\hat{\mathbf{m}}'\right)$ where d_l is the orthogonal distance of a point to a line, giving a new function to minimise:

$$d_l\left(\mathbf{x}, F^T\hat{\mathbf{m}}'\right)^2 + d_l\left(\mathbf{x}', F\hat{\mathbf{m}}\right)^2 \tag{4.8}$$

The global minimum of this function will be an ML estimate because by definition it will always be the orthogonal projection of $\mathbf{x}$ on $F^T\hat{\mathbf{m}}'$ and $\mathbf{x}'$ on $F\hat{\mathbf{m}}$ which minimise the sum of squared distances in equation 4.7. The distance of points to their orthogonal projection on a line is naturally equivalent to the orthogonal distance of points to lines, so the two minimisations are the same.

It is actually possible to minimise equation 4.8 using an algebraic search method such as gradient descent or Levenberg-Marquardt. After an initial estimate of F has been obtained, the methods to be presented in section 5.4.3 (page 102) can be used to determine the error free matches $\hat{\mathbf{m}} \leftrightarrow \hat{\mathbf{m}}'$ exactly for the given F. The minimisation can then proceed allowing F to vary and re-calculating the error free matches $\hat{\mathbf{m}} \leftrightarrow \hat{\mathbf{m}}'$ for each updated F. However, such a method is computationally very expensive and it will be shown later that there is a better approach based on projection matrices and 3D structure.

Although optimal, there is a major problem with the previous function in that $\hat{\mathbf{m}}$ and $\hat{\mathbf{m}}'$ are unknowns. Instead, at the cost of optimality the following function can be minimised instead:

$$d_l\left(\mathbf{x}, F^T\mathbf{m}'\right)^2 + d_e\left(\mathbf{x}', F\mathbf{x}\right)^2 \tag{4.9}$$

The obvious advantage of this function is that given n points it only requires estimation of 7 parameters instead of $7 + 4n$ as for equation 4.8. However, minimising it using standard methods no longer produces an ML estimate under the Gaussian assumption since noisy points are transferred. However, it does provide a very close approximation.

An intuitive alternative to all the previous discussion is to replace the error free points $\hat{\mathbf{m}} \leftrightarrow \hat{\mathbf{m}}'$ in equation 4.7 with the projection of 3D structure:

$$d_e\left(\mathbf{x}_i, P\mathbf{X}\right) + d_e\left(\mathbf{x}_i', P'\mathbf{X}\right)$$

For projection matrices P, P' and structure $\mathbf{X}$. Whilst minimising this function produces a maximum likelihood estimate it does suffer from the drawback that P, P' and $\mathbf{X}$ are unknowns. It should be noted that this represents an improvement over the function in equation 4.8 by reducing the number of unknowns from $4n + 7$ to $3n + 11$ (assuming P is fixed).

Minimisation of this function is viable, and is achieved by using the methods of sections 5.3 and 5.4 in chapter 5 to produce cameras and structure from an initial guess of the fundamental matrix. Minimisation can then proceed using the well known bundle adjustment, a nonlinear refinement where both structure and cameras are allowed to vary at the same time. This minimisation is discussed in detail in appendix B, and it is notable that it can be implemented very efficiently. Considering all this, minimisation of re-projection error is usually the best approach.

There are also other alternative error measures. Particularly worthy of note is the recasting of the estimation problem as minimising the distance between the four dimensional point $(\mathbf{x}_i, \mathbf{x}'_i)$ and the quadratic surface defined by $\mathbf{x}'^T F \mathbf{x} = 0$. This difference can be determined exactly with great computational effort using the techniques in [LDFP93], or can be approximated to first order using the technique of [Sam82] (originally developed for conic fitting). However, it has been shown in [LDFP93, LF96b] that this technique has very similar results to the error function in equation 4.9 so shall not be considered here.

4.4.2 Relation to the Linear Criterion and Iterative Methods

It is interesting to relate the nonlinear measure in equation 4.9 to the linear criterion in equation 4.3 $\mathbf{x}'^T F \mathbf{x} = 0$. If orthogonal distance is used to measure distance between points $\mathbf{x}'_i$ and corresponding epipolar lines $F\mathbf{x}_i = \mathbf{l}_i = [l'_1, l'_2, l'_3]$, it gives:

$$d_l\left(\mathbf{x}'_i, F\mathbf{x}_i\right) = \frac{\mathbf{x}'^T_i \mathbf{l}'_i}{\sqrt{l'^2_1 + l'^2_2}} = \frac{1}{\sqrt{l'^2_1 + l'^2_2}} \mathbf{x}'^T_i F \mathbf{x}_i \tag{4.10}$$

From looking at this equation it is apparent that it is the same as equation 4.3 but weighted by $\frac{1}{\sqrt{l'^2_1 + l'^2_2}}$. It follows that if the weighting in equation 4.10 were known it could be used to replace the distance measures in equation 4.9, to make the error measure linear.

However, the weights themselves depend on the estimate of F, which immediately suggests the possibility of an iterative algorithm. In this case all weights are initially set to 1 and a solution found using the 8 point algorithm. Subsequent stages, apply weights which are calculated from the previous stage and thus can minimise equation 4.9 until there is

no improvement in the error measure. However, reviews of fundamental matrix estimation [Zha98] as well as personal experience have shown iterative linear methods to provide little or no improvement so they will not be considered further here.

4.4.3 Summary of Error Functions

Since such a large number of error functions have been presented the methods will all be briefly summarised here:

- Algebraic distance: $\mathbf{x}'^T F\mathbf{x} = 0$. This measure has the advantage of being linear, and hence can be minimised in a least-squares sense in closed form. However, it does not produce Gaussian distributed errors, is not invariant to Euclidean transforms of the image space and is sensitive to changes in the scale of F.

- Iterative Linear: This approach uses the algebraic distance to generate a guess of F, and then based on this guess determines weights for each residual function so that it produces a fully euclidean distance measure (rather than the algebraic approximation). However, iterative methods often fail to improve errors and frequently fail to converge on an exact minimum but iterate to wildly different (and sometimes worse) results.

- Euclidean distance to error free epipolar lines: $d_l\left(\mathbf{x}, F^T\hat{\mathbf{m}}'\right)^2 + d_l\left(\mathbf{x}', F\hat{\mathbf{m}}\right)^2$. Has the advantage of producing errors with a Gaussian distribution, and of being normalised. Comes with the disadvantage of being nonlinear and complex and expensive to evaluate fully.

- Euclidean distance to epipolar lines: $d_l\left(\mathbf{x}, F^T\mathbf{x}'\right)^2 + d_l\left(\mathbf{x}', F\mathbf{x}\right)^2$. Provides only an approximation of the maximum likelihood estimate. Has advantage of only containing 7 unknowns, of being simple to evaluate and of being normalised.

- Re-Projection: $d_e\left(\mathbf{x}_i, P\mathbf{X}\right) + d_e\left(\mathbf{x}'_i, P'\mathbf{X}\right)$. Only disadvantage is the nonlinearity, requiring minimisation of both projection matrices and structure. Has the advantages of being normalised, being efficient to minimise, producing a maximum likelihood estimate and having less unknowns than the equivalent maximum likelihood measure involving fundamental matrices.

4.4.4 Parameterisation of F

As well as requiring a new error function, the fundamental matrix also needs to be parameterised so as to take into account the scale factor and rank constraints. Three methods for performing this will be presented here.

Using the Determinant Constraint

An intuitive way to enforce that the rank of F is 2 is to use the constraint that $\det(F) = 0$. Expanding out $\det(F) = 0$ provides a cubic equation in the coefficients of F that can easily be used to determine one item of F given the other eight. The scale factor constraint can then be enforced by fixing the largest remaining parameter of F to 1. This method will be referred to as $N1$.

Considering the Fundamental Matrix as a Singular Matrix

In order for a fundamental matrix F to have a rank of 2 and hence be singular, one row or column must be a linear combination of the other two e.g. given columns $\mathbf{c}_1, \mathbf{c}_2, \mathbf{c}_3$ it must be the case that

$$(\exists j0, j1, j2 \in [1,3])\ (\exists \lambda_1, \lambda_2 \in \mathcal{R})\,,\ \mathbf{c}_{j0} + \lambda_1 \mathbf{c}_{j1} + \lambda_2 \mathbf{c}_{j2} = 0 \tag{4.11}$$

$$(\nexists \lambda \in \mathcal{R})\,,\ \mathbf{c}_{j1} + \lambda \mathbf{c}_{j2} = 0 \tag{4.12}$$

Condition 4.12 is a non-existence condition that enforces the rank of F is not less than 2. Since it cannot easily be expressed by a parameterisation it will not be used.

Given the problem is symmetrical it makes sense to enforce this for both columns and rows, resulting in a description for F by four variables, and two pairs of scalings. If the scalings are collected into a vector associated with the relevant columns or rows of the fundamental matrix and a 1 is added they can be seen to be equivalent to the left and right null-spaces (kernels) of the fundamental matrix - the epipoles. For example, given epipoles $(x, y, 1)$, $(x', y', 1)$ and the four unknown parameters a, b, c, d the matrix can be written as (selecting the third column and row to be linear combinations).

$$F = \begin{bmatrix} a & b & -ax - by \\ c & d & -cx - dy \\ -ax' - cy' & -bx' - dy' & (ax + by)x' + (cx + dy)y' \end{bmatrix} \tag{4.13}$$

Finally, to impose the scale factor constraint, the largest parameter of the four remaining parameters a, b, c, d is normalised to 1. Depending upon which row or column is expressed as a linear combination of the other two, it can be seen that excluding the scale factor there are 9 possible parameterisation of the fundamental matrix.

In [Zha98, CZZF96] the best parameterisation was then selected by maximising the rank of the 9x8 Jacobian of the appropriately re-parameterised fundamental matrix. This method will be referred to as $N2$, and full details of it can be found in [CZZF96].

Parameterisation in Terms of Left and Right Kernels

There is also a slightly different formulation of the previous parameterisation that is much clearer and more general. This formulation has also found use for parameterisation of the trifocal tensor [PF98] which can be decomposed into a set of 3 matrices similar to fundamental matrices and also of rank 2.

The formulation considers that both left and right null-spaces are attached to specific properties of the system of cameras, epipoles in the case of fundamental matrices. It follows that the space $\mathcal{M}(\mathbf{L}, \mathbf{R})$ of all matrices with a given left kernel $\mathbf{L} = [l_1, l_2, l_3] \neq \mathbf{0}$ and right kernel $\mathbf{R} = [r_1, r_2, r_3] \neq \mathbf{0}$ is of some importance. If the left and right kernels are considered to be homogeneous quantities then it follows that the rank of any matrix in $\mathcal{M}(\mathbf{L}, \mathbf{R})$ must be at most 2.

The space defined by $\mathcal{M}(\mathbf{L}, \mathbf{R})$ is in fact a linear space of dimension 4. Consequently, it is possible to find a basis and so describe any matrix of $\mathcal{M}(\mathbf{L}, \mathbf{R})$ in terms of 4 coordinates marking a position in the linear space. In fact, these 4 coordinates correspond to the 4 coefficients of the original homography that relates epipolar lines in the two images (a, b, c, d) in the previous method.

Unfortunately there is no means of specifying a basis for $\mathcal{M}(\mathbf{L}, \mathbf{R})$ that will be valid for any choice of $\mathbf{L}$ and $\mathbf{R}$ since certain entries in $\mathbf{L}$ and $\mathbf{R}$ may be 0. However, since $\mathbf{L}$ and $\mathbf{R}$ are epipoles it is known that $\mathbf{L} \neq \mathbf{0}$ and $\mathbf{R} \neq \mathbf{0}$. Because of this, if it is assumed that the highest components in magnitude of both $\mathbf{L}$ and $\mathbf{R}$ are in first position it is guaranteed that $l_1 \neq 0$ and $r_1 \neq 0$. Consequently the following four matrices of rank 1 always constitute a basis of $\mathcal{M}(\mathbf{L}, \mathbf{R})$

$$M_1 = \begin{bmatrix} r_3 l_3 & 0 & -r_1 l_3 \\ 0 & 0 & 0 \\ -r_3 l_1 & 0 & r_1 l_1 \end{bmatrix}, M_2 = \begin{bmatrix} -r_2 l_3 & r_1 l_3 & 0 \\ 0 & 0 & 0 \\ r_2 l_1 & -r_1 l_1 & 0 \end{bmatrix}$$

$$M_3 = \begin{bmatrix} -r_3l_2 & 0 & r_1l_2 \\ r_3l_1 & 0 & -r_1l_1 \\ 0 & 0 & 0 \end{bmatrix}, M_4 = \begin{bmatrix} r_2l_1 & -r_1l_2 & 0 \\ -r_2l_1 & r_1l_1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

This means a fundamental matrix with left and right kernels $\mathbf{L}$ and $\mathbf{R}$ can be described in terms of the basis just given and four coordinates a_1, a_2, a_3, a_4 as

$$F = a_1M_1 + a_2M_2 + a_3M_3 + a_4M_4 \tag{4.14}$$

It can also be seen that M is of rank 1 iff $a_2a_3 - a1a_4 = 0$. In order to ensure that the largest value of $\mathbf{L}$ and $\mathbf{R}$ are in the first position, the rows and columns of the fundamental matrix must be circularly permuted appropriately prior to decomposition, and then back again after re-composition. Furthermore, for increased numerical stability the largest value of the left and right kernels should also be normalised to 1. This method will be referred to as $N3$.

In fact, if equation 4.14 is multiplied through, a very similar parameterisation to that used for method $N2$ from the previous section will be derived. The only difference is that the parameterisation $N2$ assumes the final coordinate of the epipole is 1 i.e. $\mathbf{e} = (ex, ey, 1)$. As mentioned in section 4.2.4 this can be numerically unstable and so the method $N2$ can be modified to deal with the problem. To allow for arbitrary scaling of the epipoles, the condition 4.11 is modified so that an extra scaling factor is included

$$(\exists j0, j1, j2 \in [1,3])\ (\exists \lambda_1, \lambda_2, (\lambda_3 \neq 0) \in \mathcal{R})\,,\ \lambda_3\mathbf{c}_{j0} + \lambda_3\lambda_1\mathbf{c}_{j1} + \lambda_3\lambda_2\mathbf{c}_{j2} = 0$$

Using this new condition, and normalised epipoles $\mathbf{e} = (e_1, e_2, e_3)$ and $\mathbf{e}' = (e'_1, e'_2, e'_3)$ such that the largest item in the epipole is 1, the fundamental matrix can now be written as (in this case, the first row and column are selected to be linear combinations of the other two):

$$F = \begin{bmatrix} (e_3a - e_2b)\, e'_3 + e'_2\,(e_2d - e_3c) & (e'_3b - e'_2d)\, e_1 & (e'_2c - e'_3a)\, e_1 \\ (e_3c - e_2d)\, e'_1 & e_1e'_1d & -e_1e'_1c \\ (e_2b - e_3a)\, e'_1 & -e_1e'_1b & e_1e'_1a \end{bmatrix} \tag{4.15}$$

If it is compared with the parameterisation $N3$ just given, this new form is now exactly equivalent. Equivalence can also be established the other way around by enforcing that the largest item in the left and right kernels is permuted to be in e_3, e'_3 and is set to 1, then use the parameterisation 4.13 directly.

4.4.5 Summary

Overall this section has given a large number of parameterisations and error functions for fundamental matrix estimation. Whilst this has reviewed most state of the art nonlinear techniques it hasn't paid particular attention to which is the most effective.

Recall that to produce the ideal nonlinear method two things are needed, both an error function and a parameterisation. Ideally the error function should be a maximum likelihood estimator under the assumption that projection has resulted in Gaussian distributed errors being added to the image points. This leaves only two criteria:

- Euclidean distance to error free epipolar lines: $d_l\left(\mathbf{x}, F^T\hat{\mathbf{m}}'\right)^2 + d_l\left(\mathbf{x}', F\hat{\mathbf{m}}\right)^2$. Has the advantage of producing errors with a Gaussian distribution and of being normalised. Comes with the disadvantage of being nonlinear, but can still be minimised efficiently using the Trivedi-Simplex algorithm [Tri87] (see [LPT00] for information on performance).

- Re-Projection: $d_e\left(\mathbf{x}_i, P\mathbf{X}\right) + d_e\left(\mathbf{x}_i', P'\mathbf{X}\right)$. Main disadvantage is the nonlinearity, requiring minimisation of both projection matrices and structure. However, is normalised, efficient to minimise using existing Bundle Adjustment algorithms, and has less unknowns than the equivalent maximum likelihood measure above that involves fundamental matrices. Note that to avoid fixing the coordinate basis of projective 3 space arbitrarily the cameras should be kept in canonical form (see section 3.2.8 on page 52).

All parameterisations enforce all the possible constraints, and so the best must be selected for numerical stability and generality. Method $N3$ from section 4.4.4 is definitely the ideal. This method is equivalent to method $N2$ but with more control over the normalisation of the epipole, which helps avoid problems with infinite epipoles, as well as increasing numerical stability. Extensive testing elsewhere [LF96b] has shown method $N1$ to be inferior numerically.

To conclude, the ideal nonlinear minimisation uses the re-projection error. A small improvement can sometimes be made by using parameterisation $N3$ and the Euclidean distance to error free epipolar lines, which will help remove the 4 degrees of freedom that cannot be fixed by putting cameras into canonical form, and can use fewer unknowns if $\hat{\mathbf{m}}$ and $\hat{\mathbf{m}}'$ from equation 4.8 are recalculated at each iteration. However, it is notable that overparameterising the model rarely makes a significant difference to the result.

4.5 Robust Estimation of the Fundamental Matrix

So far, it has been assumed that all the measurement and matching errors of points conformed to a Gaussian distribution around the ideal value. However, if some points are mismatched, this assumption will often be wildly incorrect for those mismatched points. Even with the presence of only a few of these so called outlying points, the least-squares methods being used in the previous sections are normally rendered useless.

Standard least-squares methods attempt to minimise $\sum_i r_i^2$, where the residual r_i can be defined as the difference between the ith observation and the ith fitted value. Because the function is squared, large residuals associated with outliers will have a dominating effect on any estimated parameters. For the estimation of the fundamental matrix, the large quantity of outliers that are often present has led to the use of random sampling methods for outlier removal, in particular LMedS [RL87], RANSAC [FB81] and MLESAC [TZ00]. The use of random sampling methods in fundamental matrix estimation is now relatively standard [BTZ96, FZ98b, Zha97, TZ98]. For a comprehensive review of robust methods applied to fundamental matrix estimation see [Tor95].

The basis of any random sampling method is to pick random sub-samples of the data set and estimate the model parameters using each sub-sample of data. The best of the estimates is then determined according to the whole data set and used to eliminate outliers, the idea of this being that if enough sub-samples are taken there is a good chance at least one sub-sample will be outlier-free. Consequently, when using a random sampling method it is very important that the minimum number of points per sub-sample are used so as to reduce the probability of an outlier being included.

All of RANSAC, LMedS and MLESAC work on these principles, with the main difference being defined by the number of samples taken and the function minimised. LMeds attempts to minimise the median error, RANSAC attempts to maximise the number of inliers and MLESAC attempts to minimise a robust function of the residuals involving both inliers and outliers (e.g. a Huber function - see section C.2, page 256). The number of sub-samples is often defined so as to ensure a particular chance of successfully selecting a sub-sample containing no outliers. Alternatively because the process is a minimisation, an arbitrary and fairly large number of sub-samples is taken to ensure a good minimum and not just no outliers. Although, all three of these algorithms have been implemented for the purposes of this work, MLESAC proved to produce the best solutions. It has been shown elsewhere to outperform other random sampling methods in the case of fundamental matrix estimation

[TZ00], and so was used in all situations. To give a general feel for random sampling, the LMedS and MLESAC algorithms have been described in detail in appendix C.

Outlier Removal

The random sampling techniques work much better than least-squares when there are outliers, but are very inefficient in the presence of Gaussian noise. To remedy this, the fundamental matrix MLESAC produced is used as a ground truth to determine which matches are outliers so that they can be removed. Once the outliers have been removed, it is then reasonable to assume a Gaussian error for the remaining matches.

Outliers are determined by checking to see which matches are outside a certain confidence region of the probability distribution for errors in the data set. If the residual function r_i is the orthogonal distance of points to corresponding epipolar lines, then it would be expected for residuals to conform to a chi-squared distribution with 1 degree of freedom. Consequently, for each point if $r_i^2 \leq 3.84\sigma^2$ the point should be discarded as being an outlier. The constant 3.84 is selected because it represents a confidence limit of 95% for a chi-squared distribution with one degree of freedom, i.e. an inlier will be incorrectly rejected only 5% of the time.

However, this test is not yet usable because it requires the standard deviation σ. Unfortunately, this cannot be obtained since the data set contains outliers and so does not have a Gaussian probability distribution. Instead, a robust approximation to the standard deviation must be used, such as (see [RL87]):

$$\sigma = 1.4826\,[1 + 5/\,(n - p)]\,\sqrt{M_j}$$

where M_j is the median of squared residuals for the F estimated by random sampling, n is the number of matches in the data set and p is the number of matches in a sub-sample. Finally, outliers can be removed and, if desired, the fundamental matrix recalculated using least-squares methods.

4.6 Summary of Methods for Fundamental Matrix Estimation

A large number of different algorithms for fundamental matrix estimation have just been proposed, but they are not equally effective. From experience and results in other works

(particularly [LF96b]), the following approach to fundamental matrix estimation is taken by this work:

- *Robust Estimation:* MLESAC is used to robustly determine the fundamental matrix. Outliers are identified and removed. Note that this uses a nonlinear minimisation to further refine the robust Huber function also being minimised by random sampling.
- *Linear Estimation:* After point matching, the linear 8 point algorithm presented in section 4.3.1 is used. Points are also normalised, using the method of section 4.2.4 prior to this calculation, and the rank 2 constraint is imposed after the minimisation.
- *Nonlinear Estimation:* This is then refined by using a nonlinear minimisation of equation 4.8. See section 4.4 for details.
- *Bundle Adjustment:* Cameras and structure are instantiated using the techniques of sections 5.3.2 and 5.4 and a bundle adjustment minimising re-projection error is run. See section 6.3.1 (page 110) for details.

4.7 Estimating the Trifocal Tensor

The previous section has reviewed methods for calculating the fundamental matrix for a pair of views. Since these techniques have proved to be very robust and practically useful, similar techniques and theory for the calculation of the trifocal tensor across a triplet of views have also been developed. In this section, a very brief overview of these methods for robust computation of the trifocal tensor will be given. Less detail will be given in these descriptions since trifocal tensor computation will only be used for comparison with new algorithms in later chapters. Trifocal tensor estimation will not form part of the complete reconstruction system to be presented in chapter 10.

As discussed in section 3.4.3 (page 63) of the previous chapter, the trifocal tensor provides a linear relationship between the projections of points in three images (often referred to as the trilinear equations):

$$m^k \left(m'^i m''^m T_k^{jl} - m'^j m''^m T_k^{il} - m'^i m''^l T_k^{jm} + m'^j m''^l T_k^{im} \right) = 0^{ijlm} \tag{4.16}$$

One big advantage of the three image case is that it is now possible to match lines and so a similar constraint, linking lines in two images to the exact line in the third image, can be

found. This relationship is:

$$\mathbf{l}_i \simeq \mathbf{l}'_j \mathbf{l}''_k T_i^{jk} \tag{4.17}$$

It should also be noted that there is a close relationship between the tensor of three views and a triplet of projection matrices in canonical form ($P_1 \simeq [I|\mathbf{0}], P_2 \simeq [A|\mathbf{a}], P_3 \simeq [B|\mathbf{b}]$) due to Hartley [Har94c]:

$$T_i^{jk} \simeq A_i^j \mathbf{b}^k - \mathbf{a}^j B_i^k \tag{4.18}$$

and so the trifocal tensor can be created from the normalised projection matrices with ease.

The trifocal tensor marks a significant strengthening of the geometric constraints over the fundamental matrix. Whereas in the two view case, a point in one image could be constrained to lie only on a line in another image, for the three image case, given two projections of a point, an exact projection in the third image can be predicted. The implications of this are significant since, in the two view case, a matching point may lie anywhere on a line in the other image, meaning that it can be outlying and still fit the geometry. Such points will become outlying when the three image constraints are applied.

4.7.1 Linear Methods

Using the Transfer Relations

After outliers have been removed using robust methods, it makes sense to attempt a reconstruction of the trifocal tensor using all available matches. The linear constraint provided by the tensor provides a good starting point for a linear algorithm:

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

The equation zeros out on the left for $i = j$ or $l = m$, and swapping i and j or l and m simply changes the sign of the equation. The resulting variation means that there are twelve equations, of which only four are independent. For example, setting $j = m = 3$, letting i and l range freely and setting $x^3 = x'^3 = x''^3 = 1$ gives a simple set of 4 independent equations:

$$m^k \left(m'^i m''^l T_k^{33} - m''^l T_k^{i3} - m'^i T_k^{3l} + T_k^{il} \right) = 0^{il}$$

for varying $l, m = 1, 2$. Stacking these equations for each triplet match $\mathbf{x} \leftrightarrow \mathbf{x}' \leftrightarrow \mathbf{x}''$ means that, given at least 9 points, this equation can be solved in a least-squares sense. This is achieved by forming a moment matrix as in the 8 point algorithm for the fundamental matrix and solving using eigen analysis. See [Har95b, Har97, HZ00] for more details.

Note that if lines are also to be considered, then it is also possible to include the constraints provided by lines in equation 4.17. Eliminating the scale factor results in an additional 2 constraints per line per image.

After computation with the linear algorithm, it is usually a good idea to impose the internal constraints of the tensor. Unlike the case of the fundamental matrix where there was only a simple rank constraint, performing this for the tensor is much more involved (see [HZ00] for details). For implementations presented here, the parameterisation of [PF98] was used to enforce all the constraints after using any of the linear algorithms. However, it is likely this is not the ideal method (mainly due to it's complexity).

From Projection Matrices

A good alternative approach to calculating the trifocal tensor in a linear manner is first to determine three projection matrices representing the three images and then using equation 4.18 create the trifocal tensor from them.

Obtaining the first two camera matrices P_1 and P_2 is very simple, and can be achieved by using the fundamental matrix estimation techniques of section 4.3 above. 3D structure and cameras for the two views can then be estimated using the techniques to be presented in sections 5.3 (page 98) and 5.4 (page 101).

The third camera matrix can then be estimated using a technique known as resectioning. Almost invariably, some of the matches between images two and three will share common points in image two with matches between images one and two. These matches allow a relationship to be established between 3D structure and corresponding projections in the third image. Resectioning can then be used to determine the third camera - see section 5.3.1 (page 98) for a detailed description of resectioning.

After all three projection matrices have been determined, the trifocal tensor is easily recovered using the relationship in equation 4.18. The main disadvantages of this method are that lines cannot be included very easily and that the result will often be biased toward the first two images for which a fundamental matrix was calculated.

4.7.2 Nonlinear Methods

As with the linear methods for estimating the fundamental matrix, the linear methods for the trifocal tensor use an inferior error measure, and overlook certain internal constraints on the tensor. For the case of the trifocal tensor this is quite a significant problem, since it depends

on 18 parameters, yet is estimated by the linear algorithm using 26. These constraints are fairly involved and will not be discussed here, so the interested reader is referred to [PF98, FP97, TZ97] for details on how to parameterise the tensor and to [TZ97] for details on an error measure that gives a first order approximation to an ML error measure.

In experimenting with these nonlinear methods, the author found that the method in [PF98], whilst it improved results, was not as effective as calculating all the structure and cameras, then running a bundle adjustment. Although this may be a reflection on the quality of the implementation, there is still some doubt as to the usefulness of a direct nonlinear refinement of the trifocal tensor. This seems to be largely because the error measure for tensor transfer does not result in a maximum likelihood estimate, whereas the measure used by bundle adjustment does.

4.7.3 Robust Methods

The trifocal tensor can also be calculated using the same robust methods as the fundamental matrix (section 4.5). These robust algorithms require a method which takes a minimal number of observations to produce an estimation of the model parameters. In this case, the trifocal tensor has 18 parameters, and so six points across 3 images give $3*6 = 18$ constraints and so represents the minimal amount of data. The so called six point algorithm for minimal reconstruction of the trifocal tensor is given in appendix D.

4.8 Summary

This chapter has presented a number of methods for calculating both the fundamental matrix describing the geometry of two views and the trifocal tensor describing the geometry of three views.

In order to calculate a fundamental matrix, it is recommended to use the eight point algorithm described in section 4.3.1, followed by imposing the rank 2 constraint using either of the methods in section 4.3.1. In general, the method based on SVD is to be recommended for purposes of simplicity. If desired, the final stage in fundamental matrix estimation is a nonlinear refinement in order to find a maximum likelihood estimate of the fundamental matrix. The best algorithm for this is the bundle adjustment approach using structure and cameras.

For estimation of the trifocal tensor, if only points are to be considered, it is recommended

to first obtain a linear estimate using the method described in section 4.7.1. Otherwise, if lines are to be included, the method based on transfer relations should be used instead. Finally, a refinement of the triplet geometry should be achieved using a bundle adjustment approach based on minimising the re-projection of structure using camera matrices. Since the tensor is not necessary for any of these stages, it can finally be constructed using the simple relation in equation 4.18.

Regardless of which quantity is being estimated, if there is any possibility of outliers then the robust approaches presented in sections 4.5 and 4.7.3 should be used.

Chapter 5

Projective Reconstruction of 3D Cameras and Structure

5.1 Introduction

The multilinear constraints discussed so far provide a convenient, minimal, and above all else, easily calculated representation for the geometry of two, three or four images. However, for some applications, or if more images are to be considered, multilinear forms can become inadequate or even difficult and cumbersome. Consequently, it is often beneficial to convert the representation from multilinear forms to the more intuitive form of projection matrices and three dimensional structure.

There are naturally advantages to using projection matrices and structure, instead of multilinear forms, to describe geometry. The biggest difference is that multilinear forms can only describe the cameras and not the structure in the scene. Structure can clearly be useful for many applications, and also in maximum likelihood estimation. In addition, the use of projection matrices instead of multilinear forms enables the modelling of much longer image sequences in a much more convenient manner.

This chapter will address the problems of determining projection matrices and structure from the multilinear forms, or from an existing reconstruction. Essentially, these methods consider the problem of obtaining reconstructions for only very small sequences. Chapters 6 and 7 will then present a number of effective methods for determining cameras for much longer sequences of images. These methods for long sequences will often rely heavily on the methods presented in this and the previous chapter.

5.2 Reconstruction Ambiguity

It is important to remember that, when modelling a scene with camera matrices and structure, the reconstruction will be subject to an arbitrary transformation of the world space; a gauge freedom (as discussed in section 3.2.7 on page 50). This means that an arbitrary projectivity T can be applied to both cameras P and structure X without altering the reconstruction:

$$\hat{P} \simeq PT^{-1}$$
$$\hat{\mathbf{X}} \simeq T\mathbf{X}$$

Reprojecting the altered reconstruction cancels T out:

$$\mathbf{x} \simeq PT^{-1}T\mathbf{X}$$

This gauge freedom means that two projective reconstructions must have the same projective basis in order for cameras and structure in one reconstruction to relate to the other reconstruction. This presents a problem for the cameras and structure representation since in the projective case 15 extra degrees of freedom exist.

A similar problem does not exist when considering the geometry as described by the multilinear forms. Since the multilinear forms work with image quantities only, the reconstruction ambiguity has been cancelled out by the projection process as just shown.

5.3 Reconstruction of Cameras

This section will address the problem of determining projection matrices given a fundamental matrix, trifocal tensor, or if there is some way of relating known 3D structure to 2D projections.

5.3.1 Resectioning: Using Projections of Known 3D Structure

It is not uncommon, when reconstructing, for some structure to be known, either from actual measurements of the world or from an existing reconstruction. In this case, it is possible to add camera matrices for new images, provided a relation can be established between the known 3D structure $\mathbf{X}_i$ and projections of that structure in the new images $\mathbf{x}_i$, e.g. by image

based matching. Given this relationship, the projection process can be used to provide linear constraints on the unknown camera matrix P:

$$P\mathbf{X}_i \simeq \mathbf{x}_i$$

Eliminating the unknown scale factors yields two linear constraints on P per point as follows:

$$\begin{bmatrix} \mathbf{0}_4^T & -w_i\mathbf{X}_i^T & v_i\mathbf{X}_i^T \\ w_i\mathbf{X}_i^T & \mathbf{0}_4^T & -u_i\mathbf{X}_i^T \end{bmatrix} \begin{pmatrix} \mathbf{p}_1 \\ \mathbf{p}_2 \\ \mathbf{p}_3 \end{pmatrix} = A \begin{pmatrix} \mathbf{p}_1 \\ \mathbf{p}_2 \\ \mathbf{p}_3 \end{pmatrix} = 0 \tag{5.1}$$

where $\mathbf{p}_n$ indicates the column vector associated with the nth row in P. In this case, at least 6 points are required to determine the twelve parameters of P. If more than six points are available, the criterion 5.1 can be minimised in a least-squares sense, subject to the scale factor constraint on P. Imposing the scaling using Lagrange multipliers (as in section 4.3.1, page 79) means a least-squares solution can be found as the eigenvector associated with the smallest eigenvalue of the moment matrix A^TA.

It is fairly obvious that the linear criterion in equation 5.1, used to estimate the projection matrix, is not going to produce a maximum likelihood estimate for a sensible noise model. The algebraic distance it minimises will not be invariant to transformations of the image plane and is not normalised. Consequently, after obtaining the linear estimate, a maximum likelihood estimate for the camera matrix (assuming no error on the structure $\mathbf{X}_i$) is found by using an error criterion based on re-projection error and a Gaussian noise model:

$$\min_P \sum_i d_e^2 \left(P\mathbf{X}_i, \mathbf{x}_i\right)$$

This minimisation can be carried out using an iterative method such as gradient descent or Levenberg-Marquardt (see appendix A). Note that this criterion is not a maximum likelihood estimator, if all the $\mathbf{X}_i$ are determined subject to errors. A true maximum likelihood estimator would need to minimise both structure and camera matrices simultaneously (a bundle adjustment) and will be presented in section 6.3.1 (page 110).

Overall, this resectioning process finds many uses. It has the advantage that it allows a camera matrix to be added to a current reconstruction and hence to be found in the same projective basis as the reconstruction. On the other hand, the quantity it minimises is questionable and produces worse results than many other techniques, certainly those based on multilinear forms.

5.3.2 From the Fundamental Matrix

If the fundamental matrix is available, then there is a very good method for camera determination without the need for resectioning. This method was already given in section 3.6.3, page 67 where the camera matrices were directly related to the fundamental matrix as:

$$\begin{aligned} P_1 &= \left[I_{3\times 3}|\mathbf{0}_3^T\right] \\ P_2 &= \left[[\mathbf{e}_{12}]_\times F_{12} - \mathbf{e}_{12}\pi^T|\alpha\mathbf{e}_{12}\right] \end{aligned} \tag{5.2}$$

for projection matrices P, fundamental matrix F, epipoles $\mathbf{e}$, a scaling α and some 3 vector π. Since this relates the projection matrices to fundamental matrices directly, there is no loss of accuracy in converting between representations. It also has the advantage that existing structure is not required to produce a reconstruction. However, it does have the disadvantage that using this relation requires fixing the projective basis arbitrarily (by the form of P_1 and the 4 parameters π and α), and so this does come with the drawback that reconstructions are limited to only two images.

5.3.3 From the Trifocal Tensor

Obtaining camera matrices from the trifocal tensor is a little more complex than for the fundamental matrix. In brief, epipoles and then fundamental matrices are determined from the tensor. The first two cameras can then be reconstructed from the fundamental matrix between the first two images F_{12}, as just described in section 5.3.2. It is important to remember that these projection matrices will be determined subject to fifteen arbitrary degrees of freedom corresponding to a change of projective basis.

Since these fifteen degrees of freedom have been fixed, the relation in section 5.3.2 cannot be used with F_{13} to create a third camera. Fortunately, 11 parameters can be dealt with easily since F_{12} and F_{13} share the same first camera. By convention, fixing this camera as $[\mathbf{I}_{3\times 3}|\mathbf{0}_3]$ immediately eliminates 11 of the unknown parameters.

However, four degrees of freedom still remain, as represented by π and α in equation 5.2. In order to calculate these four degrees of freedom, the third camera is parameterised using equation 5.2 in terms of the four unknowns π, α and then substituted along with the first two cameras into the equation relating cameras and the tensor (equation 4.18, page 93). Rearranging the equation allows the four unknown parameters to be determined, and a third camera in the same projective basis to be produced. A complete and much more detailed description of this process can be found in [HZ00].

5.4 Reconstruction of 3D Structure

Once camera projection matrices have been obtained, it is possible to estimate 3D structure, provided projections of that structure are available in two or more images. There are numerous approaches to this 'triangulation' problem, and a comprehensive review of these methods applied to projective reconstruction can be found in [HS94, HS97]. For purposes of brevity, only the most effective methods for projective reconstruction will be explained in detail here.

Before continuing, it is worth mentioning the well known midpoint method often encountered in texts concerning reconstruction. This approach back-projects the image points to lines in space. Since the observed points are subject to measurement errors, these lines will not intersect exactly and so the space point is reconstructed as the midpoint of the common perpendicular to the two rays. Because the object space used for the case in hand is projective, concepts such as perpendiculars and distance are meaningless, resulting in a method that is not invariant to projective transformations of the object space, and hence not desirable. It should, however, be noted that the midpoint method is widely regarded as the method of choice when working with a Euclidean object space [HS94].

5.4.1 Linear Method

In order to produce a method that is invariant to projective transformations of the object space, it is best to use measurements in the image space only. Considering the projections of the 3D structure $\mathbf{X} = [x, y, z, t]^T$ into a set of any n images, and labelling the projection in image i as $\mathbf{x}_i = [u_i, v_i, 1]^T$, $\forall i \in (1, \ldots, n)$, then re-projection using the relevant camera P_i gives a simple set of 3 linear constraints per image:

$$s_i \, [u_i, v_i, 1]^T = P_i \, [x, y, z, t]^T$$

If the scale factors $s_i = p_{i3}^T \mathbf{X}$ are eliminated and the constraints for all n points are stacked into a matrix A, $2n$ independent constraints on $\mathbf{X}$ are obtained as follows:

$$\begin{bmatrix} \mathbf{p}_{i1}^T - u_1 \mathbf{p}_{i3}^T \\ \mathbf{p}_{i2}^T - v_1 \mathbf{p}_{i3}^T \\ \vdots \\ \mathbf{p}_{i1}^T - u_n \mathbf{p}_{i3}^T \\ \mathbf{p}_{i2}^T - v_n \mathbf{p}_{i3}^T \end{bmatrix} \mathbf{X} = A\mathbf{X} = 0$$

where $\mathbf{p}_{in}$ indicates the column vector associated with the nth row in P_i, provided that $n \geq 2$ this criterion can be used in a least-squares minimisation. Care still needs to be taken though, since $\mathbf{X}$ is itself only defined subject to a scale factor. This scale factor constraint $\|\mathbf{X}\| = 1$ can be imposed by again using Lagrange multipliers (see section 4.3.1, page 79) and, subsequently, a least-squares solution can be found as the eigenvector associated with the smallest eigenvalue of the moment matrix $A^T A$.

5.4.2 Nonlinear Method

The best approach to estimating structure is to develop a maximum likelihood estimator. In this case, the assumption that point localisation is perturbed by noise with a Gaussian distribution is made and so the subsequent maximum likelihood estimator would need to minimise the squared re-projection error with respect to both structure and cameras. Since only structure is being estimated here, the assumption that P_i is known without error can be made and an approximation to the true maximum likelihood estimate found by minimising:

$$\min_{\mathbf{X}} \sum_i d_E^2 \left(P_i \mathbf{X}, \mathbf{x}_i\right) \tag{5.3}$$

for structure $\mathbf{X}$ projected through cameras P_i to give image points $\mathbf{x}_i$. Since this is nonlinear, in general it can only be used as a refining stage applied, after an approximation of structure has been obtained, for example using the linear algorithm. Minimisation could, for example, proceed using the Levenberg-Marquardt algorithm (see appendix A).

5.4.3 Hartley-Sturm Match Correction for Two Images

In [HS94, HS97], a closed form solution for the maximum likelihood estimate of structure is presented, in the context of points matched for two images only. The method is based on using a simple maximum likelihood criterion for fundamental matrix estimation which minimises the squared distance between the error free images points to be estimated $\hat{\mathbf{m}} \leftrightarrow \hat{\mathbf{m}}'$ and the observed image points $\mathbf{x} \leftrightarrow \mathbf{x}'$:

$$d_e \left(\mathbf{x}, \hat{\mathbf{m}}\right)^2 + d_e \left(\mathbf{x}', \hat{\mathbf{m}}'\right)^2$$

subject to the constraint that $\hat{\mathbf{m}}'^T F \hat{\mathbf{m}} = 0$. The reader is referred to chapter 4 for full details and development of this criterion (particularly section 4.4 and equation 4.7). After minimising this least-squares criterion, the error free points $\hat{\mathbf{m}}$ and $\hat{\mathbf{m}}'$ will be known and

so 3D structure minimising equation 5.3 can be reconstructed perfectly using any linear technique (since there is no error to minimise).

As mentioned in section 4.4.1 the difficulty with this minimisation is in applying the constraint that $\hat{\mathbf{m}}'^T F \hat{\mathbf{m}} = 0$. Fortunately, there is a fairly simple solution because any pair of points $\hat{\mathbf{m}} \leftrightarrow \hat{\mathbf{m}}'$ that perfectly satisfy the epipolar constraint will themselves lie on a pair of corresponding epipolar lines $\hat{\mathbf{l}} \leftrightarrow \hat{\mathbf{l}}'$. Of all points on these epipolar lines, it will be the orthogonal projection of $\mathbf{x}$ on $\hat{\mathbf{l}}$ and $\mathbf{x}'$ on $\hat{\mathbf{l}}'$ which minimise the sum of squared distances in equation 4.7.

Considering this, $d_l\left(\mathbf{x}, \hat{\mathbf{l}}\right)^2$ can be substituted for $d_e\left(\mathbf{x}, \hat{\mathbf{m}}\right)^2$ to give a new function to minimise:

$$d_l\left(\mathbf{x}, \hat{\mathbf{l}}\right)^2 + d_l\left(\mathbf{x}', \hat{\mathbf{l}}'\right)^2 \tag{5.4}$$

subject to the constraint that $\hat{\mathbf{l}}$ and $\hat{\mathbf{l}}'$ are corresponding epipolar lines. Once $\hat{\mathbf{l}}$ and $\hat{\mathbf{l}}'$ have been found by minimising this equation, then $\hat{\mathbf{m}}$ and $\hat{\mathbf{m}}'$ can be found as the orthogonal projections of $\mathbf{x}$ on $\hat{\mathbf{l}}$ and of $\mathbf{x}'$ on $\hat{\mathbf{l}}'$ respectively.

Performing the Minimisation

The approach for minimising equation 5.4 relies on a parameterisation of the epipolar lines in the images in terms of only one parameter t, thus recasting the minimisation as a polynomial in terms of that parameter. This is achieved by applying a rigid transformation to the points in both images in order to take $\mathbf{x}$ and $\mathbf{x}'$ to the origin $(0, 0, 1)$. The epipoles are then rotated so that they are placed on the x axis at points $(1, 0, f)$ and $(1, 0, f')$. The translation to take $\mathbf{x}$ to the centre is very straightforward. Given $\mathbf{x} = (u, v, 1)$ it can be represented by:

$$T = \begin{bmatrix} 1 & 0 & 0 & -u \\ 0 & 1 & 0 & -v \\ 0 & 0 & 1 & 1 \end{bmatrix}$$

Next, to place the epipole $\mathbf{e}$ on the x axis, a rotation is performed around the origin by an angle α. To prevent problems with infinite epipoles, the rotation of the epipolar line containing $\mathbf{x}$ and $\mathbf{e}$ is considered, i.e. $\mathbf{l} = (l_a, l_b, l_c) = \mathbf{x} \times \mathbf{e}$. To rotate this line to the form y=0 (and hence place the epipole on the x axis), an anticlockwise rotation of the following form is used:

y=0 (and hence place the epipole on the x axis), an anticlockwise rotation of the following form is used:

$$R = \begin{bmatrix} cos(\alpha) & -sin(\alpha) & 0 \\ sin(\alpha) & cos(\alpha) & 0 \end{bmatrix}$$

The angle α can be calculated using the constraint that the line $\mathbf{l}$ must have the form $(0, 1, 0)$:

$$RT^* * \mathbf{l} = \lambda * (0, 1, 0)$$

where λ is an arbitrary non zero scale factor and T^* indicates the matrix of cofactors of T (Note if a matrix M transforms points, then M^* is the equivalent transformation for lines). Developing this equation results in three individual equations, one of which can be used to recover α easily, for example:

$$\alpha = arctan\left(\frac{l_a}{l_b}\right)$$

Assuming an $\mathbf{R}'$ and $\mathbf{T}'$ are found that have the equivalent effect on the second image, then the fundamental matrix which works with the new coordinate systems F can be found from the original fundamental matrix F_0 as:

$$F = (R'T')^{-T} F_0 (TR)^{-1} = R'T'^{-T} F_0 T^{-1} R^T$$

Returning to the minimisation problem, since $F\mathbf{e} = \mathbf{e}'^T F = 0$, F must now have a simpler form with only 5 degrees of freedom (2 degrees of freedom have been removed by reducing the epipoles to only 2 parameters instead of 3):

$$\mathbf{F} \simeq \begin{bmatrix} ff'd & -f'c & -f'd \\ -fb & a & b \\ -fd & c & d \end{bmatrix}$$

The advantage of this is that all epipolar lines can now be described by only one parameter. Considering an epipolar line passing through the point $(0, t, 1)$ (by varying t all epipolar lines not parallel to the y axis can be obtained) and the epipole $(1, 0, f)$, we get a vector description of the line as $\mathbf{l}(t) = (0, t, 1) \times (1, 0, f) = (t * f, 1, -t)$. This means that the squared distance from the line to the matched point (at the origin) is now:

$$d_l(m, \mathbf{l}(t))^2 = \frac{t^2}{1 + (tf)^2}$$

Using the simplified fundamental matrix to find the corresponding epipolar line in the other image, the distance measure becomes:

$$d_l(m, \mathbf{l}'(t))^2 = \frac{(ct + d)^2}{((-ct - d)f')^2 + (at + b)^2}$$

Thus the total squared distance to minimise is given by:

$$s(t) = \frac{t^2}{1 + (tf)^2} + \frac{(ct + d)^2}{((-ct - d)f')^2 + (at + b)^2}$$

Maxima and minima of this function can be found by enforcing that $\frac{ds(t)}{dt} = 0$. Differentiating it yields a sixth order equation in terms of t which may have up to 6 real roots. Finding the real roots of this equation then gives the values of t for which maxima and minima of $s(t)$ occur. By placing these values of t into $s(t)$ and evaluating, the global minimum can be found, and coordinates for the new point positions found as orthogonal projections of the origin on to the epipolar lines represented by t. It then simply remains to undo the effect of the coordinate system change incurred by $\mathbf{R}$ and $\mathbf{T}$ to get a solution to the minimisation problem.

Note that, although all the roots of a 6th order polynomial cannot be guaranteed to be found, the effectiveness of root finders, such as the Jenkins-Traub technique used in the author's implementation, mean that, almost all the time, the global minimum will be found.

It is important to note that the addition of this method for finding error free point matches to the direct relationship of fundamental matrices to projection matrices allows a conversion of a fundamental matrix to cameras and structure without incurring any extra error in the representation.

5.5 Orienting a Reconstruction

The idea of orienting a reconstruction has already been introduced in section 3.7 (page 69). To recap, a projective reconstruction consisting of cameras P_i and structure $\mathbf{X}_j$ can be upgraded to an oriented projective reconstruction by imposing that all structure projects to the images it is observed in $\mathbf{x}_j^i$ with a positive scale factor i.e.:

$$\lambda \mathbf{x}_j^i = P_i \mathbf{X}_j \text{ where } \lambda > 0$$

This can be achieved by selecting the signs of the structure and cameras (i.e. multiplying by -1) so that this is always the case. However, because both cameras and structure have scale factors, it becomes necessary to fix one of these scale factors and enforce λ to be positive using the other scale factor alone.

To do this, the first projection matrix is left as it is, and all structure visible in the first image is projected and then multiplied by -1 if $\lambda < 0$. Each subsequent camera is then dealt with one by one, and the matches between the previous camera and the current camera obtained. The structure associated with each of these matches will already be oriented, but the current camera will not be. Subsequently, all of this structure is projected with the

current camera and, if the majority are found to project with $\lambda < 0$, the camera is multiplied by -1.

After this, any of the matches that project with $\lambda < 0$ can be removed because they represent incorrectly oriented points. All the structure in the current image can then be oriented and the process move onto the next image until there are no more images. The result is an oriented reconstruction.

5.6 Summary

This chapter has presented a number of methods for determining projective cameras and structure by utilising observed image features and either knowledge of existing structure or some multilinear form. In particular, a method was provided which allowed the fundamental matrix governing the camera positions for a pair of images to be converted, along with a set of point matches to a camera and structure representation without the addition of any more error (in the maximum likelihood sense) than that inherent in the fundamental matrix. These methods will all prove invaluable as building blocks in many of the methods to be presented in the following chapters.

Chapter 6

A Review of Projective Reconstruction for Extended Sequences

6.1 Introduction

The projective reconstruction techniques considered so far have all relied on the so called multilinear constraints that can be obtained from 2,3 or 4 images. Although effective, these techniques suffer from the major drawback of being limited to reconstructions involving at most 4 images. Since it is often desirable to reconstruct image sequences of greater length, alternative algorithms have also been developed. These can roughly be categorised into two main types, sequential and batch methods.

For sequential methods, a multilinear constraint from some part of the sequence is used to initialise structure and camera matrices for those views. New images are then added sequentially, and matching between existing structure and the new image allows calculation of the new camera matrix and updating/addition of structure. Some more recent examples of this type of system include [BZM97, BTZ96, AS98] amongst many others.

For batch methods, all structure and camera matrices are computed simultaneously. If the camera model is restricted to an affine approximation then the factorisation method of [TK92] is optimal. Similar factorisation like methods [ST96, HBS99, MH00] exist for the full projective model, but minimise only an algebraic approximation to re-projection error as well as being limited to solving for systems in which all points are visible in all views

(a problem addressed at some cost to accuracy in [Jac97]). When Euclidean distance is used for re-projection, reconstruction becomes a nonlinear problem and it is necessary to use algebraic search methods such as the well known bundle adjustment [Har94b]. The problem of projective reconstruction then becomes one of finding a good approximation to the structure and motion, in order to initialise the bundle adjustment.

Recently, an alternative hybrid approach was presented in [FZ98b] where the strength of methods for reconstruction from image triplets was utilised by producing different reconstructions for each triplet and then merging the reconstructions hierarchically, using bundle adjustment to refine the reconstruction at each stage. This removes the dependency of sequential systems on a good initial estimate, whilst achieving a higher degree of accuracy and flexibility than factorisation based approaches.

This chapter will first review all of these techniques briefly in order to set the scene for the presentation of a new algorithm, as well as to aid later comparisons.

6.2 Sequential Methods

The sequential addition of new cameras and structure to an existing reconstruction is the most well established projective reconstruction method and there are many variants on it. Sequential processing will always have uses, because it is perfectly adapted to on line algorithms where new images become available all the time (e.g. robot navigation).

6.2.1 Triplet based

The technique presented here is a sequential methodology based on image triplets. In many ways, it is similar to that presented in [BTZ96]. It is presumed that the reconstruction is to be obtained from a linear sequence of images, such as might be obtained from a video sequence. At the start of the process, the trifocal tensor associated with the first image triplet is used to initialise cameras P_j and structure $\mathbf{X}_i$ for that triplet. This can be achieved using the techniques in chapter 4. For each new image k that is then added, a new triplet is formed from the last two images of the reconstruction so far and the new image. Robust correlation based matching, using the robust estimation techniques of chapter 4, is then carried out across the triplet so as to obtain a set of points tracked for the new triplet.

Some of the points that have been matched across the new triplet will have been matched

into the common image(s) in previous triplets, and so a correlation between existing 3D structure $\mathbf{X}_i$ and features in the new image $\mathbf{x}_k^j$ can be determined. Plugging this into the standard projection equation $\mathbf{x}_k^j \simeq P_k \mathbf{X}_i$ results in a simple linear system in the unknown projection matrix P_k (see section 5.3.1 on page 98). This linear system can be solved using robust random sampling methods, such as LMedS or RANSAC, with minimal random samples of 6 points. After outliers have been removed, P_k can be estimated using linear least-squares and refined using a nonlinear least-squares method, as described in the context of trifocal tensor estimation in section 4.7.1 of chapter 4. Finally, the structure can be recalculated or updated using a Kalman filter, or the Variable State Dimension Filter (VSDF) (see [MM95]).

Given the new projection matrix P_k, it is then possible to project existing structure from the sequence as a whole into the new image, and search around the projected point for suitable image features that match the existing structure. The results of this are additional matches between existing 3D structure and image features which can be used to robustly re-calculate P_k and refine the accuracy of the results.

6.2.2 Variations

There are many variations to the above scheme. In particular, it is not uncommon to use pairs of images rather than triplets. Although pairwise schemes are clearly less robust and less accurate, they are easier to implement and faster to execute. A good example of a system of this type can be found in [BZM97] where it is applied to robot navigation.

More recently, an interesting variation was presented in [AS98] which involved 'threading' fundamental matrices together to create a set of projective camera matrices all in the same projective basis. The approach exploits a decomposition of the tensor that relates the trifocal tensor with the fundamental matrix of the first 2 views, a homography from image 1 to 2 via some arbitrary plane and the camera motion between images 2 and 3. This decomposition, when used in the standard tensor transfer functions (see equation 4.16, page 92), provides linear constraints on the camera motion between images 2 and 3 without the use of any 3D structure. Whilst avoiding 3D structure can be useful, transferring points using the tensor is inaccurate because the quantity being minimised is meaningless in a least-squares sense. Altogether, this makes the stability of the method questionable.

Note that the merging algorithms of the next chapter achieve a similar minimal approach, but with a meaningful error criteria.

6.3 Batch methods

Batch methods cover all those methods that attempt to solve for all 3D structure and 3D cameras at the same time. If a projective reconstruction is to be determined and Euclidean re-projection error is used then the ideal method is that of bundle adjustment. However, Euclidean distance measures give rise to nonlinear equations, and to provide a linear approximation, a factorisation type method can be employed. Since these two cases are distinctly different, they will be treated separately in the following two subsections.

6.3.1 Bundle Adjustment

The bundle adjustment [Bro58] is a well known and very well established method for providing a nonlinear refinement of all structure and all cameras in a scene (see [Sla80, TMHF00, Har92, SKZ99]). Although it was initially designed for refining manual reconstructions, it has proved simple to adapt to refining automated reconstructions, and even projective reconstructions.

The basis of bundle adjustment is to find the least-squares solution that minimises the re-projection error:

$$\sum_{ij} d_E^2 \left(P_i \mathbf{X}_j, \mathbf{x}_j^i \right) \tag{6.1}$$

for all cameras P_i in image i, 3D structure $\mathbf{X}_j$ and associated 2D image features $\mathbf{x}_j^i$. This equation is nonlinear, involving unknowns for both structure and cameras as well as an unknown scale factor that has to be eliminated by dividing through. For projective cameras, in general, the best that can be done to minimise the exact error measure in equation 6.1 is to refine a supplied initial solution using a gradient descent technique such as Levenberg-Marquardt or Newton iteration (see appendix A for a detailed description).

Whilst it would be quite straightforward simply to use the error measure in equation 6.1 in a conventional Levenberg-Marquardt implementation, it would unfortunately not be practical because of the size of the problem involved. For example, consider a normal scene involving 40 images with 2000 points. This leads to $40 * 11 + 2000 * 3 = 6440$ unknowns, an intractable problem when using normal methods. Nevertheless a solution is still feasible, because the Jacobian matrix for the problem has a special sparse block structure. This leads to a similar sparse block structure for the normal equations used in Levenberg-Marquardt or Newton iteration. If the sparsity is properly exploited, it is possible to obtain an enormous simplification in the solution of the normal equations. See appendix B for a full description

of the specialised bundle adjustment method in [Har92], including some minor practical refinements.

6.3.2 Factorisation Methods

Factorisation methods have received much attention over the years, and have proved very difficult to get as accurate as the more conventional relating 3D to 2D structure used in the sequential methods. Recently however, results have become very good.

The basis of factorisation approaches lies in a closer analysis of the projection equation, which after making the scale factor λ explicit can be compactly written as:

$$\lambda_j^i \mathbf{x}_j^i = P_i \mathbf{X}_j \tag{6.2}$$

given $\mathbf{x}_j^i$ as the projection of jth item of 3D structure $\mathbf{X}_j$ into image i with camera matrix P_i. Sometimes the scale factors λ_j^i are referred to as the projective depths of the points (or just the depths). The set of depths for all the projections of a particular 3D point are not unique, but are defined subject to an arbitrary non zero scale factor per point as well as per image (i.e. all depths for a particular image are themselves also defined up to an arbitrary non zero scale factor). If these scale factors are fixed (for example making λ in image 1 equal to 1), then the resultant so called kinetic depths have been shown in [Hey95, Spa94] to be independent of the chosen image coordinate systems and to completely describe the multi-imaging situation.

The best established factorisation method for projective reconstruction is based on the method of [TK92] for producing an affine reconstruction. Stacking equation 6.2 into matrix form, it is possible to introduce the following matrix W for the coordinates of all n points in all m images:

$$W = \begin{bmatrix} \lambda_1^1 \mathbf{x}_1^1 & \lambda_1^2 \mathbf{x}_1^2 & \dots & \lambda_1^m \mathbf{x}_1^m \\ \lambda_2^1 \mathbf{x}_2^1 & \lambda_2^2 \mathbf{x}_2^2 & \dots & \lambda_2^m \mathbf{x}_2^m \\ \vdots & \vdots & \ddots & \vdots \\ \lambda_n^1 \mathbf{x}_n^1 & \lambda_n^2 \mathbf{x}_n^2 & \dots & \lambda_n^m \mathbf{x}_n^m \end{bmatrix}$$

The matrix W will be referred to as the re-scaled measurement matrix because it also contains

the scale factors. Similarly, a matrix X is defined for the 3D structure x_i in image i:

$$X = \begin{bmatrix} x_1 & x_2 & \dots & x_n \\ y_1 & y_2 & \dots & y_n \\ z_1 & z_2 & \dots & z_n \\ w_1 & w_2 & \dots & w_n \end{bmatrix}$$

and finally, a matrix P for each of the projection matrices:

$$P = \begin{bmatrix} P_1 \\ P_2 \\ \vdots \\ P_m \end{bmatrix}$$

Now the original projection equation 6.2 can be written as the following matrix product

$$W = PX$$

The basis of most factorisation methods is to attempt to decompose the left half of this equation (or a similar one) to get the right hand side of the equation. It is worth noting that, if the correct projective depths λ_j^i are used to determine W, then W has a rank of at most 4. The matrix W can then be factored using a singular value decomposition to give:

$$W = U\text{diag}\left(\sigma_1, \sigma_2, \dots, \sigma_s\right) V$$

for singular values σ_s, where $s = \min\left(3m, n\right)$ and the singular values are arranged in descending order of magnitude. Since W is of rank 4 only, σ_i for which $i < 4$ should be non zero, and so only the first 4 columns of U and rows of V contribute to the matrix product. Given U' and V' as the matrix of these first 4 columns or rows, W can be written as:

$$W = U'_{3m\times 4}\text{diag}\left(\sigma_1, \sigma_2, \sigma_3, \sigma_4\right) V'_{4\times n} = U'\Sigma V'$$

Any factorisation of Σ into two 4×4 matrices Σ' and Σ'', $\Sigma = \Sigma'\Sigma''$ leads to:

$$W = \underbrace{U'\Sigma'}_{\hat{U}}\underbrace{\Sigma''V'}_{\hat{V}} = \hat{U}_{3m\times 4}\hat{V}_{4\times n}$$

This is valid regardless of the factorisation of Σ because the whole reconstruction is subject to an arbitrary projective transformation of the world space (both structure and cameras are unknown). The matrix $\hat{U}$ can be interpreted as a collection of m (3×4) projection matrices, and P_i, $\hat{V}$ as a collection of n 4 vectors X_j representing the 3D structure.

Finding the Projective Depths

Given that the scale factors used in the re-scaled measurement matrix are unknown, an initial result can be obtained by using the above technique and just setting all scale factors to one. In general, this produces poor results, and other techniques for determining the scale factors should be used. If the camera model being used is the affine model, then the model is designed so that these scale factors come for free, but if a full projective model is used, a little more work is necessary. Following [ST96], it is possible to make use of the so called closure constraints to determine the projective depths.

The derivation of the closure constraints is somewhat lengthy and will be omitted here. The interested reader is referred to [Tri95, ST96] for more details. Briefly, the constraints are:

$$F_{ij}\left(\lambda_j \mathbf{x}_j\right) + \mathbf{e}_{ij} \wedge \left(\lambda_i \mathbf{x}_i\right) = 0 \tag{6.3}$$

for 2 images i and j, epipoles $\mathbf{e}_{ij}$ and fundamental matrix F_{ij}. For 3 images i, j and k with trifocal tensor T_i^{jk}, the closure constraint is given by:

$$T_i^{jk}\left(\lambda_i \mathbf{x}_i\right) - \left(\lambda_j \mathbf{x}_j\right)\left(\mathbf{e}_{ki}\right)^T + \mathbf{e}_{ji}\left(\lambda_k \mathbf{x}_k\right)^T = 0$$

These equations can be rearranged to give the scale factor for an image in terms of the scale factors in the other image(s). Naturally, the 3 image closure constraints will be more robust, but also much more complex. The simplest scheme involves just using equation 6.3. By estimating a sufficient number of fundamental matrices, it is possible to amass a system of homogeneous linear equations in terms of the unknown depths.

However, there is a further problem that the fundamental matrices and epipoles can themselves be recovered only up to an unknown scale factor. To overcome this scale factor problem, it is possible to use only the minimal set of fundamental matrices, i.e. F_{12}, F_{23}, ..., $F_{m-1}m$, and use the unknown scale factors for each image to absorb the arbitrary relative scale of F and $\mathbf{e}$. If redundant equations are used, it becomes essential to choose self consistent scaling for the estimated fundamental matrices and epipoles as described in [Tri95].

A further advantage of using a minimal set of fundamental matrices is that it is possible to simply chain equation 6.3 together to find successive scale factors. Because the scale factors for each point are only defined subject to a scale factor, the first scale factor can be fixed for example to 1 and then the rest determined consecutively by solving equation 6.3 in

least-squares to find λ_j in terms of λ_i:

$$\lambda_i = \frac{(\mathbf{e}_{ij} \wedge \mathbf{x}_i)\,(F_{ij}\mathbf{x}_j)}{\|\mathbf{e}_{ij}\mathbf{x}_i\|^2}\lambda_j$$

An additional improvement has been added to the algorithm for the purposes of this work. Solving equation 6.3 in a least-squares sense involves minimising a quantity that is not really meaningful. Instead, before using equation 6.3, the point match across that image pair is Hartley-Sturm corrected (see section 5.4.3 on page 102). Since the corrected points will fit the epipolar geometry perfectly, the equivalence expressed by the image pair closure constraint will be exact. Consequently, instead of minimising an arbitrary quantity, the orthogonal Euclidean distance from points to epipolar lines has been exactly minimised, resulting in vastly improved results.

One final improvement of note is available, but not assessed in this work. That is the recent development [MH00], which calculates the unknown scale factors by solving a generalised eigenvalue problem derived from a subspace constraint on all the projections of a 3D point. However, this still minimises a meaningless value and it seems unlikely it will produce any large improvement (the author has not tested this hypothesis).

Other Factorisation Methods

Alternative approaches for factorisation do also exist. Most recently in [HBS99], a method was proposed that relies on subspace methods only, and hence provides the significant advantage of being independent of the world coordinate system. However, the method is somewhat slower and recovers 3D structure only. In order to determine the camera matrices, it is necessary to solve for them in a least-squares sense using the known depths and 3D structure. Since this can only be performed using a linear approximation, there is some degradation in performance.

An alternative, new and very promising approach is presented by the so called plane and parallax methods. These are based on the realisation that the most significant aspect of a camera description is the centre of projection. Rotations and calibration changes produce trivial image deformations that can be described by 2D homographies, whereas translation results in the parallax effects from which structure can be determined. In order to cancel out the simple calibration and rotation deformations, all image points are placed into the coordinate system of a particular 3D plane in the images. The disparity of points then becomes the projective distance of points from the selected plane and the underlying 3D and matching tensor geometry becomes much simpler.

However, there are clear limitations of this technique which have yet to be addressed. For example, it is necessary to be able to accurately determine the same plane in all images, which is not always feasible if there is a lot of camera movement and no easily identifiable planes. Secondly, it is hard to know which plane is best to use and, thirdly, a bad choice of plane can seriously distort the original images. For more details see [Tri00]

6.4 Summary and Conclusions

A number of algorithms have been presented in this chapter, all of which have their own advantages and drawbacks. In an attempt to keep this all in perspective, a brief summary of the algorithms will now be given, along with a brief statement of the advantages and disadvantages of each approach.

- *Sequential methods:* Methods based on a sequential methodology attempt to produce a reconstruction an image at a time. Structure is initialised at the start of the sequence, and images added one by one. For each new image, matching between existing structure and points in the new image allows new camera matrices and structure to be calculated.

  The big advantages of this methodology are that it is very simple to implement, and that images are handled on line making it perfect for applications for which images become available over time (e.g. robot navigation). On the other hand, it suffers from an accumulation of error as the sequence increases in size. In general, the reconstruction has a bad tendency to drift with points at the end of the sequence in a very different coordinate frame to those at the beginning. This means a small section of poor quality reconstruction can throw the whole reconstruction.

- *Bundle Adjustment:* This is a nonlinear refinement method that minimises the maximum likelihood error measure of distance between projected features and observed image features. Since this measure is nonlinear, it is necessary to use gradient descent or iterative methods to refine some initial guess.

  A great benefit of bundle adjustment is that it is a maximum likelihood estimator. However, because the criterion minimised is nonlinear, the obvious drawback of bundle adjustment is that it is highly dependent on the quality of the initial guess. It is also fairly complex to implement, requiring special handling of the huge sets of equations to provide a tractable solution.

- *Factorisation:* If the projection equations for all points in all images are stacked together it becomes possible to factorise the joint matrix of image points into the structure and camera positions.

  The principal reason for the attractiveness of factorisation methods are that they solve for all camera matrices and structure at the same time, thus balancing error evenly across the whole sequence. The simplicity of a simple factorisation also results in a very quick method, especially since there is no need for intermediate results (on the other hand this also means there are no intermediate results to guide or aid matching). Unfortunately the cost of all this is that the methods can produce pretty bad results, because they do not minimise a meaningful error measure. This is largely remedied by use of the closure constraints (see section 6.3.2), but again lack of a meaningful error measure causes bad degradation in results as the size of the image sequence increases. Finally, they become impractical for long sequences, because points that are missing projections in images cannot be handled without further loss of accuracy.

As can be seen existing methods offer little scope for flexibility. Given exact knowledge of the particular application, it is usually clear which method to use, but long sequences remain a significant problem.

Chapter 7

Robust Merging Based Projective Reconstruction

7.1 Introduction

It has been shown in previous chapters that a key problem to be addressed in projective reconstruction is to find an initial estimate of the scene structure and camera motion, using only the observed projections of a real scene.

The previous chapter reviewed a number of well established methods for finding this initial approximation. However, all the techniques were found to suffer from significant drawbacks. Sequential methods, which continually add new structure and cameras to an existing reconstruction are heavily reliant on a good initial estimate of structure which can be updated across the whole sequence. Factorisation methods overcome these problems by calculating all cameras and structures at the same time, but suffer from a lack of robustness and flexibility for purposes of further matching, or when points are missing projections in some images. When combined with the lack of a meaningful error criterion, it becomes clear that factorisation methods are at best inaccurate over long sequences and at worst inapplicable.

To overcome these problems, it was proposed in [FZ98b] to reconstruct small sequences and then merge them together hierarchically to create larger sequences, using bundle adjustment to refine the results at each stage. This chapter will present a new method for projective reconstruction that generalises this hierarchical approach to a merging based approach to reconstruction. The previously unexplored flexibility of this merging approach is

then examined, and applied to specific applications such as image sequences.

Finally, a large number of new and robust techniques for the merging of projective reconstructions are introduced and it is then shown that the quality of reconstruction from these new methods are dramatically improved. In practice, it is shown that the improvements are so great that merging reconstruction can be used to supplant even methods based on multilinear forms such as the trifocal tensor. Finally, to justify these claims, a comprehensive comparison with existing projective reconstruction methods is given.

7.2 Merging Methodology

As already mentioned, the basis of the new method presented in this chapter is to merge projective reconstructions from small sub-sequences of images to create a reconstruction for the whole sequence. These initial reconstructions could come from multilinear forms such as the fundamental matrix or trifocal tensor. Alternatively, small sub-sequences could be reconstructed using factorisation techniques, which can be achieved robustly by using recently proposed robust methods for sequences of arbitrary length [SZH00].

Initially, this means the scene is represented by a set of independent projective reconstructions. Since independent projective reconstructions of the same scene can be related by a change of projective basis, it is necessary to use corresponding scene structure to compute this basis change in the form of an arbitrary projectivity of $\mathcal{P}^3$. This process will be described in detail in section 7.4 with the associated robust method described in section 7.5.

Assuming availability of robust techniques to compute the aligning homographies, registration proceeds robustly merging sub-sequences into new larger sub-sequences before application of a bundle adjustment to redistribute error in an optimal manner. An example of this is given in figure 7.1 where, starting from a sequence of image triplets, the triplets are merged hierarchically so as to keep two images overlapping at all times.

7.2.1 Overlap and Correspondence

An important consideration in the merging approach is that in order to register sub-sequences, common structure needs to be found. If sub-sequences share one or two images then establishing these correspondences is trivial because some features in the common images will relate to 3D structure in both of the different sequences. However, for the case of zero overlapping images, it is necessary to return to the images themselves and look for matches

1. Robustly produce cameras and structure for the first image pair.
2. For each new image, produce cameras and structure for the image pair containing the new image and the last image in the existing reconstruction.
3. Use the 1 view overlapping techniques to robustly merge the two reconstructions.
4. Repeat from step 2 until no more images.
5. (Optional) Bundle adjustment of the complete sequence.

Table 7.1: Sequential merging for image pairs

between the features in both sub-sequences. This is an important advantage of using overlapping sequences.

A further advantage conferred by using more than one overlapping image is that of outlier detection. If a point is found to track inconsistently, it can be flagged as an outlier. For example, if a point has projections u, u', u'' in images 1,2 and 3 from triplet one, but has projections u', v, u'' in images 2,3,4 from triplet two, then something is clearly wrong and the robust schemes in section 7.5 should be used to remove all outlying projections.

7.3 Merging Schemes

One of the advantages of the merging based approach to reconstruction is a great deal of flexibility. It is a general technique and does not necessitate any assumptions about the nature and distribution of images that are available, e.g. that they are all in a closely connected sequence. As a consequence, it may be used for many different reconstruction applications, and in many different ways. To illustrate the flexibility, the merging approach has been used here to design algorithms capable of performing reconstruction for different types of image acquisition. This can be achieved using exactly the same set of techniques for each algorithm, relying only on the existence of a specific feature matcher suitable for the task in hand. Indeed, the details of the merging have been deliberately omitted until a later stage because there is no need to adapt them to the individual algorithms:

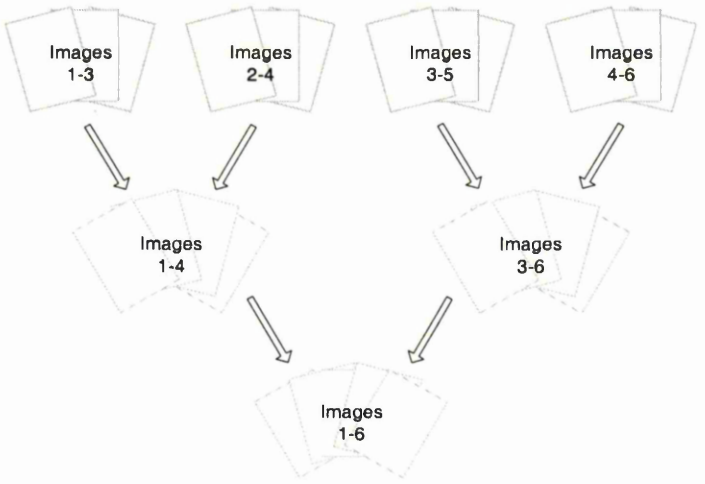

Figure 7.1: Hierarchical merging of sub-sequences - 2 image overlap scheme

7.3.1 Sequential Merging

For this scheme, merging is arranged so as to reconstruct in a similar manner to the sequential reconstruction method based on re-sectioning (see section 6.2 on page 108). In the re-sectioning method, new images are added to an existing reconstruction one by one, and the new camera estimated by the relationship between existing structure and associated matches in the new image.

The merging techniques are also applicable to this form of problem, and to illustrate this an image pair based scheme will now be presented. Processing starts as for conventional sequential reconstruction, by producing a reconstruction for the first image pair using the techniques of chapters 4 and 8. When a new image is added, it is paired with the last image from the existing reconstruction, and robust methods are used to match and produce an independent reconstruction for the pair. The result is two reconstructions overlapping by one image, which can be robustly merged using the techniques of this chapter. The sequential merging algorithm is summarised in table 7.1.

7.3.2 Hierarchical Merging

If a complete sequence of images are available prior to processing, then the sequential method just outlined is not necessarily the best method. This is because it tends to favour the images used for the initial reconstruction, and because it has a tendency to drift out of the original projective coordinate frame when near degenerate sections of sequence are encountered. If all

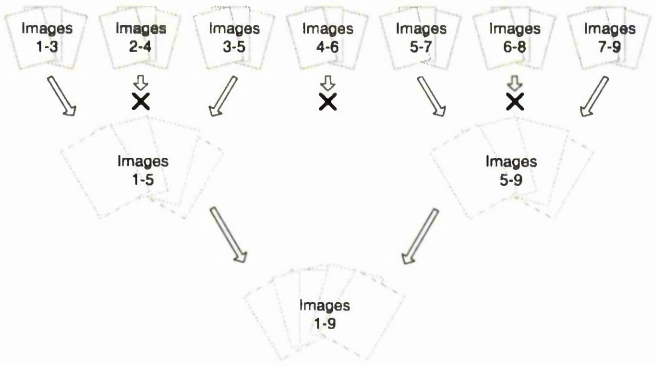

Figure 7.2: Hierarchical merging of sub-sequences - 1 image overlap scheme

images are available at the start of processing, and consecutive images have small baselines then an alternative is to merge sub-sequences hierarchically.

An example of this hierarchical reconstruction is given in figure 7.1 where, starting from a sequence of image triplets, the triplets are merged hierarchically so as to keep two images overlapping at all times. Of course, this scheme is highly flexible and could start from arbitrary length sequences overlapping by arbitrary numbers of images.

The advantages of hierarchical merging are that it tends to distribute error more evenly, is easily able to handle missing images or sub-sequences, that the bundle adjustment can sometimes be faster when it is split up in this manner and that it is highly suited to parallel or distributed implementation. However, for larger sequences, the bundle adjustment can become quite slow, and so if desired, bundle adjustment can be stopped at a certain level and merging simply proceed without it. In these cases it is very important to reconstruct or update the 3D of all merged points and cameras. A bundle adjustment can then be performed at the end when the complete sequence is available.

Merging Schemes

As mentioned, hierarchical merging is very flexible, allowing sub-sequences to be omitted and overlap to occur for differing numbers of views. Consequently, there are many ways to vary the overlapping scheme, for example:

- For greater computational efficiency only one view needs to be kept overlapping, as is illustrated in figure 7.2
- To also produce speed increases, hierarchical registration can stop at a certain level, and registration proceed sequentially as described in section 7.3.1.

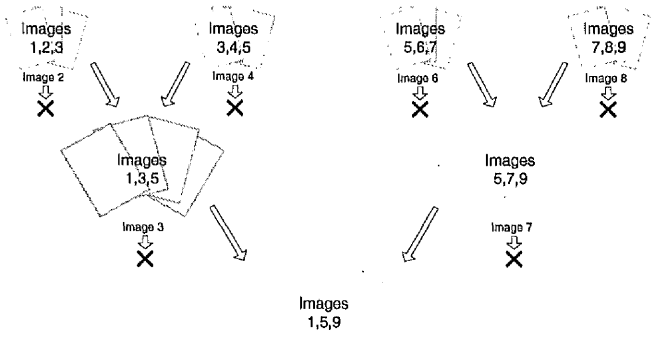

Figure 7.3: Hierarchical merging of sub-sequences with Image Dropping

- To keep the quantities of data more manageable images can be dropped as illustrated in figure 7.3. It is only a good idea to do this when merging small sequences. In these cases, the inaccuracies of the small baseline mean that it is effective only to keep the large baseline matches. Note, it is not a good idea to downgrade triplets to pairs in this manner because of the significant weakening of the matching constraint.

7.3.3 Application to Sparse Collections of Images

The final problem to be addressed with merging based reconstruction, is to produce reconstructions for collections of images which have no temporal connection - for example, a collection of images of the same scene taken arbitrarily using a normal camera. In these cases, a different approach is suggested because the relationship between images is arbitrary and not ordered.

Firstly, a pair-wise reconstruction is built for each image pair for which a fundamental matrix can be calculated. The problem then becomes one of choosing the best way to merge the pairs together so that each image is included in the sequence.

This is done by building a graph structure, with pair-wise reconstructions at the nodes, and links between nodes indicating that the pair-wise reconstructions have some common 3d structure. Each node stores the mean re-projection error, and each link stores the amount of shared structure between reconstructions.

Choosing the best way of merging sequences is therefore equivalent to finding a spanning tree for the graph, ensuring that each image appears in at least one node of the tree, the sum of re-projection errors at each node is minimised, and the amount of structure shared between each node is maximised. This is achieved using a modified version of Prim's algorithm

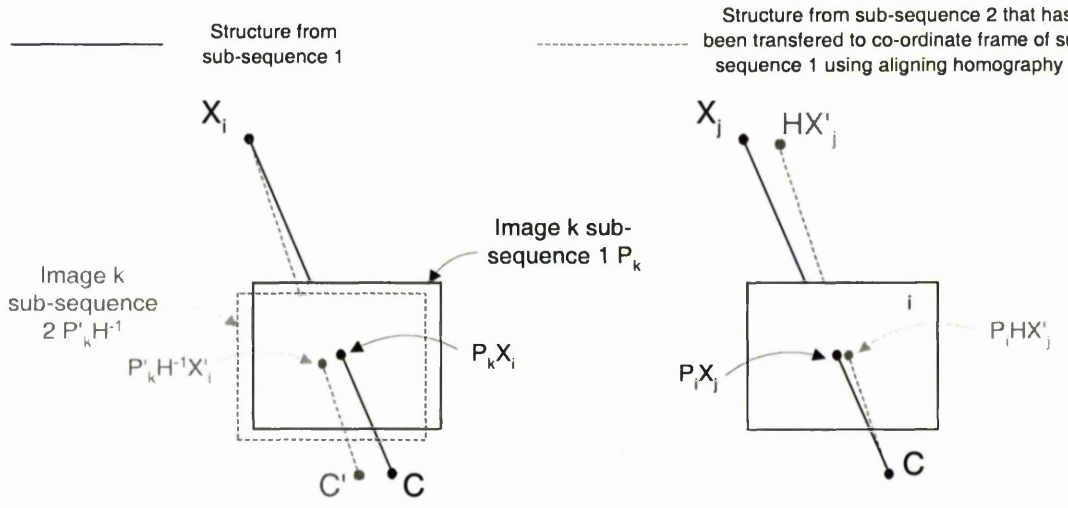

Figure 7.4: Illustration of re-projection constraints making use of matching cameras (fig. a) or matching structure (fig. b). In figure a the same item of structure $\mathbf{X}_i$ from sub-sequence 1 is projected through each of the matching cameras, but the camera from sub-sequence 2 P'_k is modified by the aligning homography so it is in the sub-sequence 1 co-ordinate frame. In figure b matching structure $\mathbf{X}_j \leftrightarrow \mathbf{X}'_j$ is projected through the same sub-sequence 1 camera P_i, but the structure from sub-sequence 2 $\mathbf{X}'_j$ is modified by the aligning homography so that it is in the co-ordinate frame of sub-sequence 1. Note that for purposes of illustration the mis-alignment between sub-sequence 1 and transferred sub-sequence 2 structures has been greatly exaggerated

[Wei99].

Once a spanning tree is constructed, the merging can take place. The tree is traversed multiple times. During each pass, the leaf nodes in the tree are robustly merged with their immediate parent. This is repeated until a single (complete) sequence is left at the root of the tree.

Note that this particular reconstruction algorithm is due largely to Simon Gibson.

7.4 Merging Different Projective Reconstructions

So far, the core problem associated with the merging based approach has not been addressed. That is to say, given reconstructions for two sub-sequences of the same scene, how can they

be robustly merged into one reconstruction in the same projective basis? This section aims to describe in detail some methods for this registration process, under the basic assumption that some of the 3D points and possibly camera views are common to both sub-sequences, and that at least some of these common points or camera views are known.

Suppose point j has matching structure represented by the coordinate vector $\mathbf{X}_j$ in the first sub-sequence and $\mathbf{X}'_j$ in the second sub-sequence. Since projective reconstructions are equivalent if they are related by a projectivity H of $\mathcal{P}^3$, it follows that:

$$\mathbf{X}_j \simeq H\mathbf{X}'_j \tag{7.1}$$

$$P_i \simeq P'_i H^{-1} \tag{7.2}$$

where P_i and P'_i are camera matrices for the same view i of the scene, but in the differing projective basis of the two sub-sequences. This provides two approaches to applying constraints on the unknown projectivity H; either by using matches between items of structure $\mathbf{X}_j \leftrightarrow \mathbf{X}'_j$ or by using matches between projection matrices $P_i \leftrightarrow P'_i$.

With real data, equations 7.1 and 7.2 will not be satisfied exactly, and so an error minimising estimate must be found instead. Since the reconstructions are assumed to be projective it will be most sensible to work with image measurements alone, and use some form of re-projection error. This re-projection error can exploit either matches between structure or matches between cameras. For a geometric interpretation of this, the reader is referred to figure 7.4,

One approach is to take matches between structure $\mathbf{X}_j \leftrightarrow \mathbf{X}'_j$ and using the unknown projectivity H convert these matches into the other sub-sequence and project them (figure 7.4b).

$$\sum_{ij} \frac{1}{n_1} d^2\left(P_i H \mathbf{X}'_j, P_i \mathbf{X}_j\right) + \sum_{kj} \frac{1}{n_2} d^2\left(P'_k H^{-1} \mathbf{X}_j, P'_k \mathbf{X}'_j\right) \tag{7.3}$$

for all structure j, images in sub-sequence 1 i, images in sub-sequence 2 k for which this structure has been observed, and number of residuals in sequence 1 and 2 $n_1 = \sum_{ij} 1$, $n_2 = \sum_{kj} 1$. This measure basically states that, for a common pair of 3D points, we wish to convert each of the 3D points into the other sub-sequence, and minimise the re-projection error in all images of that other sub-sequence in which it is visible. Note that figure 7.4b provides an illustration of what this means for the first part of equation 7.3 only.

Whilst the measure just suggested makes use of structure matches, it follows that it must also be possible to create an error measure that only relies on matches between projection matrices $P_k \leftrightarrow P'_k$ (figure 7.4a). Since this does not require matching structure, any structure

can be used regardless of whether it has been found in both sub-sequences. In particular, for structure X_i in sub-sequence 1 projecting into matching images k, and structure X'_j from sub-sequence 2 projecting into the same images k:

$$\sum_{ki} d^2\left(P'_k H^{-1}\mathbf{X}_i, P_k\mathbf{X}_i\right) + \sum_{kj} d^2\left(P_k H\mathbf{X}'_j, P'_k\mathbf{X}'_j\right) \tag{7.4}$$

The advantage of this criterion is that it allows extra constraints from unmatched structure as well as providing an error measure that is not susceptible to corruption by structure mismatches between the sub-sequences (because it does not use matches). However, not using matches means that only 11 parameters of H can be constrained if a single image is used, rather than the full homography. Note also that figure 7.4a provides a geometric interpretation of the first half of equation 7.4 only.

In all the previously given criteria, the distance function $d(\mathbf{X}, \mathbf{Y})$ takes two homogeneous n-vectors representing points in $\mathcal{P}^{n-1}$ and can represent either Euclidean distance:

$$d_E^2(\mathbf{X}, \mathbf{Y}) = \sum_{k=1}^{n-1}\left(\frac{X_k}{X_n} - \frac{Y_k}{Y_n}\right)^2 \tag{7.5}$$

or algebraic distance:

$$d_A^2(\mathbf{X}, \mathbf{Y}) = \sum_{k=1}^{n-1}\left(X_k Y_n - Y_k X_n\right)^2 \tag{7.6}$$

Also note that, since all of $P_i, P'_i, \mathbf{X}_j, \mathbf{X}'_j$ are computed subject to errors, a minimum of either equation 7.3 or equation 7.4 does not represent a maximum likelihood estimator for the whole structure and motion problem - only for the estimation of H.

7.4.1 Merging with One Overlapping View:

The previous section has discussed two constraints that can be imposed on the aligning projectivity H. These assume matches between structure are available and/or matches between cameras are available. Since the application of these criteria depend on the number of camera and structure matches that are available it is appropriate to use different methods based on the data available. In this section the most useful case for the task in hand will be examined - that of one overlapping view (hence one camera match), combined with a set of outlier free structure matches.

Whilst the criterion in equation 7.4 allows constraints to be placed on the merging projectivity H using the matching camera, in this case the existence of only one such match

means the equivalence can be exact (i.e. equation 7.2 - $P_i \simeq P'_i H^{-1}$ can be satisfied exactly). In [FZ98b], it was shown how these overlapping camera matrices, P and P', can be registered exactly using equation 7.2, and hence be used to remove 11 degrees of freedom from the 15 degrees of freedom in the projectivity relating the two sub-sequences.

Because a projection matrix P can be transformed to a canonical form $[I|0]$ by multiplication with its pseudo inverse P^+, it follows that the projectivity H converting P to P' exactly can be found as P^+P' subject to four extra degrees of freedom $\mathbf{a}$. Note here that both P^+ and P' have been upgraded from their usual 3×4 form to a 4×4 form. This is simply upgrading the dimension into which the projection matrix projects, and considering the image space as embedded within the structure space (which it is - it is a plane). This requires fixing the extra information to some arbitrary value. In this case, it can be expressed using notation with the multiplication $PI_{4\times4}$.

Given this, if we wish to find the projectivity H that minimises equation 7.3 subject to $PH = P'$, then the solution will belong to the 4-parameter family of homographies:

$$H(v) = P^+P' + \mathbf{h}\mathbf{a}^T \tag{7.7}$$

where $\mathbf{h}$ is the null-space of P. Plugging this expression for H into equation 7.3, substituting $\mathbf{x}_k = P_i\mathbf{X}_j$ and ignoring the terms involving H^{-1}, yields two linear equations in terms of $\mathbf{a}$ per projected point:

$$(\mathbf{p}_1 - \mathbf{x}_k\mathbf{p}_3)\left(\mathbf{h}\mathbf{a}^T\mathbf{X}'\right) = (\mathbf{x}_k\mathbf{p}_3 - \mathbf{p}_1)\left(P^+P'\mathbf{X}'\right) \tag{7.8}$$

where $\mathbf{p}_n$ represents the nth row of P and $k \in (1, 2)$. This is a very similar scheme to that presented in [FZ98b], but using re-projection error instead of 3D error (3D error will be discussed later, see equation 7.17).

Note that, by moving to re-projection, the exact alignment of the overlapping cameras has caused some interesting effects in the error measure. Those points that project into the overlapping images will impose no constraints on the unknowns $\mathbf{a}$ in the transformation, and so should not be considered. This must be the case because both the projection matrices will be exactly the same after registration using the above technique.

An Alternative Formulation

If the registered overlapping projection matrix is transformed to the form $[I_{3\times3}|0]$, then the null-space $\mathbf{h}$ becomes $(0, 0, 0, 1)$ and so $\mathbf{h}\mathbf{a}^T$ has a simplified form. Consequently, the

constraints on H provided by equation 7.7 can be formulated slightly differently:

$$\begin{aligned} H &= [I|0]^{+}\left([I|0]\, I_{4\times4}\right) + (0,0,0,1)^{T} v^{T} \\ &= I_{4\times4} + A \text{ where } A = \begin{bmatrix} I_{3\times3} & 0 \\ \mathbf{a}_{1,2,3} & \mathbf{a}_4 \end{bmatrix} \end{aligned} \tag{7.9}$$

In this equation, $\mathbf{a}_{1,2,3}$ represents 3 unknown parameters and $\mathbf{a}_4$ an unknown scaling. For greater simplicity in later descriptions, it will be assumed at this point that all structure and cameras have been transformed so that the overlapping camera matrix P_o for both sub-sequences is of the form $[I_{3\times3}|0]$, i.e for the first sub-sequence:

$$\begin{aligned} \hat{P}_i &= P_i P_o^{+} \\ \hat{\mathbf{X}}_j &= P_o \mathbf{X}_j \end{aligned}$$

This changes equation 7.9 to the much simpler form of $\hat{H} = A$ and so equation 7.8 becomes:

$$(\hat{\mathbf{p}}_k - \mathbf{x}_k \hat{\mathbf{p}}_3)\, A\hat{\mathbf{X}}' = 0 \tag{7.10}$$

After this transformation, the original H can easily be recovered as $H \simeq PAP'^{+}$.

Improving the Algorithm

There may appear to be little gain by using re-projection error instead of 3D error because both error measures are meaningless. However, an important improvement can be found by relating algebraic and Euclidean distance for re-projection error. Euclidean error d_E gives:

$$\frac{\hat{\mathbf{p}}_k A\hat{\mathbf{X}}'}{\hat{\mathbf{p}}_3 A\hat{\mathbf{X}}'} - \mathbf{x}_k = d_E \tag{7.11}$$

whereas algebraic error gives (expanding equation 7.10):

$$\hat{\mathbf{p}}_k A\hat{\mathbf{X}}' - \mathbf{x}_k \hat{\mathbf{p}}_3 A\hat{\mathbf{X}}' = d_E \hat{\mathbf{p}}_3 A\hat{\mathbf{X}}' \tag{7.12}$$

It can easily be seen that equation 7.11 can be obtained by dividing equation 7.12 by $\hat{\mathbf{p}}_3 A\hat{\mathbf{X}}'$. This is very useful, because by using the constraint from equation 7.1 that $\hat{\mathbf{X}} = \lambda A\hat{\mathbf{X}}'$, an equivalent value to $\hat{\mathbf{p}}_3 A\hat{\mathbf{X}}'$ can be found and then used to weight the measure in equation 7.12, thus approximating Euclidean re-projection.

The approximation to $\hat{\mathbf{p}}_3 A\hat{\mathbf{X}}'$ can be found by projecting the equivalent structure from the other sub-sequence $\hat{\mathbf{p}}_3 \lambda\hat{\mathbf{X}}$. Of course $\hat{\mathbf{p}}_3 A\hat{\mathbf{X}}'$ and $\hat{\mathbf{p}}_3 \lambda\hat{\mathbf{X}}$ will not normally be exactly

the same due to differing reconstructions in the two sub-sequences, but assuming accurate reconstruction they will be similar enough.

In practice equation 7.1 does not express an exact equivalence between $\hat{\mathbf{X}}$ and $\hat{\mathbf{X}}'$, and the two are related by the unknown transformation A and a scale factor λ that is common to all structure:

$$\hat{\mathbf{X}} = \lambda A \hat{\mathbf{X}}' \tag{7.13}$$

Fortunately, because the first 3 rows of A represent an identity mapping, the scale factor λ can easily be determined from $\hat{\mathbf{X}}_k = \lambda I_{3\times 3}\hat{\mathbf{X}}'_k$ for $k \in \{1,2,3\}$. It has been found in experiments that it is best determined by solving using least-squares and the 3 available constraints on λ to give:

$$\lambda = \sqrt{\frac{1}{3}\left(\frac{\hat{\mathbf{X}}_1^{\ 2}}{\hat{\mathbf{X}}'_1} + \frac{\hat{\mathbf{X}}_2^{\ 2}}{\hat{\mathbf{X}}'_2} + \frac{\hat{\mathbf{X}}_3^{\ 2}}{\hat{\mathbf{X}}'_3}\right)}$$

Since λ is now known, a new and very nearly Euclidean linear distance measure can be used in the minimisation:

$$\frac{\hat{\mathbf{p}}_k A \hat{\mathbf{X}}'}{\hat{\mathbf{p}}_3 \lambda \hat{\mathbf{X}}} - \frac{\mathbf{x}_k \hat{\mathbf{p}}_3 A \hat{\mathbf{X}}'}{\hat{\mathbf{p}}_3 \lambda \hat{\mathbf{X}}} = d_{EApp} \tag{7.14}$$

A Further Improvement: Adding the Inverse

By reformulating the problem as in equation 7.9, the affine parameters come in an easily invertible form to give:

$$A^{-1} \simeq \begin{bmatrix} a_4 & 0 & 0 & 0 \\ 0 & a_4 & 0 & 0 \\ 0 & 0 & a_4 & 0 \\ -a_1 & -a_2 & -a_3 & 1 \end{bmatrix}$$

The implications of this simple inverted form is that it is now possible to keep the inverse part of equation 7.3 linear. Considering only this part for now, substituting $\mathbf{x}'_k$ for $P'_k\mathbf{X}_j$ and following the reasoning in the previous section, the following error measure is arrived at:

$$\hat{\mathbf{p}}'_k A^{-1}\hat{\mathbf{X}} - \mathbf{x}'_k \hat{\mathbf{p}}'_3 A^{-1}\hat{\mathbf{X}} = d_E \hat{\mathbf{p}}'_3 A^{-1}\hat{\mathbf{X}}$$

It would be desirable to use the weighting trick again and find μa_4 in $\hat{\mathbf{X}_k}' = \mu a_4 \hat{\mathbf{X}_k}$, for $k \in \{1,2,3\}$. Identifying this with equation 7.13 gives $\mu a_4 = \frac{1}{\lambda}$ and a new error measure of:

$$\frac{\lambda \hat{\mathbf{p}}'_k A^{-1}\hat{\mathbf{X}}}{p'_3 \hat{\mathbf{X}}'} - \frac{\lambda \mathbf{x}'_k \hat{\mathbf{p}}'_3 A^{-1}\hat{\mathbf{X}}}{p'_3 \hat{\mathbf{X}}'} = d_E \tag{7.15}$$

This can be used with the error measure in equation 7.14, in the least-squares minimisation 7.3 to give a complete linear approximation. Generally, the addition of the inverse component provides only small improvements to the results. Sometimes, it can result in worse solutions and so, for best results, a solution both with and without the inverse should be found, and the best of the two selected using the full error measure.

Poorly Aligned Sequences

If the two sub-sequences being merged are very different, then it can often be the case that the overlapping camera matrices will not present a good means of aligning the two sequences. In these situations, it is generally better to use structure to estimate the 11 parameters of the projectivity instead of aligning the projection matrices.

To do this, an alternative method based on using the constraints provided by matching projection matrices (equation 7.4) can be used instead of using the direct alignment given by $P_i \simeq P'_i H^{-1}$. In this case, for only one overlapping image pair $P \leftrightarrow P'$, equation 7.4 reduces to:

$$\min_H \sum_j d^2\left(P'H^{-1}\mathbf{X}_i, P\mathbf{X}_i\right) + d^2\left(PH\mathbf{X}'_j, P'\mathbf{X}'_j\right) \tag{7.16}$$

for overlapping cameras P, P' and all structure i in sequence 1 and structure j in sequence 2. The advantage of this measure is that, because it considers only one overlapping camera P, it is invariant to affine transformations of the structure space $\mathcal{P}^3$. Consequently, it applies constraints to only the top 3x4 sub-matrix of the projectivity H. This means an H of the following form can be estimated:

$$H = \begin{bmatrix} H11 & H12 & H13 & H14 \\ H21 & H22 & H23 & H24 \\ H31 & H32 & H33 & H34 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

This form of estimation is in fact closely related to the resectioning algorithm (see section 5.3.1 on page 98). Just considering one part of equation 7.16, it can be seen that estimating H is exactly equivalent to re-estimating P_k as in resectioning. However, in this case, it is balanced by expecting the inverse to be true.

A linear algorithm can be derived, using the same method as for resectioning (i.e. substituting algebraic distance into equation 7.16). Since there are two possible linear algorithms, depending on which half of equation 7.16 is used, it is recommend to try both as well as

the direct registration method (i.e. $H = P^+P'$) and select the one minimising the full error measure.

Whilst this algorithm is fairly effective, it does present a problem. For the one view criterion in equation 7.9, the form of the projectivity was greatly simplified by aligning the overlapping camera matrices with the basis. If the first camera is not aligned with the standard basis, then the projectivity has to have the form in equation 7.7:

$$H(v) = P^+P' + \mathbf{h}\mathbf{a}^T$$

By altering the form of the aligning projectivity, so that the two cameras no longer register exactly, it is possible to move only one camera to the standard basis, leaving $H = [I|0]P' + (0,0,0,1)^T \mathbf{a}^T$. This means H does not have the same simple form as in equation 7.9, and so will not be linear in the unknowns if it is inverted for the minimisation using the distance criterion in equation 7.15. Hence any H that is part estimated in this manner cannot be used with the complete linear algorithm (only the part of 7.15 involving the non inverted H).

One final important note concerning this algorithm is that it is robust and does not rely on the accuracy of any structure matches that may or may not be available. Subsequently, the method can be run before any outlier identification. However, this algorithm is not strictly speaking necessary and rarely results in anything more than a 1% or 2% improvement overall (on the tests to be presented in the results section) unless very large errors are encountered (a few pixels), or there is a particularly inaccurately reconstructed image.

Practical Considerations

In practice, both the algorithms considered so far only give a close approximation to Euclidean structure and are not absolutely perfect. Because of this, it is recommended to try calculating A as normal, swapping P, P' and X, X' then calculating A^{-1}. The one that gives the best result according to equation 7.3 with Euclidean distance should then be accepted.

So far, only linear approximations to the merging error have been discussed. Although the results from these are excellent, it is often worthwhile refining the results from them using the full nonlinear measure. In this case, an iterative technique, such as Levenberg-Marquardt (see appendix A), can be used to minimise equation 7.3 with Euclidean distance for all the 15 parameters of H.

7.4.2 Merging with Other Degrees of Overlap

The one view algorithm just presented is without a doubt the most effective method. Indeed, if there is more than one view overlapping, the one view algorithm can still be applied, but in that case it should be tried individually on all the overlapping views and the best result selected. However, algorithms can still be developed to better handle either fewer or more overlapping camera views.

Zero Overlapping Views:

In this case, constraints are provided by equation 7.1 only, and it is only possible to use the error measure 7.3 alone (with a complete 15 parameter homography). If Euclidean distance is used, then both error measures are nonlinear and no direct solution can be obtained. Instead the algebraic distance measure can be used, and the inverse omitted, in order to obtain a linear approximation, before refinement with a nonlinear stage using the full Euclidean measure. It is worth noting that the zero view overlapping methods are applicable to any number of overlapping views provided structure matches can be found.

Two or More Overlapping Views

For the case of two overlapping views, the constraints offered by matching projection matrices (via equation 7.4) are sufficient to determine H completely. Recall from section 7.4.1 that with only one overlapping projection matrix, only 11 parameters can be determined. The two or more view projection matrix only algorithm tends to be less effective than the one view overlapping algorithms but has uses because it is robust to structure mismatches (a point that will be very relevant later in the chapter).

The constraints for this algorithm have already been provided in equation 7.4 and take the form:

$$\sum_{ki} d^2\left(P'_k H^{-1}\mathbf{X}_i, P_k\mathbf{X}_i\right) + \sum_{kj} d^2\left(P_k H\mathbf{X}'_j, P'_k\mathbf{X}'_j\right)$$

If algebraic distance is used with these equations, and the part of the equation involving H^{-1} is omitted, a linear algorithm can be used to determine H. Naturally, the process should be repeated after swapping the sub-sequences to get H^{-1} and the best solution kept. Alternatively one of the overlapping projection matrices can be aligned exactly (all can be tried and the best kept) as in the one view overlap algorithm and the aligning homography reduced to 4 unknown parameters. This means the same trick can be used to determine

accurate scale factors and include the inverse component, and can result in a very accurate algorithm.

7.4.3 Alternative Criteria for Merging Sub-Sequences

The methods presented above all relied on the two error measures presented in section 7.4 which rely on either camera or structure matches. Whilst these are the only two error measures that will be used in this work, both different measures and different variations on these two criteria are possible. In the interests of completeness some of these measures will now be reviewed and discussed here.

In [FZ98b, ZBR95], two other least-squares minimisation criteria were suggested for use in merging projective sub-sequences. The first of these is to minimise the squared distance between 3D points common to both sub-sequences:

$$\min_H \sum_j d^2\left(\mathbf{X}_j, H\mathbf{X}'_j\right) \tag{7.17}$$

The second criterion minimises the squared re-projection error to the original observed corners $\mathbf{x}^i_j$ from which the 3D points were triangulated:

$$\min_H \sum_{ij} d^2\left(P_i H\mathbf{X}'_j, \mathbf{x}^i_j\right) + d^2\left(P'_i H^{-1}\mathbf{X}_j, \mathbf{x}^i_j\right) \tag{7.18}$$

Since there is no concept of distance in projective space the first measure in equation 7.17 is only strictly meaningful if the 3D frame is a metric one. However, in the metric case it was pointed out in [FZ98b] that H is limited to being a similarity transform and can be solved in closed form using equation 7.17 with Euclidean distance. Since this is optimal, there is little need to further consider the least-squares registration problem in the calibrated case. Similarly, if the frame is quasi-Euclidean (a Euclidean approximation based on inaccurate calibration), this 3D measure can often give good results, but an algebraic distance measure still needs to be used.

For the fully projective case, it is a good idea to work with image quantities only and hence with the measure in equation 7.18. However, this particular criterion has drawbacks. Firstly, it suggests that re-projection errors should only be minimised in images common to both sub-sequences. And secondly, rather than minimise the difference between the two reconstructions, it minimises the difference between one reconstruction after transfer to the projective basis of the other reconstruction and the observed data.

The alternative criterion suggested in this work (equation 7.3) minimises the difference between the two reconstructions, leading to a two fold advantage. Firstly, both reconstructions have been estimated using all data in the relevant sub-sequence, and so their projections should be a more reliable estimate of the actual feature position than the observed feature. Secondly, because the same projection matrix is being used to project both the points being compared, they will be subject to the same uncertainty due to the projection matrix, simplifying the error model and making it less prone to large errors (it is still not a maximum likelihood estimator for the full problem though).

In addition, by not just using residuals for points that appear in overlapping images, the new measure has a further advantage in that it will favour the more reliable further tracked points. On the other hand, the symmetry of equation 7.18 is lost so that it is possible for one sub-sequence to contribute more residuals than the other. In practice a slight improvement can be made by weighting the two halves of equation 7.3 to make the contribution of residuals from both sub-sequences even (n_1 and n_2 in equation 7.3).

There are also other numerous minor variations on these and other error criteria (for example considering projection of matched structure in matched images). However, the author has experimented with some of these and found that there is little difference between them. It seems best to simply involve as many cameras and items of structure as possible rather than use elegant methods which require very specific data.

7.5 Robust Merging

It is reasonable to expect structure outlying to the sub-sequences used to initialise the merging process has been removed, and as such it would be expected that there would be very few outlying structure matches between the sub-sequences. Indeed, if the initial sub-sequences are longer than two images and sub-sequences share common images, it is often quite possible to obtain a reasonable reconstruction using normal least-squares techniques alone. However, this is by no means a guarantee against outliers and they can still occur, even with points that have been tracked for many images.

For example, given two points that have been tracked for 3 images each, and which share one common point, it is quite possible that image effects such as shadow or occlusion have caused one of the points to be matched incorrectly and then tracked anyway in a manner consistent with the geometry. However, when the points are joined they will become outlying.

Although rarely significant, except in specialised cases, outliers can sometimes have a

large effect on the result. Consequently, to ensure a viable solution it is recommended to use a robust method, then detect and remove outliers. At the heart of any robust method is a different, more robust function of the residuals than that used in normal least-squares methods. As such, it will be appropriate to first describe the function used for the purposes of this work.

7.5.1 A Robust Error Criterion

Standard least-squares methods attempt to minimise $\sum_i r_i^2$, where the residual r_i can be defined as the difference between the ith observation and its fitted value. When there is the possibility of outliers in the data, a different function $\rho(x)$ that is robust to outliers should be minimised instead, meaning the robust function to be minimised can be written as:

$$\sum_i \rho\left(\frac{r_i}{\sigma}\right)$$

For this case, r_i can be taken from equation 7.3 and the standard deviation σ can be replaced with the robust standard deviation, defined as:

$$\sigma = 1.4826\left[1 + \frac{5}{n-p}\right] \underset{i}{\text{median}} |r_i|$$

for n observations and a parameter space of dimension p (see [RL87] for full details). For the case in hand, a robust Huber function [Hub81] is suggested for $\rho(x)$:

$$\rho(x) = \begin{cases} x^2 & x < \tau \\ \tau^2 & x \geq \tau \end{cases}$$

See appendix C for more details of this. The constant τ should be selected based on some confidence limit. If an algebraic approximation to distance is being used for r_i, then $\rho\left(\frac{r_i}{\sigma}\right)$ will conform to a χ^2 distribution with one degree of freedom and so the value $\sqrt{3.84}$ can be used for a 95% confidence level. If, on the other hand, Euclidean distance is being used, then $\sqrt{5.99}$ can be used for the same 95% confidence level.

The big advantage of this Huber cost function is that it conducts the minimisation over all structure whether or not it is outlying. By applying a fixed cost for all outliers, it is assumed outliers are drawn from a uniform distribution. Another big advantage of this is that points that are poorly localised but are not outlying still contribute to the result (such points can become inlying after re-calculation of structure).

Whilst minimisation of this robust Huber function utilising r_i from 7.3 will work, much better results can be obtained if it is realised that both the different sub-sequences will have different errors in their reconstruction and hence different standard deviations. Consequently, equation 7.3 should be split into two parts:

$$\sum_{ij} \frac{1}{n_1} d^2 \left(P_i H \mathbf{X}'_j, P_i \mathbf{X}_j \right)$$
$$\sum_{kj} \frac{1}{n_2} d^2 \left(P'_k H^{-1} \mathbf{X}_j, P'_k \mathbf{X}'_j \right)$$

each of which should be evaluated separately, using a different robust standard deviation for each part. This is very important because otherwise, if one sub-sequence has been reconstructed with a much lower standard deviation, it will cause all residuals in the other sub-sequence to be flagged as outliers.

Now that the robust function has been presented, it will be possible to address how that function may be minimised. One approach would be to use an iterative method such as Levenberg-Marquardt (see appendix A) to minimise the function directly. This can be effective, but it still requires that an initial solution be found.

Recently, random sampling techniques have proved highly successful at minimising these sorts of functions (see work relating to MLESAC, [TZ00]). However, because of the sparsity and lack of effect of outliers, it is proposed that in many cases, particularly where computational efficiency is an issue, it would be more appropriate to use standard M-estimator techniques and remove outliers based on the results this produces.

7.5.2 M-Estimators

An M-Estimator attempts to minimise the nonlinear Huber function directly. Since, in this case, a good estimate of the parameters can be obtained using normal least-squares even with outliers, a good approximation of this minimisation can be found by recasting it as an iterative re-weighted least-squares problem (see [Zha97] for details).

Since the Huber cost function relies on an estimate of the standard deviation, a least-squares method is run repeatedly, and the results from the previous run used to determine the robust standard deviation and hence weightings for the next run. If a residual r_i is found that is greater than τ, it is weighted by $\frac{\tau}{r_i}$ otherwise it is not weighted at all.

This solving and re-weighting is repeated until there is no significant change in the sum of the Huber function for all r_i. Note that, if the data set contains outliers with very large

residuals (not likely when merging), then the initialisation may fail and the solution vary wildly. This needs to be detected and handled by using a different technique (such as random sampling).

7.5.3 Random Sampling Methods

The effectiveness of M-estimators for the case in hand does not entirely preclude the need for random sampling methods for some specialised cases, in particular the case of zero views overlapping or poor quality point tracking (for example merging sub-sequences with only two images in each). In these situations, outliers can be very significant and a random sampling technique will be most effective.

The basis of random sampling is to pick random sub-samples of the data set and estimate the model parameters using those samples of data only. The best of these estimates is then determined based on the error measure being minimised. When using a random sampling method, it is therefore important that the minimum number of points are used to estimate the model parameters so as to reduce the probability of an outlier being included in the random sample. This necessitates minimal algorithms for differing degrees of image overlap.

Zero View Overlap

In this case, the only constraints available are those offered by corresponding 3D structure. Since a general projectivity has 15 degrees of freedom, it follows that at least 5 points are needed to compute the projectivity. 5 points in fact form a projective basis of $\mathcal{P}^3$, and so it is appropriate to reformulate the problem as a change of projective basis. If 5 3D points $\mathbf{X}_n$ are selected in the first sub-sequence, they are easily made into the standard projective basis by a transformation B of the form:

$$B = \begin{bmatrix} \lambda_1\mathbf{X}_1 & \lambda_2\mathbf{X}_2 & \lambda_3\mathbf{X}_3 & \lambda_4\mathbf{X}_4 \end{bmatrix}$$

where

$$\begin{bmatrix} \lambda_1 & \lambda_2 & \lambda_3 & \lambda_4 \end{bmatrix} = \begin{bmatrix} \mathbf{X}_1 & \mathbf{X}_2 & \mathbf{X}_3 & \mathbf{X}_4 \end{bmatrix}^{-1} \mathbf{X}_5$$

A similar transformation B' can be found to move the same 5 points in the other sub-sequence to the standard projective basis. Since this aligns both sub-sequences in the same projective basis, the projectivity H in equations 7.1 and 7.2 can be found as $H = B'^{-1}B$.

The results from using this algorithm are, naturally, heavily dependent on the accuracy of the 5 points selected, and can often produce useless results when some of the 5 selected

points are poorly localised inliers. This means it is necessary to use at least somewhere in the region of five hundred random sub-samples to ensure a sufficiently accurate result.

One View Overlap

In the case of only one overlapping view, the minimal algorithm is simply that given in section 7.4.1 which requires four projections of four, preferably different 3D points. An advantage of this algorithm is that it is minimal for calculation of triplet geometry, yet requires only four unknown parameters be determined. Recall that, in the case of a triplet of images, there are 18 unknown parameters describing the trifocal tensor. However, for the merging case, 14 of these have been eliminated by making use of the fundamental matrices to produce reconstructions for the two image pairs. This confers definite performance advantages if the two fundamental matrices are already available.

So, how should the points be selected? Although only four projections are required for the minimal algorithm, it is not usually a good idea to give it only the minimal data. Instead, it is recommended to select four pairs of matching 3D points and use all projections of these points. To select the structure, selection is performed from the set of all projections. The advantage of this is that it increases the chance of a point tracked for many images being selected. This is not a bad thing because such points are very likely to be reliable because they have been tracked so far.

7.5.4 Two or More Overlapping Images: Robust Error Criteria

Whilst robust methods are effective, it would be better still if some normal least-squares criterion could be used that is not affected by potentially outlying matches between structure. To do this, constraints can be imposed by using the relationship between different projection matrices for the same image (as in equation 7.2) rather than by using matches between structure (equation 7.1). Each overlapping image can impose 11 constraints in this manner, and so at least two overlapping images are required to determine H completely.

One suitable technique along these lines has already been mentioned in section 7.4.2. For this method, the error criteria based on matches of projection matrics is used to produce a method insensitive to mismatched structure. See section 7.4.2 for details of the method.

7.6 Merging Two Sub-Sequences

After robustly calculating homographies, it becomes possible to merge the two sub-sequences. This first requires outlier removal, followed by re-estimation of the merging projectivity using the least-squares methods before finally the sub-sequences can be merged into the same projective basis.

7.6.1 Removing Outliers

Once a merging projectivity has been robustly estimated, it is necessary to remove outliers before the more effective least-squares methods can be used. Outliers are removed on a per-projection basis, and are determined by using Euclidean distance in the cost function given in equation 7.3 for every ij and ik. Note that this does not require any re-calculation of structure and so is very fast.

It is important to note, that as discussed back in section 7.5.1, the merging results for projecting one sequence into the other are often significantly different from the converse results. This is particularly prevalent if one sub-sequence is reconstructed far more accurately then the other. Consequently, to prevent favouring one sequence (sometimes disastrously), the outlier rejection should be run twice, once for the projection of each sub-sequence into the other, i.e. equation 7.3 should be split into two:

$$\sum_{ij} d^2\left(P_i H \mathbf{X}'_j, P_i \mathbf{X}_j\right)$$

$$\sum_{kj} d^2\left(P'_k H^{-1} \mathbf{X}_j, P'_k \mathbf{X}_j\right)$$

Similarly, it is important to note that no residuals for feature matches in overlapping images should be included in robust outlier rejection. This is because such points are common to both items of structure and hence definitely outlying to neither.

Under the assumption that structure with only two projections is unreliably matched, if either of the two points being merged is reduced to having less than two projections it should be removed totally. Similarly, if after outlier rejection, the resulting merged point is reduced to having less than 3 projections, both points should be removed. This final check is only necessary if more than one image is overlapping, because this allows two items of structure to be reduced to having exactly the same pair of projections, e.g. Given matches u,u',u" in sequence 1 and u',u" in sequence 2 then if u is identified as an outlier only a pair of points will be left.

Noting that the Euclidean distance is approximated by a χ^2 variable with 2 degrees of freedom, this means that a projection of a point is flagged as an outlier if it has a squared re-projection error greater than $5.99\,(\sigma)^2$. Here, 5.99 again corresponds to the 95% confidence level and σ to the robust standard deviation (calculated seperately for each sub-sequence).

For Small Sub-Sequences

When merging one or more short sub-sequences such as image pairs then per-projection outlier rejection is usually highly inaccurate. This is mainly because of the inaccuracy of point reconstruction from only two images. Instead of rejecting on a per-projection basis, the two sub-sequences are merged using the technique in section 7.6.2, all structure is re-calculated using all projections and then outlier rejection is performed based on the average re-projection error for each item of structure against observed features. Matches are then rejected if the merged structure projects outside a 95% confidence limit i.e.:

$$\sum_i d_E^2\left(P_i\mathbf{X}, \mathbf{x}_i\right) > 5.99\,(\sigma)^2$$

where σ is the robust standard deviation for the above re-projection error and $\mathbf{x}_i$ is the observed projection of merged feature $\mathbf{X}$ in image i. If an outlier is found, the item of structure with least projections that formed the match is rejected outright. If either item of structure in an outlying pair has less than 3 projections before merging, then it is also rejected outright. This method of outlier rejection is much more effective for merging image pairs or triplets.

A further consideration which must be observed when dealing with merging involving one or more sub-sequences with two images, is that due to the sometimes large number of outliers the two image reconstructions can sometimes be very poorly aligned. Subsequently, after outlier removal, any image sub-sequences should be re-constructed without the outliers.

7.6.2 Merging the Sub-Sequences

After the projectivity that can be used to merge two sub-sequences has been robustly computed, and outliers removed, the actual process of merging can proceed. To perform this, all structure and cameras from one sub-sequence is appropriately transformed using equations 7.1 and 7.2 and then placed into the other sub-sequence. This presents a problem because, in overlapping images there will be more than one potential projection matrix, and for structure that is common to both sub-sequences there will be more than one 3D reconstruction.

To deal with this, all structure that is present in both sub-sequences is assigned the 3D reconstruction that produces the lowest error (Euclidean re-projection squared) for all the structures projections in both sub-sequences. For projection into common images, the error for both possible projection matrices is included. After all structure has been merged, the best projection matrix can be selected for each overlapping image, again based on squared Euclidean re-projection error for points in that image.

If speed is not an issue further improvements can be made by re-calculating any merged 3D structure and seeing if it produces an improved result. Similarly, cameras in the overlapping images can be re-calculated or nonlinear refined using re-sectioning. Alternatively a Kalman filter can be used to update the reconstruction, and a final bundle adjustment omitted totally.

A final point particularly worth noting is that points which have been tracked for only 2 images should be considered unreliable and as soon as it has been identified they will not be tracked by a merge they should be removed. It is important to note that they should only be removed from the merged sequence, and if the sub-sequences are to be re-used in any future merges then the two image tracks should be included for that merge.

Since there generally tends to be a large number of points tracked for only 2 images, if they are included in processes such as bundle adjustment they can really slow things down and throw the accuracy so this trimming of points is highly recommended.

7.7 Merging Algorithm Summary

For clarity, the complete merging algorithm will now be outlined. Note that this is independent of the reconstruction algorithm being used, and is applicable to any of the algorithms outlined in section 7.3.

1. Identify all structure common to both sub-sequences.

2. Remove any structure that has only two projections and is not tracked between the sub-sequences.

3. If more than one image is overlapping, discard any structure from both sequences if they track inconsistently, e.g. the structure from sequence 1 projects to u,u' in the overlapping images and the structure from sequence 2 to v,u'.

4. Robustly calculate the merging projectivity using any of the methods in section 7.5. A robust estimate can be obtained using random sampling followed by a nonlinear minimisation of a robust Huber function. For two or more views overlapping, the robust criterion can be used instead, and random sampling avoided if desired.

5. Remove outliers from the set of common structure using the techniques in section 7.6.1.

6. If any sub-sequences are only two images in size, they should be re-estimated using only the remaining inliers.

7. (Optional) Re-estimate the merging projectivity using a linear least-squares method, followed by an optional nonlinear refinement. See section 7.4 for more details. Because of it's accuracy, the one view overlapping method utilising the inverse (see section 7.4.1, page 128) almost invariably produces significantly better results than any robust method for any number of overlapping views (even if a nonlinear refinement is performed on the Huber function).

8. Merge the sub-sequences using the technique in section 7.4.

9. Re-calculate or refine all structure that was matched, and all overlapping projection matrices.

10. (Optional) bundle adjust the resulting sequence.

7.8 Results

In order to evaluate the different solutions to the merging problem presented in this chapter, a number of experimental results have been obtained for both real and synthetic data. These have been split into two different sections. The first of these sections will attempt to determine the effectiveness of the alternative approaches to merging, and the second will compare the merging approach to other projective reconstruction methods.

7.8.1 Synthetic Data

The method used to generate synthetic data for all the tests in this chapter is basically the same, and so it shall be described first. For synthetic simulation, the setup was kept as close as possible to real life, with a scene consisting of a set of 200 ± 50 3D points $\mathbf{X}_i$ scattered

randomly in a cube with edges of size 2000 units. Initially, the camera is positioned 2500 units away from the centre of the cube and is given a focal length of 600 plus a uniform random number of ± 200. Skew, aspect ratio and principal point are set to uniform random values between ± 0.1, ± 1.0 and ± 10 off image centre respectively. These intrinsic parameters then stay constant throughout the whole sequence.

To create a sequence, additional cameras P_j are added by perturbing a camera trajectory. Initially, the camera is placed at the centre of the coordinate system and given no movement in a random direction. For each new image, a small uniformly random rotation of $\pm \frac{\pi}{20}$ radians and uniformly random translation of ± 100 units is added to the current movement direction and magnitude, which is then used to displace the previous camera position. This provides a more natural model of movement than simply applying random displacements to the previous camera, an important consideration when dealing with long sequences.

When the cameras have been determined, all the structure is projected into 800×600 images, and random Gaussian noise with standard deviation σ is added to the image points. If there are too few points in any image, then the sequence is discarded and another created. Given a viable sequence, it can then be split into suitably overlapping image sub-sequences and all points that are not visible in at least 3 images are removed. To model matching failure, a random percentage between 0% to 15% of projected points are removed from each image in the sub-sequence. Note that this modelling of matching failure is not used when factorisation methods are being considered because factorisation methods cannot effectively handle the missing data (except at a cost to accuracy).

If outliers are to be added, then a final stage will randomly select the relevant percentage of points and add a uniform distributed amount of noise to each of those points. This noise is at most the size of the image, but if it results in a point within the 95% confidence limit of the added noise or a point outside the image then the outlier offset is re-estimated until it becomes suitable.

7.9 Results for Merging Reconstruction

This section will aim to study the merging approach to reconstruction and compare the different merging schemes and algorithms. In more detail, the two main points to be addressed are:

No. Iterations	Average Error
350	0.594213
500	0.582406
800	0.568105
1000	0.559434
1500	0.54377

Table 7.2: Performance of the six point random sampling algorithm as number of samples is varied. All results are for noise of standard deviation 0.4 pixels added to the data.

- Results will be given to indicate whether or not it is desirable to start the merging process from image pairs or image triplets. This will be achieved by comparing robust merging reconstruction for a triplet with other robust triplet reconstruction techniques.
- Next, an attempt will be made to compare the different merging schemes. This will be done using both real and synthetic data. It will primarily address the effect of numbers of overlapping images, and which merging algorithm to use.

7.9.1 Merging Pairs to Create Triplets

One approach to producing a reconstruction from a triplet of images is to use the relevant multilinear form - the trifocal tensor. Methods for robust reconstruction using the trifocal tensor are well established, and have been shown through experience to be very effective and accurate (see [BTZ96]). As an alternative, the merging approach can produce a reconstruction for an image triplet by merging reconstructions of two image pairs that overlap by one image.

If this transpires to be as effective as the trifocal tensor approach, then it means that merging based reconstruction can be initialised with pairs rather than triplets and some complexity avoided. It is especially worth noting that computation of the trifocal tensor can be very complicated, particularly when producing a maximum likelihood estimate (see for example [PF98]) and so it would be beneficial to be able to avoid it.

Robust Triplet Reconstruction

To robustly reconstruct an image triplet, it is possible to use either the six point algorithm to calculate the trifocal tensor (see appendix D), or alternatively to reconstruct both the

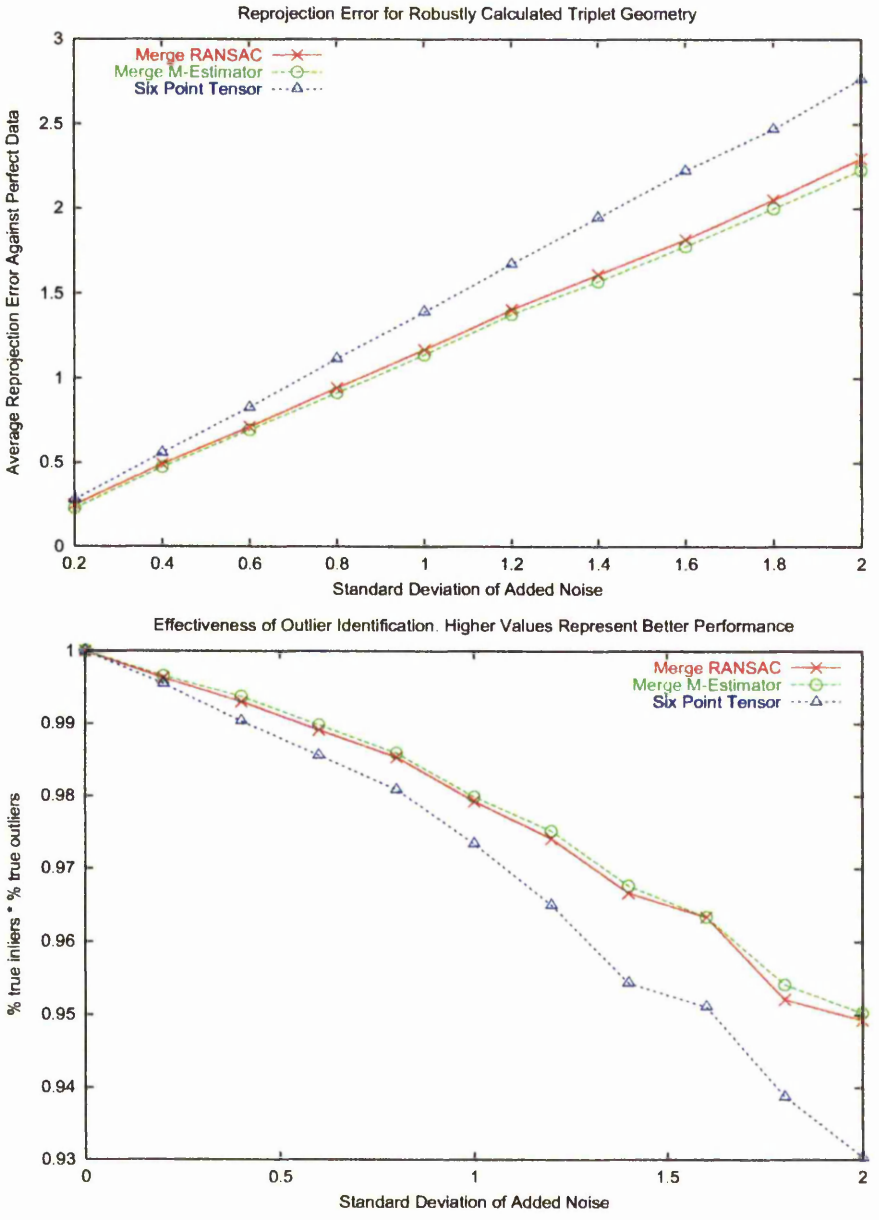

Figure 7.5: Comparison of robust image triplet reconstruction algorithms for up to 20% points as outliers

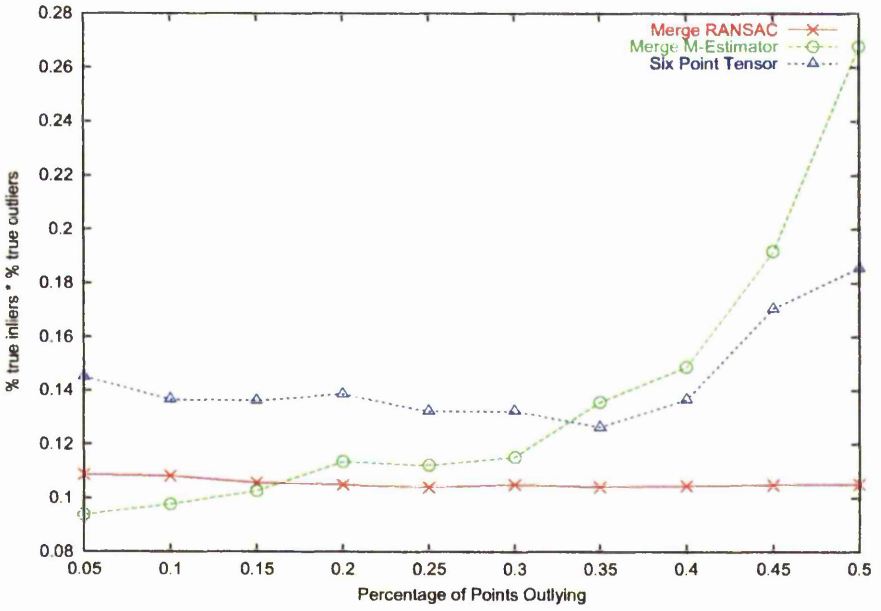

Figure 7.6: Comparison of robust image triplet reconstruction algorithms for varying proportion of outliers

consecutive image pairs in the triplet and then merge them. This allows either the four point algorithm given in section 7.5.3, or the M-estimator algorithm given in section 7.5.2, to be used.

The random sampling four point merging algorithm already has a clear computational advantage over the six point algorithm since it requires fewer sub-samples to gain the same probability of selecting in-lying points. Similarly, the M-estimator does not sample and so is far more efficient. Furthermore, both merging algorithms come complete with 3D structure and camera matrices, allowing very efficient measurements of error. This is unlike the six point algorithm which requires either some form of structure to be computed or use of an inaccurate transfer error (although a good and fast to compute first order approximation to the ideal error is possible).

It remains to establish that the new robust algorithms are as effective. To do this, synthetic sequences of 3 images were generated, but modified to contain a random proportion of up to 20% outliers. An outlier in this case, is defined as any point outside the 95% confidence limit on the normal distribution of the outlier free data, that is still within the 800×600 image region.

All the different approaches to robust triplet reconstruction were then applied. For clarity, the exact method used for both approaches will now be detailed. In the case of the six point robust algorithm this is:

1. Take 1000 random samples of 6 points

2. For each sample produce an estimate of the trifocal tensor using the six point algorithm (see appendix D). This produce one or three solutions for each sample.

3. For each estimate of the trifocal tensor, evaluate a Huber function based on a 95% confidence limit ($5.99 * \sigma$ where σ is the robust standard deviation) and re-projection error. In this case re-projection error means each point is reconstructed using the estimates of the cameras and then the sum of it's re-projection error taken in all three images.

4. The estimate associated with the smallest Huber function result is taken as the solution and decomposed into three projection matrices.

5. Outliers are flagged as those points outside the 95% confidence limit used in the minimum Huber function.

For the four point merging method, the following algorithm is used:

1. For each of the two sub-sequences composed of images 1,2 and images 2,3 robustly generate a fundamental matrix F_{12} and F_{23} :
   (a) Take 600 random samples of 7 points.
   (b) For each random sample estimate a fundamental matrix using the seven point algorithm.
   (c) For each estimate of the fundamental matrix evaluate a Huber function based on a 95% confidence limit and transfer error using the fundamental matrix.
   (d) Select the result with the lowest Huber function as the estimate of the fundamental matrix.
2. Produce a reconstruction from each of the fundamental matrices using factorisation to produce the projection matrices and Hartley-Sturm correction to produce 3D points (see chapter 5).
3. Take 200 random samples of 4 points.
4. For each random sample estimate the complete merging projectivity using the one view overlap algorithm.
5. For each estimate of the merging projectivity evaluate a robust Huber function based on a 95% confidence limit and re-projection error as stated in section 7.5.1.
6. Select the lowest result as the estimate of the merging homography.
7. Merge the two reconstructions using the selected result to produce a set of three consistent projection matrices.
8. Reject outliers using the method of section 7.6.1.

Note that for the merging algorithms a robust reconstruction was produced for both pairs of consecutive images, but no outliers were rejected until after the merge. This is important to ensure that the quality of outlier rejection for the triplet is not affected by the quality of outlier rejection for the image pairs.

The quality of the results that were produced were finally evaluated using two error measures. For the first error measure, projection matrices and structure were calculated after

outlier removal, and the average Euclidean image distance between re-projected points and the perfect noise free points was taken. Naturally, this was only performed for those points that were actually non-outlying. It is fairly inevitable that, the closer the reconstruction is to being perfect, the more accurate outlier detection will be.

In order to measure the effectiveness of the outlier rejection process, a second error measure was also used. This plots the number of true positive outliers multiplied with true negative outliers, i.e. the number of identified inliers that were actually inliers multiplied with the number of identified outliers that were actually outliers. This is similar to the type of score that might be plotted on an ROC curve. However, an ROC curve is not appropriate to the case in hand because an ROC curve measures the effectiveness of a classifier as the classification boundary changes, whereas this experiment measures the effectiveness of a classifier with the same classification boundary, but varying model parameters.

Results for these measures as the amount of noise σ added to the images is varied, are shown in figure 7.5. They indicate that, for any amount of noise, the new algorithms are superior (varying from 10% to 20% improvement). It is hard to know exactly why this is and would prove an interesting problem to pursue.

One possible reason for the improved results is that the merging approach actually uses up to 18 different points to estimate the complete triplet geometry as opposed to the six point algorithms which only uses 6 different points (and 3 projections of each point). It is well known that using more noisy points to estimate a mean value produces a better estimate of the mean. This has been demonstrated in the context of random sampling in [LPT00]. In theory this could be tested by creating an 18 point algorithm for estimating triplet geometry. However, such an algorithm would not be workable if lots of outliers were present because the chance of picking up an outlier in a random sample would be very large. The merging algorithm circumvents this problem by breaking the 18 point sample into smaller bits.

Another possible reason for the improved results is the error measure. In the merging approach structure has been estimated very accurately using the underlying pairs and then projected into the third image. This is very similar to the transfer error often used with the trifocal tensor, but in this case, the points are estimated using the already accurately determined fundamental matrices and then projected using the potentially inaccurate aligning homography.

However, these two reasonable suggestions have not taken into account the possibility of experimental error, particularly associated with errors in my versions of the code. To eliminate further possible sources of error, a further set of runs was performed for the case

of 0.4 standard deviation noise added to the points. In this case the triplet algorithm was run with successively more random samples to illustrate the adding more random samples to the trifocal tensor routine will not improve it's performance significantly. The results can be seen in table 7.2, and proves that the number of samples taken for the six point algorithm (1000) has not biased the results toward the merging algorithm since using more samples would only result in a negligible improvement.

A similar trend is borne out in graph 7.6 where, instead of varying noise, the percentage of outliers is varied. This graph illustrates the important fact that when, there are a lot of outliers, M-estimators are not to be recommended over random sampling (about 18% in this case). It is important to note that this graph only illustrates the effect of % of outliers. The point at which M-estimators initialised by least-squares fails is not only defined by the number of outliers, but also by the magnitude of the residuals produced by the outliers (assuming an unweighted initialisation). For example, just one outlier that is producing residuals well out on the tail of the distribution could make it fail. Bearing this in mind, M-estimators are not always to be relied on.

Least-Squares Triplet Reconstruction

The lack of need for 3D structure has long been seen as one of the advantages of using the trifocal tensor for reconstruction purposes, but it is proposed here that although mathematically elegant this does not always translate to accuracy or performance gains. This seems to be the case, because transfer of points with the tensor is fairly inaccurate. On the other hand, the high quality of reconstruction from image pairs (see section 3.6.3, page 67) and Hartley-Sturm reconstruction from image pairs (see section 5.4.3, page 102), means the presence of the redundant information inherent in projection matrices and 3D structure is *not* a problem for the two image reconstruction case. Indeed, it is actually an advantage because it allows easy and efficient use of meaningful error measures (and more importantly maximum likelihood estimators).

To support this (it is essentially a pragmatic proposition), a set of synthetic tests has been carried out. The synthetic system was used to generate image triplets with varying amounts of added noise σ. Each of a number of triplet reconstruction methods was used to produce a reconstruction, and the image error between the projections of the reconstructed model and the projections of the perfect model were taken. For comparative purposes, triplet reconstruction was tested with 4 different algorithms and run 4000 times for each value of

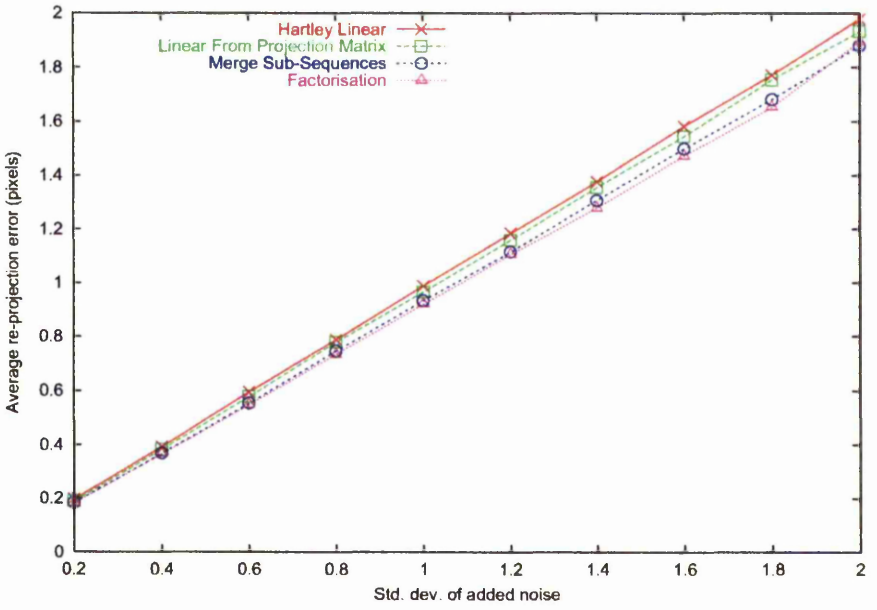

Figure 7.7: Comparison of different methods for generating a projective reconstruction of an image triplet

σ.

The first algorithm, labelled Hartley Linear is the linear method that uses the tensor trilinearities directly, as described in [Har95b, Har97]. The second method, Linear from Projection Matrices, performs a reconstruction from the fundamental matrix of the first two images in the triplets and uses the subsequent 3D to 2D relationship to calculate the third camera matrix using resectioning (see section 5.3.1, page 98). Structure is then recalculated using all 3 images and the third camera matrix re-estimated.

Of the final two methods, Merge Sequences is the new merging sub-sequence method presented in section 7.4.1. To initialise this, reconstructions were calculated from two image pairs and merged using all 1 view overlap algorithms. In order to avoid biasing the results against the purely linear method, the fundamental matrices were *not* nonlinear refined. Finally, the factorisation method is the one based on the closure constraints and Hartley-Sturm correction, and selects the best result from either the subspace or non-subspace methods for each test. See section 6.3.2 for further details on the factorisation methods. Note that for all methods, normalisation on the 2D points was used (as given in section 4.2.4, page 77), and no nonlinear refinement stages were used anywhere.

As can be seen from the results in graph 7.7, there is little difference between the algorithms. The least effective is the linear tensor method (Hartley Linear), which is heavily over parameterised and minimises a meaningless error measure. However, it should not be disregarded because it does provide an integrated means of including constraints from lines in the minimisation. Most effective is the factorisation method which is only slightly better than the merging approach (less than 0.05 pixels), which is slightly better than the linear from projection matrices method. Overall, the methods based on image merging are just as good as any other method and so can safely be recommended for general use in place of any existing triplet reconstruction methods.

7.9.2 Different Merging Algorithms

The next major question to be addressed, is to determine which of the numerous merging algorithms is most effective. To test this, random sequences randomly varying in length from 3 to 31 images were generated and split into two equally sized (or as close as possible) and appropriately overlapping image sequences. A projective reconstruction was then estimated for each of the two sub-sequences to be merged, by using the hierarchical reconstruction without bundle adjustment. Merging then proceeded from the two reconstructed sub-sequences,

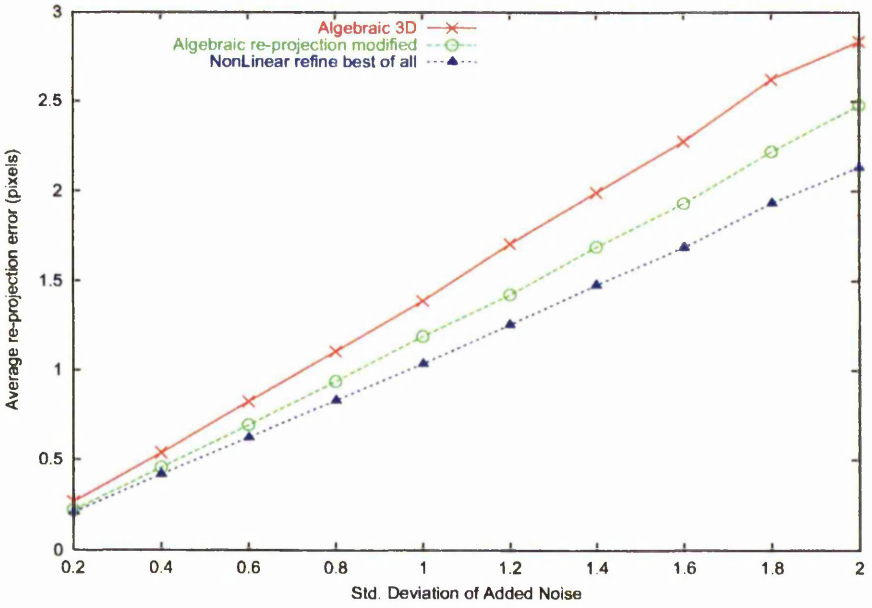

Figure 7.8: Comparison of different 1 view merging algorithms

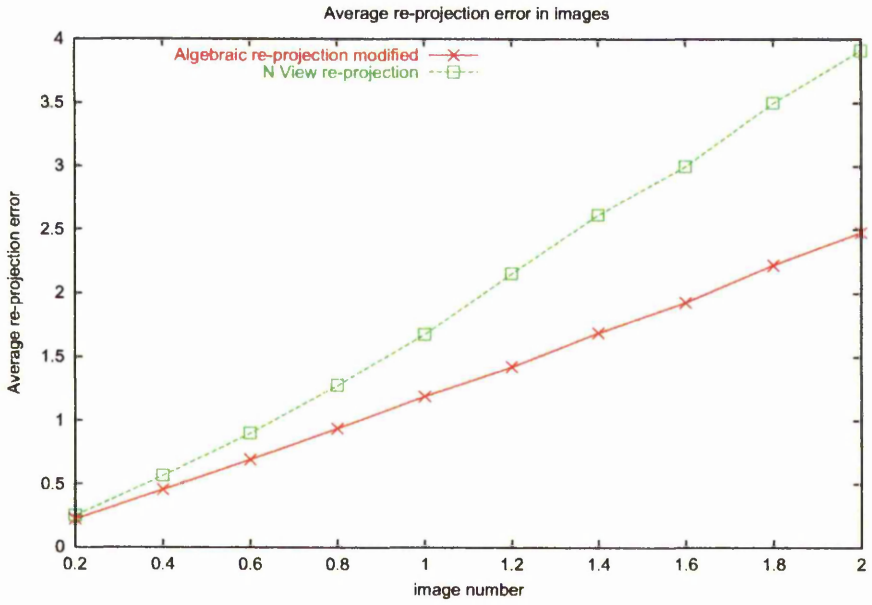

Figure 7.9: Comparison of 1 view and N view merging algorithms

using each of the merging algorithms individually to create cameras and structure for the whole sequence. The results of this final merging were then evaluated by taking the average Euclidean distance between each reprojected point and the associated projection from the perfect generated sequence. Finally, in order to prevent degeneracies causing huge upsets, an upper limit of 5σ was placed on the average error.

Figure 7.8 gives the results of this process for comparing the different algorithms. As can be seen the new 1 view algorithm performs notably better that the algorithm based on 3D error, regardless of the number of overlapping images (on average 10% better than the 3D error). However, it is far from perfect because the nonlinear refinement of the new algorithm adds some notable improvement. This is because the nonlinear method optimises all 15 parameters of a general projectivity wheras the linear method optimises only 4. It was found that if the nonlinear method only optimised the same 4 parameters as the linear method, there was neglible improvement. Although these may not seem like very significant improvements in themselves, the cumulative effect when merging hierarchically can be very significant, especially when reconstructing very long sequences.

The next figure, 7.9 illustrates the effectiveness of the algorithm which assumes no overlap (applicable to N views overlapping). It performs fairly well at low error, but is basically nowhere near as effective as the one view overlap algorithm. However, this does not mean that the algorithm is useless, because unlike the one view algorithm it involves estimating a complete projectivity rather than just 4 parameters. It follows that if the two sub-sequences being merged have reconstructed in significantly different ways, the one view algorithms will not perform well. As such, it can be concluded that it is best to include an estimate using the N view algorithm and compare it against the 1 view overlap algorithms.

7.9.3 One and Two View Overlap Comparison

The final question to be addressed is whether or not it is a good idea to use one or two overlapping views when merging. To do this, synthetic sequences were generated with varying amounts of noise and complete reconstructions produced using the hierarchical method starting from image pairs with one view overlapping and without ever using bundle adjustment. No outliers were included in this process.

The graph in figure 7.10 shows the results of this comparison. It illustrates that, having two views overlapping results in significant improvements to reconstruction quality. It seems likely that this is caused by the extra overlapping images providing more chance of a better

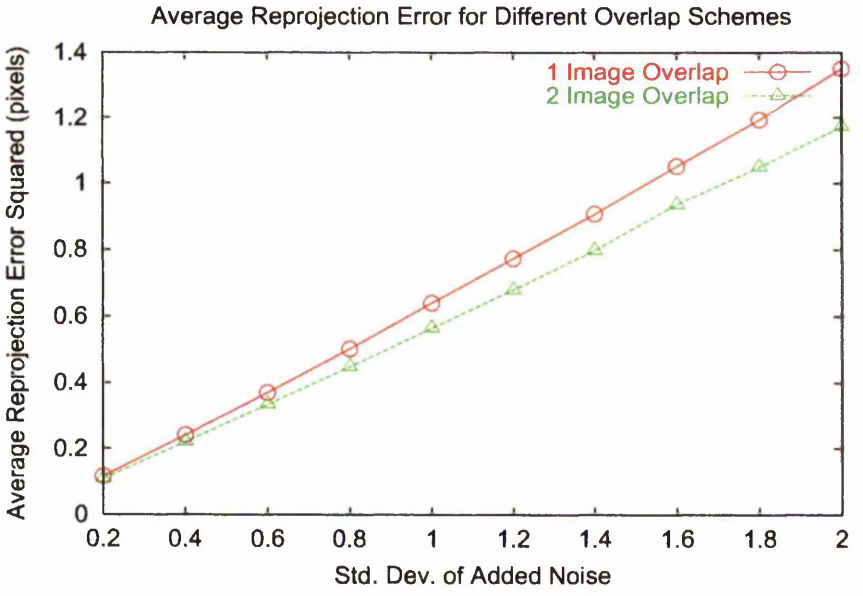

Figure 7.10: Comparison of Merging with Differing Numbers of Overlapping Images

registration for the projection matrices. It does, however, slow the whole process down, because it requires more merges.

However, this test has not considered the problem of robust reconstruction. From a robust point of view, two views overlapping is highly recommended, because of the availability of robust linear least-squares merging criteria and the possibility of outlier detection due to inconsistent tracks in the overlapping images. Because of this, two view overlap is to be highly recommended except where speed is a critical factor.

7.10 Comparison with Existing Projective Reconstruction Methods

7.10.1 Synthetic Data

In this case, exactly the same synthetic runs were performed as for testing one and two view overlap, but this time also including a sequential reconstruction algorithm as described in section 6.2 on page 109 and a factorisation based algorithm. The factorisation method is

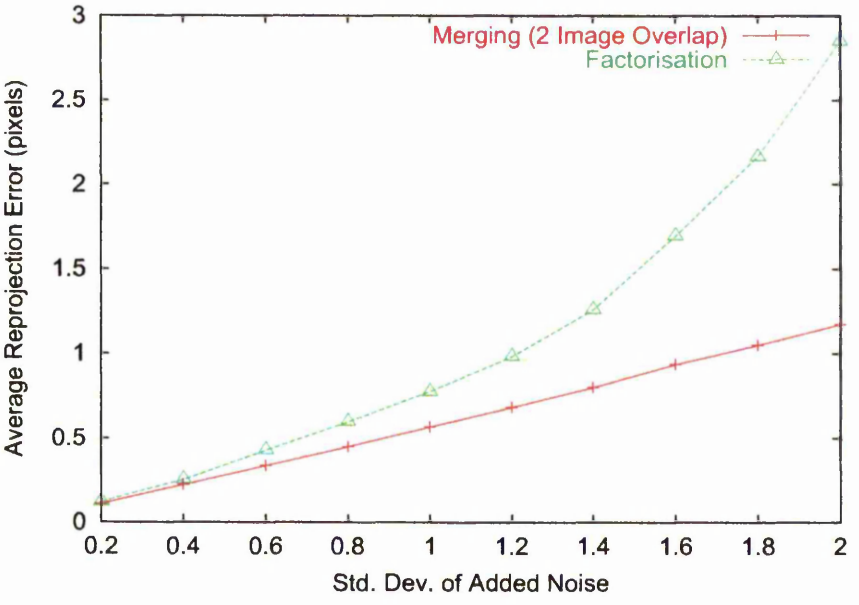

Figure 7.11: Comparison of merging and factorisation methods

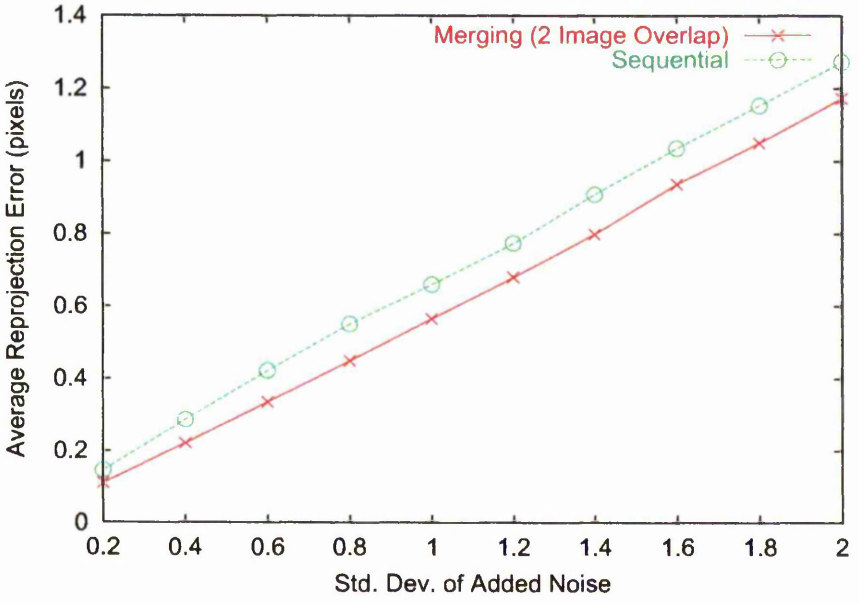

Figure 7.12: Comparison of merging and sequential methods

Figure 7.13: Images 0, 4, 8, 12 and 16 from the cluttered sequence of 17 images

the one based on the closure constraints, and selects the best result from either the subspace or non-subspace methods for each test (as described in section 6.3.2 on page 112). All algorithms were started from image pairs, including factorisation which only used the two image closure constraints and not the three image ones.

Graphs 7.11 and 7.12 shows that the new approach to projective reconstruction provides significant improvements. Of particular note is that factorisation algorithms perform very badly, even with the closure constraints and sub space methods. The new merging based algorithm on the whole performs the best. In effect, it was found that, if a bundle adjustment was run on the new algorithm's results, it produced very little improvement. In fact, the graph representing this has not been included because the average improvement is so small.

Real Data

Tests were also performed with real data. An automatic feature tracker was first used to obtain feature correspondences across some video sequences. A projective reconstruction was then found, starting from image pairs and using the different hierarchical methods with 1 view overlap and a conventional sequential method. Note that no bundle adjustment was used at any stage.

Figure 7.13 shows some images from a sample sequence of 17 images. Graph 7.14 shows

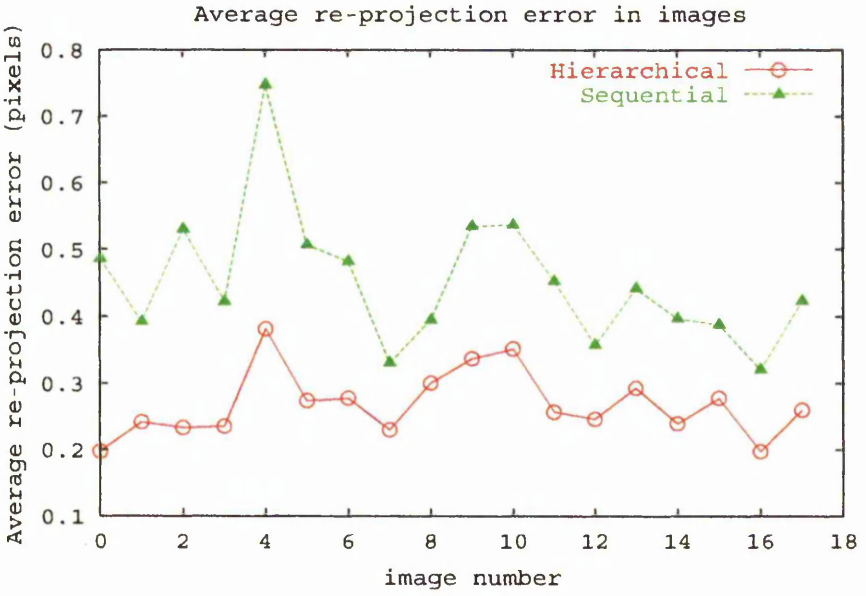

Figure 7.14: Re-projection error for points in all images of the cluttered sequence

Figure 7.15: Images 1,20,40,60 from the table sequence of 70 selected images (originally 280 images).

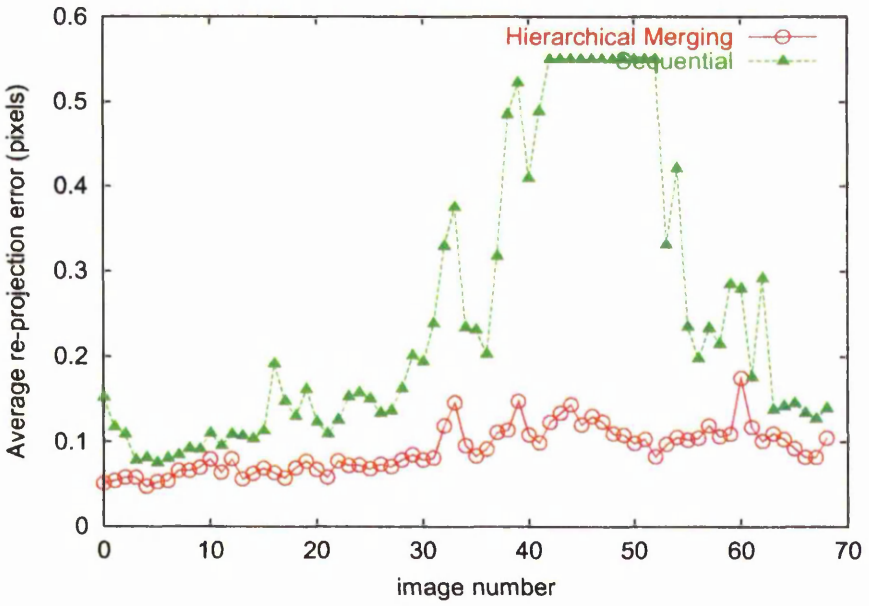

Figure 7.16: Re-projection error for points in all images of the table sequence

the different squared re-projection errors for the two methods with the resulting reconstruction after self-calibration. Both reconstructions has a very similar number of points in them (around 500).

Finally, to show the effects on long sequences, a sequence of 70 images selected from a video sequence of 280 images (figure 7.15) was taken and the re-projection errors measured for each image using both reconstruction schemes. The resultant graph in figure 7.16 shows large improvements from the hierarchical scheme. Note that, in images where the sequential method failed to produce a reasonable error, it has been capped to 0.55 pixels so as not to distort the graph.

Note that these graphs fail to illustrate one important point. This is that sequential methods have a bad tendency to accumulate error. This would be manifest as the graph for sequential error consistently getting worse and worse in consecutive images. This is not visible on the graphs presented in this section because at each stage new points were being added to the reconstruction. It is however visible if the errors on existing tracked points are observed, since these can be seen to accumulate. This was a practical necessity given the size of the image sequences used in these examples.

A further consideration, when interpreting the graphs of error, is that since these methods are robust the number of points also affects the re-projection error. In practice, it was found that fewer points were found with the hierarchical scheme but that they were tracked much further and with less error. In fact, in the table sequence, 3132 points with 19218 projections were found for the sequential scheme and 2700 points with 19144 projections were found with the hierarchical scheme. This means not only was reconstruction better for hierarchical merging, but so was the outlier rejection.

7.10.2 Summary

A very large quantity of results have just been given in the preceding section, all of which aim to prove certain points. An attempt will be made in this section to summarise the meaning of all these results, and produce a number of key conclusions.

- *Robust Triplet Reconstruction:* The robust merging approach to producing triplet geometry is without doubt superior to those based on the trifocal tensor. This is because only 4 parameters need to be estimated robustly, with the remaining 14 coming from estimation of the fundamental matrix using all data. It is also considerably faster, because the method comes complete with existing structure and requires no calculation to determine transfer error as do methods involving the tensor.

- *Triplet Reconstruction:* Merging algorithms hold their own for triplet reconstruction, but there is no truly significant difference in quality or speed. This means it is perfectly reasonable to start any merging based reconstruction from pairs and that no improvement will be achieved by starting from triplets determined using other means.

- *Robust Merging:* The robust merging approach does on the whole seem to be quite a bit more effective when applied to real data. It has been shown to fail less than the alternative robust resectioning approach.

- *Merging Overlap:* When merging, it is best to use two image overlap if speed is not a serious issue. Not only does it add robustness, but it increases reconstruction accuracy. Although it increases the number of merges that need to be performed, it also enables the use of robust merging criteria which enable slower random sampling algorithms to be avoided.

- *Merging Algorithm:* If the reconstruction is projective then only merging algorithms based on re-projection should be used.

7.11 Summary

This chapter has reviewed new techniques for projective reconstruction applicable to sizable image sequences. A number of methods based on reconstruction by merging were proposed, each tailored to certain types of image collections. A set of new techniques were then given for merging two projective reconstructions together as well as a number of robust methods. Finally, all techniques were extensively evaluated on both real and synthetic data and shown to provide increases (some dramatic) in accuracy, flexibility and speed over existing methods.

Chapter 8

Feature Tracking

8.1 Introduction

Key to all the techniques of the previous chapters has been the knowledge of features matched between images. Up to now this has simply been assumed, but in this chapter an entirely automatic technique for determining such matches across sequences will be presented.

In fact, this tracking of points across images is perhaps the most difficult problem in reconstruction. It is absolutely essential, if a reliable reconstruction is to be produced, for points to be tracked as accurately as possible, across as many images as possible with as few mis-tracks as possible. This matching and reconstruction process is complicated by image effects due to camera movement, lighting and sampling errors as well as by the potential for degenerate image pairings for which geometry cannot be calculated by normal means.

Because it forms the starting point for most image processing tasks, feature extraction and matching is a large and well developed field and so will not be addressed in any great detail here. Instead, a brief overview of the problems and specific techniques relevant to the case in hand will be given. After this, the chapter will cover the details of a scheme for tracking points through video sequences acquired using a hand-held camera. Such sequences can be large, and the motion can contain degeneracies.

Before continuing to the matching algorithm itself, a brief overview of the problems involved in feature matching and degeneracy of structure and motion estimation will be given. This entails first giving a description of the effects on image points of differing camera position and orientation as well as a description of the two main critical forms of motion degeneracy.

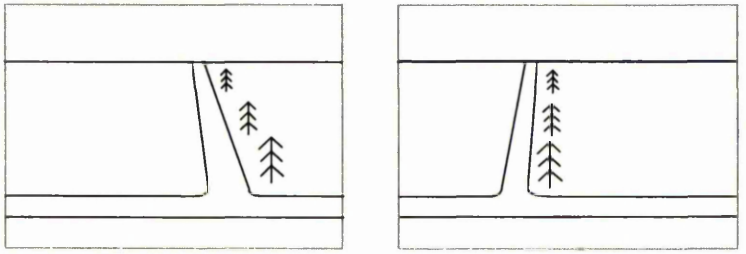

Figure 8.1: Perspective distortion due to motion parallax: objects further from the camera move less between the images.

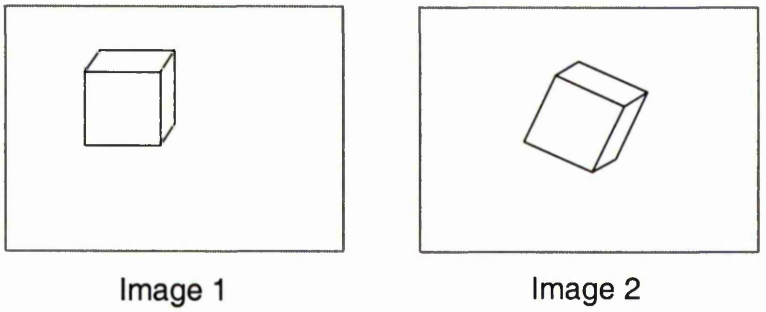

Figure 8.2: Image distortion due to camera rotation

8.1.1 Camera Motion and Image Matching

Matching schemes involve attempting to locate images of the same structure in both of a pair of images taken by a camera undergoing movement. One of the main difficulties associated with this problem is that, because the images are taken from different view points, perspective effects will cause certain distortions to occur between the images. In order to aid further discussion, an attempt will be made here to give an intuitive understanding of the effects of camera motion on the points visible in a scene. Additional effects such as aliasing, reflection, changes in lighting, noise in the electronics, camera internal parameters or image sampling will not be considered here.

Corrupting effects come from the two main forms of camera movement - camera translation and camera rotation:

- Camera Translation: the corrupting effect of translation in the images is motion parallax. This is illustrated in figure 8.1 where it can be seen that image points undergo different translations between the images, based on the distance from the camera of the world point being imaged. The extent of parallax-based distortion is determined by two main factors; how much depth variation relative to the camera there is, and how large the baseline is (distance between cameras).

  Camera translation causes two main problems for image matching; image occlusions and local inconsistency. For occlusions, features that were visible in one image become obscured by objects in the foreground and hence cannot be matched to. Furthermore, if there is depth variation relative to the camera, parallax means that a matching scene region will look different in the two images. If the baseline between the images is small, this difference can be so small as to be negligible. Parallax effects can only be removed if both camera motion and scene structure are known.

- Camera Rotation: this is rotation between cameras, causing rotation of the objects in the image (see figure 8.2). It is worth noting that this form of camera motion alone does not corrupt the image with parallax effects and so can be totally removed if the rotation is known.

So, how does this affect image matching? Many methods of image matching attempt to model these image distortions by using simplified models that hold true most of the time, particularly in small regions of an image. The most common assumption is that, away from occluding boundaries, the changes in image intensity between two images $I1, I2$ can be described with an image motion:

$$I2\left(x+\epsilon\left(x,y\right),y+\lambda\left(x,y\right)\right)=I1\left(x,y\right)$$

where the functions $\epsilon\left(x,y\right)$ and $\lambda\left(x,y\right)$ give the displacement between the images of the point at $\mathbf{x}=(x,y)$ in image $I1$. This measure basically states that all points in image $I2$ can be obtained by performing some transformation of all points in image $I1$.

At their simplest, the displacement functions take the form of a pure and constant translation $\mathbf{d}=(d_x,d_y)$ i.e. $\epsilon\left(x,y\right)=d_x$ and $\lambda\left(x,y\right)=d_y$ to give:

$$I2\left(x+d_x,y+d_y\right)=I1\left(x,y\right) \tag{8.1}$$

The simplest image matching measures, such as window based correlation (see appendix F) assume this form of model. However, it is rarely satisfied, even within a small image

region unless there is very little translation and rotation between the cameras (such as is likely to occur in consecutive frames of a video sequence).

For more significant camera motions, an affine motion model can give a far better representation of localised image distortion. In vector form, the displacement function for this affine motion model can be written as:

$$\delta(\mathbf{x}) = D\mathbf{x} + \mathbf{d}$$

where:

$$D = \begin{bmatrix} d_{xx} & d_{xy} \\ d_{yx} & d_{yy} \end{bmatrix}$$

is a deformation matrix, and $\mathbf{d}$ is the translation of the feature. A point $\mathbf{x}$ in $I1$ is moved by this model to image $I2$ as:

$$I2(D\mathbf{x} + \mathbf{d}) = I1(\mathbf{x}) \tag{8.2}$$

This model is capable of representing an affine transformation of the image plane, which means that unlike the pure translation model, it is also able to handle rotation effects as well as affine distortion (indeed any affine transformation).

Since projection is essentially a projective transformation, one final model suggests itself. This model is the familiar homography H - a $3x3$ homogeneous matrix representing an 8 parameter projective transformation between images. However, to express this in a linear manner, the points $\mathbf{x}$ must take on homogeneous form $\mathbf{x} = (x, y, 1)$ and the additional scale factors make solving for H much more difficult. It is still useful though, and is represented by (note that all quantities are homogeneous in this equation):

$$I2(H\mathbf{x}) \simeq I1(\mathbf{x}) \tag{8.3}$$

The advantage of the full projective transformation is that, unlike the affine form, it can account for very complex changes in camera internal parameters, including some camera distortions. Indeed, it can cope with any projective transformation of the image plane.

Note that all these models are not ideal, since they do not account for any depth-related effects (to remove those requires knowledge of structure), but instead account for constant image transformations only. However, particularly for planar scenes with small baselines, it is very common for these matching models to be near perfect in small image regions and away from occluding boundaries.

8.1.2 Similarity Measures

Once an approximate model of image motion, such as those given in the last section, has been decided upon, it is possible to use the simplified model to derive expressions giving the similarity between a pair of points in the two images. This similarity could simply be the sum of squared differences in a small $n \times m$ window, centred on the point at (x, y), i.e.

$$\sum_{i=-n}^{n} \sum_{j=-m}^{m} \left(I2\left(M\left(x+i, y+j\right)\right) - I1\left(x+i, y+j\right)\right)^2 \tag{8.4}$$

for deformation model M, as described in the last section (see [BYX82] for details of sum of squares methods). Instead, it could be a more complicated measure, such as normalised zero mean cross correlation - a sum of squared differences, weighted by the standard deviations of the image regions, and with image values normalised to have zero mean in the image regions (so as to produce normalised scores between -1.0 and 1.0). There have been many studies of image correlation measures (see for example [FP86, RGH80]). However, it is notable that, whilst most of the work on the pure translation model occurred in the 1970s and 1980, it is only in more recent years that models other than the pure translation one have been used for feature matching, such as the affine model ([TS94]) and the full homography (see [PZ98, TVPG99]).

8.1.3 Degenerate Camera Motions for Image Pairs

The other major problem encountered when attempting to track points, and also when attempting to produce a reconstruction, is degeneracy. That is to say motions between image pairs which are not best described using the fundamental matrix (see [TZM98] for a complete review and catalogue of degeneracy). However, only two of these degenerate motions are critical, in that, if they occur, correct estimation of the fundamental matrix cannot be performed. The other motions are also reduced forms of the epipolar geometry, but are best described by a different but equivalent form of representation involving fewer parameters (i.e. the same matrix with fewer parameters). However, even if these non-critical degenerate motions are encountered, a fundamental matrix along with cameras can still be estimated reliably, and so they will not be considered further here.

The two critical degeneracies simplify the epipolar geometry so much that the points in the images $\mathbf{x} \leftrightarrow \mathbf{x}'$ are related by an image homography $\mathbf{x} \simeq H\mathbf{x}$ instead of by a fundamental matrix. In these cases, a 2 parameter family $\mathbf{e}$ of fundamental matrices will fit the points $F \simeq$

$[\mathbf{e}]_\times H$. Consequently, when robustly estimating a fundamental matrix using such degenerate data, outliers will determine the remaining parameters $\mathbf{e}$ and so any reconstruction produced from the fundamental matrix will be useless. In more detail, the two degenerate forms of cameras and structure are cameras undergoing pure rotation and planar degeneracy. These will be described in detail in the two sections below:

Cameras Undergoing Pure Rotation

If a camera undergoes pure rotation about its centre, then there will be no motion parallax (consequently depth cannot be recovered). Instead, the transformation between the images is completely represented by a 3×3 homography H, corresponding to a transformation of the image plane from one camera position to the other (as described in equation 8.3).

Fortunately, provided that features can be tracked into the degenerate image from non-degenerate images, it is still possible to produce a reconstruction for the camera if this form of degeneracy exists. Since a projective camera representing pure rotation requires only 8 parameters instead of 11, it follows that the homography provides sufficient constraints to calculate the projection matrix relative to some existing reconstruction. This can be done either using resectioning, or alternatively, matching between the degenerate image and a valid image can be used to create a homography H that can be added to the camera for the valid image P as HP.

If a sequence of images is truly hand-held, then it is safe to say that it is unlikely that this rotational degeneracy will occur for very many consecutive frames, unless a deliberate attempt is made to rotate around the camera's centre. For a human with a hand-held camera, this would require rotating the camera on a tripod, or by turning the wrist. Neither of these is a particularly natural method of performing any large movement, and so both require premeditation. This means rotation cannot be expected to occur in large quantities, but it is still not important if it does since it may be handled elegantly.

Planar Degeneracy

The second degeneracy occurs if all the tracked features belong to the same plane. Unfortunately, this is fairly common in many types of scene - for example, imaging only one flat surface such as a wall or ceiling. In such cases, only a homography can be defined which provides a mapping of points on the given plane between the two images. However, unlike the rotational case, no camera matrix can now be produced, since the camera has a full 11

degrees of freedom but the homography only 8.

Note that, in the case of an affine or Euclidean reconstruction, a camera can be produced for the image, because in these cases the plane at infinity is also known and two planes provide sufficient constraints to completely determine a camera (see [Fau93]). Subsequently, without upgrading the structure to metric, it will not be possible to survive such degeneracies, unless matching between the images before the degenerate section, and the images after the degenerate section can be performed later on in the sequence (e.g. if the sequence returns to view something viewed earlier).

8.2 Tracking Across a Video Sequence

Now an outline of the problems associated with matching image features between image pairs has been given, it is possible to consider the problem of tracking across many images for the specific case of a video sequence. Since a video sequence is being used, it is safe to assume there is a very small baseline and hence very little difference between each pair of adjacent images in the sequence. The advantage of this is that, because the images are so similar, motion effects will be very small and so matching will be much easier. A disadvantage, though, is that these motion effects are the key to extracting depth and motion, and so reconstructions from small baselines tend to be unreliable. Similarly, the lack of motion means that consecutive image pairings are frequently near to, or actually are degenerate.

8.2.1 F-Based Tracking

One approach to tracking is to take pairs of images, extract interesting and salient features in both images, determine matches between these features using correlation, robustly estimate the epipolar geometry in the form of the fundamental matrix and then use the epipolar constraint to guide further matching. Consecutive stages estimate more fundamental matrices, and use them to build a reconstruction which in turn guides further matching. This approach to the correspondence problem, sometimes referred to as the F-Tracker, is well established and can be very effective (see [ZDFL95, ZDFL94, BTZ96, FZ98b, Pol99]).

However, this guided matching approach suffers from drawbacks when it is applied to complete video sequences. Whilst the small baselines of a video sequence reduce image parallax, and thus make matching much easier, the lack of parallax effects also results in highly inaccurate estimates of the fundamental matrix (and hence problems with guided

matching). Furthermore, small baselines are likely to represent camera motions that are degenerate or near degenerate for estimation of the fundamental matrix.

Consequently, F-based tracking is ideal for larger baselines, but needs to be modified to select between homography and fundamental matrix tracking, if it is to be used on small baseline sequences (for example, using the methods of [TFZ98, TFZ99]). However, in this case it is slow, and the use of outlier detection tends to drop tracks that are just poorly localised because of the small baseline rather than poorly matched.

8.2.2 Different Models for Different Baselines

Due to the problems with applying F-based tracking to every image of a video sequence, an alternative approach is to use simpler models for small baseline matching. In this case, the simpler translation image motion model (equation 8.1) is used to track points between consecutive images in the video sequence. This simpler image based model is far more accurate for smaller baseline matching, and only when the baseline between the images is determined to be large enough is the epipolar geometry actually calculated, outliers removed and the process of reconstruction started.

Matching Between Image Pairs

At the heart of this tracking method is a matcher for taking very small baseline image pairs and tracking features. The features to be used are points and are identified in a principled manner in the first image only, so that the n points that will match best using the translation motion model are used (see [TS94] for details). Because it is a pure implementation of the method in [TS94], in the interests of brevity only a brief overview will be given here.

The basis of the method is that, for a video sequence with very small baselines between images, the pure translation motion model of equation 8.1 is the most effective measure for matching (as shown in [TS94]). To actually match, features are either identified in the first image and/or carried over from previous matching. A match for each feature is then sought in the second image by minimising the sum of squared differences, using the pure translation motion model.

This minimisation is conducted on a per point basis, by taking a small image region around the selected feature in the first image. A no motion model ($d_x = 0$ and $d_y = 0$ in equation 8.1) is then used to initialise a Newton-Raphson style minimisation, which follows the image gradient so as to determine the translation that minimises the sum of squared

differences (equation 8.4). A multi resolution pyramid approach, which matches first at lower resolutions and then at higher resolutions, is used to enable this to work over larger image regions. The result is an extremely effective small baseline matcher, applicable to image pairs where there is unlikely to be more than a few pixels difference between matches. However, it is not effective if the no motion model is far from the truth.

A distinguishing feature of this scheme is that, unlike the F-based tracker which works to match between image pairs independently, it first attempts to track the existing features carried over from the previous pair. If any tracks fail, new features are found to replace them so as to maintain a constant number of features. The advantage of this is that it puts an emphasis on maintaining feature tracks for as many images as possible, resulting in fewer points, but much longer and more reliable tracks. This greatly reduces the number of outliers and speeds the matching, but is only effective for very small baselines. Feature to feature matching (as opposed to tracking) is much more reliable for larger baselines, particularly if the affine or projective image motion model is used.

Note that extensions to this scheme have been proposed in the literature, e.g. [TFTR98], but are mostly concerned with outlier rejection. For the purposes of this work, it was found that the number of outliers produced by the original technique were so few that outlier rejection was best handled more accurately by the later reconstruction stages.

8.3 Frame Selection: Selecting Image Pairs to Start Reconstruction

The tracker just described in section 8.2.2 above is so effective that it does not need to be supplemented by guided matching using the fundamental matrix. In fact it rarely drops more than a few tracks out of a few hundred (mostly due to occlusion or severe perspective effects) for each image, and amongst the remaining matches there are very few outliers.

To illustrate the effectiveness, figure 8.3 shows the length of the tracks for points across a sequence of 279 images, after outlier rejection using full geometric models, and using no extra matching except the image pair scheme just described. As can be seen, on the whole, points are tracked for a great number of images, on average just less than 30 images. To give a further insight into the accuracy, figure 8.4 shows the longest tracked match (images 1 to 161) plotted in the first and last images of the track. The difference in the images is undoubtedly very large and the point has been accurately matched about as far as it possibly

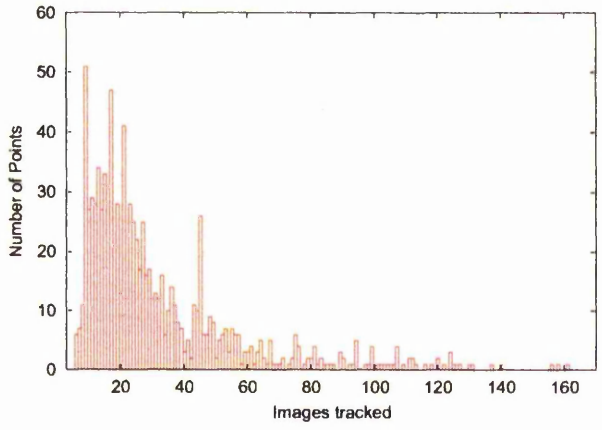

Figure 8.3: Graph of track length across sequence of 279 images

can be. The point was also fairly accurate, being reconstructed with an average re-projection error of 0.45 pixels across all 161 images.

Consequently, the matcher can safely be used to very quickly and accurately track points across large numbers of images. All that remains is to identify pairings of images that represent the optimal trade off between accuracy, number of matches and baseline size and then use the tracks between these to start a robust hierarchical reconstruction.

More precisely, the ideal pairings of images to start a reconstruction should have the following desirable properties:

1. Should not be degenerate - hence enough parallax effects in the point set larger than the image noise to enable accurate estimation of the epipolar geometry.

2. As many matches as possible

3. Should produce a low re-projection error after reconstruction

These properties can be detected by using different properties of the set of point matches without referring to the images directly. A discussion of some relevant properties are given in the following subsections.

8.3.1 Detecting Degeneracy

In order to test whether the image pairing is near degenerate, and the extent to which the image pairing exhibits parallax (criterion 1), a homography can be robustly fitted to all the

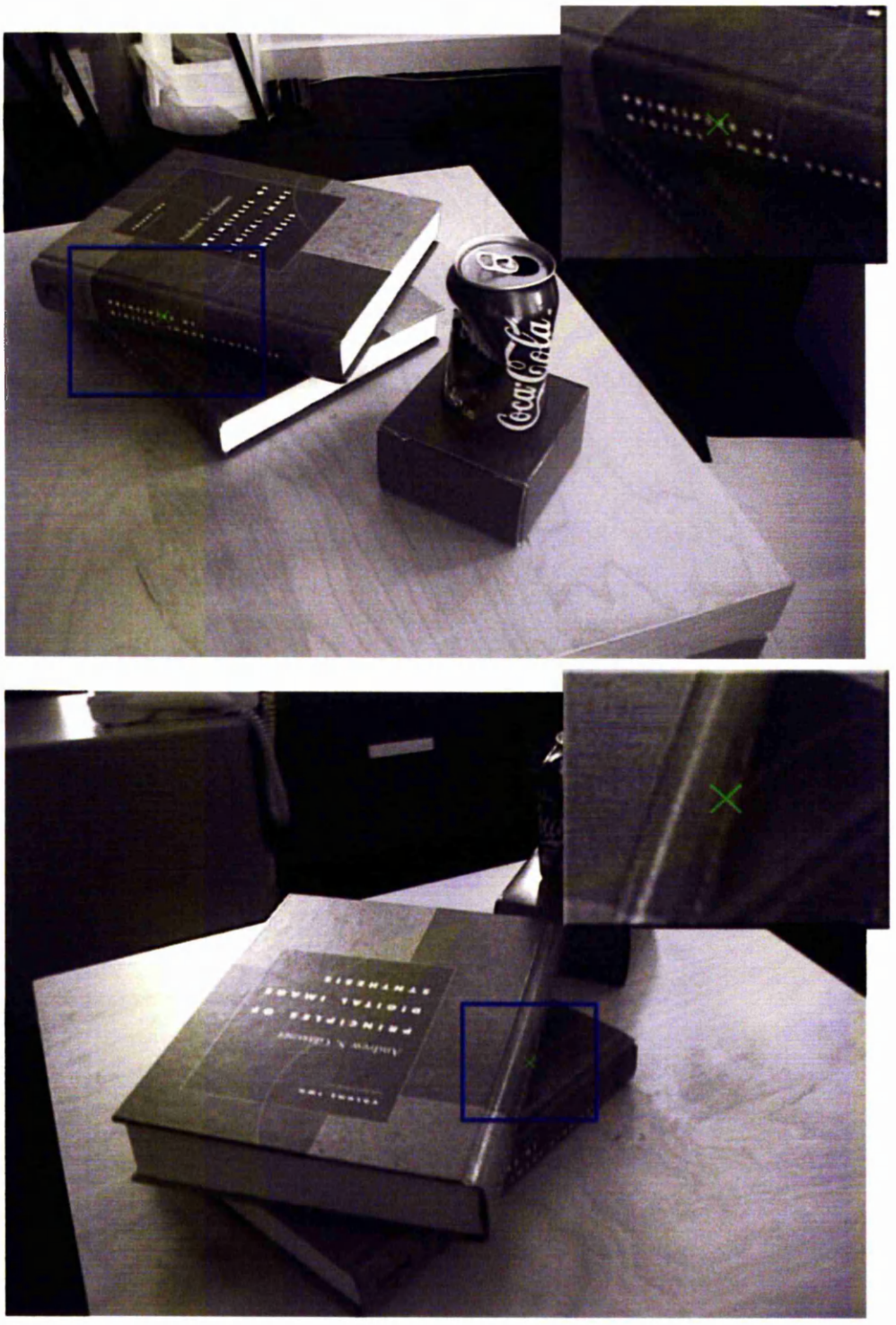

Figure 8.4: Extreme images for the longest track in the video sequence, between images 0 and 161

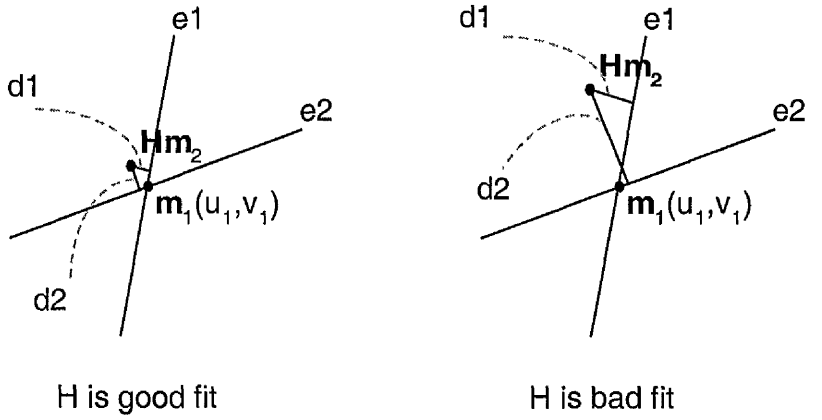

Figure 8.5: Effect of epipole position on epipolar error if a homography H can or cannot be fitted to the points well. If H is a good fit, then whether the epipole is selected to be $e1$ or $e2$ makes little difference to the errors $d1$ and $d2$ (perpendicular distance to epipolar lines). On the other hand if H is a bad fit whether $e1$ or $e2$ is selected to be the epipole makes a large difference in the errors $d1$ and $d2$.

matches in the two images. A homography will provide a good description of the differences between the image points, only if the points are at least effectively co-planar for the given degree of camera translation (this can happen for pure rotation or small translations). Note that this can also happen if the data is dominated by one plane, in which case the robust selection will remove all the points off the plane. Such a situation is also undesirable for purposes of reconstruction because depth variation is beneficial for geometry estimation.

All this means it can safely be concluded that, if a homography fits the data well, it will not be good for estimating the fundamental matrix. This can be intuitively understood by considering the fundamental matrix after decomposition into the epipole $\mathbf{e}$ and a homography H as $F = [\mathbf{e}]_\times H$ (refer back to section 3.6.3, page 67 for a full description of this decomposition and what it means). Basically, this decomposition considers epipolar transfer as having two components - firstly a homography that transfers a point from one image to the other image and secondly, the corresponding epipolar line is then found as passing through both the epipole and the transferred point.

From this decomposition, it follows that, if H transfers points very accurately, then the constraints on F will be satisfied regardless of where the epipole e is situated. For an example see figure 8.5 where the perpendicular distances to epipolar lines $d1$ and $d2$ are very similar for a good fitting H, but not for a bad fitting H. This explains why, for small baselines, the

epipole has been observed to be the least accurate component to be estimated in F (see [LF96b]). From this, it can be concluded that, if a homography can be fitted well to the data, then regardless of why it fits the data well, the pairing is not going to produce a good reconstruction of camera centres (or their images, the epipoles).

Comparing the residual produced by homography fitting to the residual produced by fitting a fundamental matrix unfortunately does not present a viable solution. Invariably the fundamental matrix will produce a lower residual because it is the more general model. To handle this problem, there are a number of methods that can deal with this problem, most notable of which are the relative GRIC scores. These will be discussed in more detail in the context of detecting (rather than avoiding) degeneracy in section 8.5.

8.3.2 Number of Matches

To handle the second criteria and ensure a large number of matches are available, the most simple scheme would try to maintain as many tracks as possible between image pairs. However, this is not necessarily the best criterion because each pairing of images will later be used to produce a complete reconstruction, and so only points that track across image triplets will be able to contribute to reconstruction of more than two images. This is important because robust reconstruction for triplets is considerably more accurate than pairs due of the strengthening of the geometric criterion (matching points are constrained to an exact position and not to a line).

8.3.3 Epipolar Error

To deal with the third criterion, the fundamental matrix can be robustly estimated. Provided there isn't degeneracy, the lower this residual is, the better the quality of reconstruction. However, care needs to be taken because fewer matches usually produce lower residuals than a large number of matches. Subsequently, when comparing different fundamental matrices, the number of matches also needs to be considered to make valid conclusions about reconstruction error from the epipolar error.

8.4 A Simple Frame Selection Algorithm

For the purposes of this work, a very simplistic approach was found to be very effective for purposes of frame selection (but it could doubtlessly be improved). In the implementation

the desirable properties are enforced by using a number of functions of the matches. These functions are combined into a similarity score and, at each stage, the pairing of images that minimises this similarity score is selected to start the reconstruction.

The algorithm itself is fairly straightforward. Starting from the first image, all possible pairings of the first image with consecutive images in the sequence are considered. This is continued until the number of matches tracked between the first image and the image under consideration falls below 60% of the total number of features in the first image, and the image pairing is non-degenerate (more precise details are given in the algorithm summary). The similarity score is then applied to all images that fall within this category, and the image pairing that minimises it is selected. The minimal scoring image then becomes the new first image, and the process repeats until there are no more images left. The whole process terminates either when there are no more images, or when the last image is included in the pairings and some suitable image pair cannot be found.

All that remains now is to define the similarity score. There are a number of different functions involved in this, each of which is used to detect different criteria from the list used in the discussion section 8.3. Note that all these scores only aim to give an approximation.

8.4.1 Epipolar Error

To measure the accuracy of reconstruction, the fundamental matrix can be robustly estimated using MLESAC, and a the median epipolar error squared, r^2 used. Note that the median epipolar error squared is used rather than the Huber function so that no judgement need be made on inliers and outliers.

8.4.2 Degeneracy

To measure degeneracy, the value $\frac{r^2}{v^2}$ is used where v is the median residual of all points to the robustly estimated homography and r is the median residual of all points to the robustly estimated fundamental matrix (as defined in section 8.3.3). It follows that the smaller this value is, the worse the homography fits, and the diminishing nature of the $1.0/x$ function means very bad homography fits are not particularly favoured over bad fits. This is desirable because in effect a bad fit is just as good for epipolar geometry estimation as a very bad fit, and so the selection between the associated image pairings should be made based on other criteria than homography fitting.

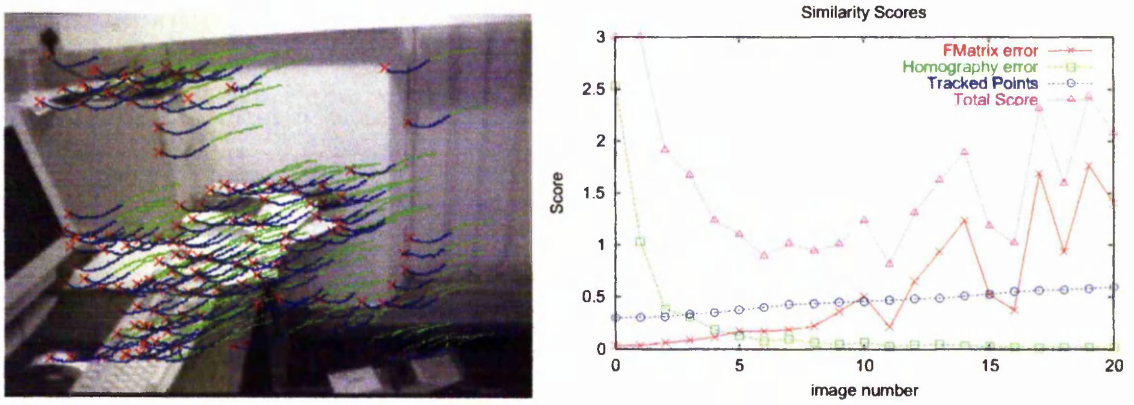

Figure 8.6: Similarity score and tracks for sample sequence section of 21 images

Note that this measure attempts to obtain a certain degree of invariance to the amount of noise in the images by using the epipolar error as well as the homography fit error. This is far from the ideal approach to this, which would use the covariance matrices to determine the actual image noise given the residual errors for the particular model. Consequently, the relationship between the two measures could be based on this less model specific measure. However, such a scheme would be complex, and it will be shown in the results section 8.4.6 below that the method presented here represents a practical solution for the range of errors to be expected under normal circumstances.

8.4.3 Number of Tracks

Bearing in mind the discussion of section 8.3.2 it is not desirable to pick images so far apart that large numbers of points cannot be tracked across triplets of the selected images. To remedy this, the number of tracks that are counted is the number of tracks that are carried over from the previous pairing into the current pairing (i.e. share the same feature in the second image of the triplet). Note that, to handle the first pairing of images easily, the number of tracks between the image pair can be used instead.

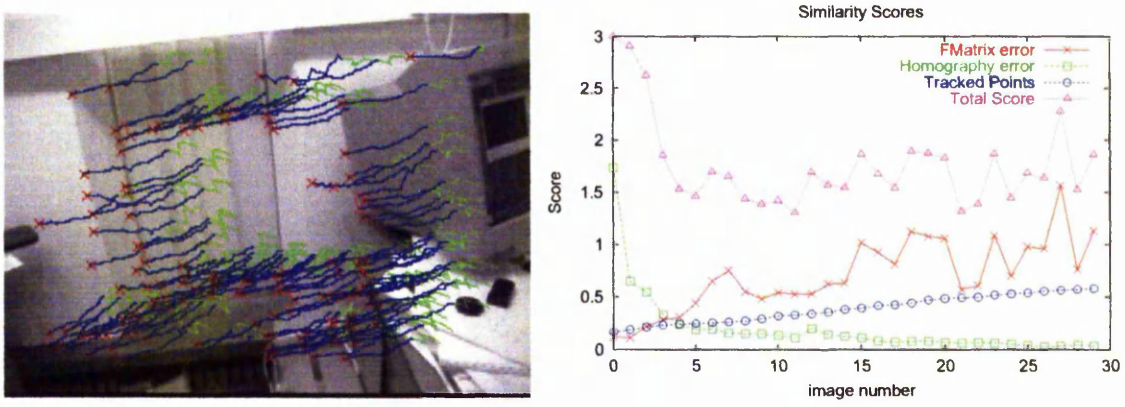

Figure 8.7: Similarity score and tracks for sample sequence section of 30 images

8.4.4 The Complete Criterion

The three functions just outlined are then weighted and combined very simply to give the complete score to be minimised:

$$\frac{r^2}{v^2} * w_1 + \left(1.0 - \frac{n}{m}\right) * w_2 + r^2 * w_3$$

where v is the median squared homography residual, m is the number of features in image 1 of the pair that are tracked from previous images, n is the number of features tracked to the second image of the pair and r is the median squared epipolar error residual. The weightings w_i allow the significance of each score to be altered.

The weightings used are fairly important. In the authors implementation, w_1 is set to 15.0 and w_3 is set to 1.0, selected so that a homography that produces a 15 times greater residual than the fundamental matrix will be assigned a score exactly the same as the fundamental matrix. This is the point at which the homography score is considered just as relevant as the fundamental matrix score. Finally, because, the number of points is unimportant relative to their distribution amongst different planes in the image (this distribution is detected by the homography fitting), w_2 is set to 1.0.

These weightings are designed so that the homography estimation is dominant if effectively degenerate motions are encountered. Assuming a bad fit for the homography (15 times worse residual in this case), the homography measure becomes less relevant and the best image pairing is selected based on reconstruction error and number of points. If the reconstruction error is consistently low, then the best image pairing is selected based on

number of tracks, otherwise it is selected so as to minimise the reconstruction error. This gives a good principled balance.

Note that different weightings and different measures can easily be used to provide different emphasis, e.g. to try and include as many images as possible at a time, to increase accuracy, or to account for very poor quality images. Exactly what is best can be determined by other factors such as speed and accuracy requirements. Weightings also need to be adapted to the response of the particular implementations used to calculate homographies and fundamental matrices. For example, if the fundamental matrix estimation code is much more effective than the homography estimation code used here, a 15 times difference may no longer be appropriate.

To help illustrate the way these measures relate to each other, figures 8.6 and 8.7 show graphs of how the unweighted individual scores and the total score vary as the image pairing varies. Also shown is an image of the features and where they have been tracked. The track line in green indicates the part of the track that was before the minimum in the similarity score, and the part in blue indicates the part of the track that was after the minimum. The graphs of similarity score show that a reasonable amount of the track is selected, with the score initially getting lower as the baseline gets larger. Then, when the tracker starts to become inaccurate because of the distance, the epipolar error increases. In the second example, figure 8.7, the score reaches a minimum at the end of a very jittery section of camera movement. This is because such movements cause tracking inaccuracy and also degeneracy (since the effective baseline is small). Due to the effective degeneracy, the break has been deferred until after the degeneracy has disappeared (after the jittery section).

8.4.5 Algorithm Summary

To summarise, the steps of the algorithm are as follows:

1. Extract m features in the first image as described in section 8.2.2.
2. Attempt to track all features into the next image using the method of section 8.2.2.
3. For all untracked features identify new features to replace them and start tracking them instead. Attempt to maintain m features whenever possible.
4. Return to step 2 until no more images.

5. Calculate similarity score from section 8.4.4 for pairing of first image with each consecutive image in the sequence. This is continued until less than 60% structure has been tracked into the second image. At this point, if a homography has not been found which has a median residual greater than 4.0 pixels then pairing continues until 30% structure has been tracked. Can also increase speed by terminating if n poor quality fundamental matrix fits are observed in a row (e.g. median residual error greater than 3.0 pixels).

6. Select best scoring image pairing and calculate a reconstruction for that pairing to initialise the hierarchical scheme.

7. Consider the second image as the new first image and return to step 5. Terminate if the second image is the last image of the sequence and if the median residual for the homography fit, on the selected image pairing, is less than 4.0 pixels.

8.4.6 Results

Now the scheme has been presented it will be appropriate to discuss some of the issues and possible improvements. Where relevant, these will be illustrated with examples on synthetic data.

The first issue to be addressed is whether or not this scheme is actually any good at detecting degeneracy. To this end synthetic sequences have been generated using the same trajectory based method as described in section 7.8.1, page 141, but this time matching failure is modelled by removing up to 1% of points from each image (rather than 15%). In addition, a large degenerate section of 40 images has been placed at the beginning of the sequence during which only rotation and no translation occurs.

The frame selection process was run on this sequence after noise of 0.4 and 1.0 pixels standard deviation had been added. The results can be seen in figure 8.8 for the different measures. Whilst there is no doubt that the degeneracy dominates the score when it is prevalent, these graphs also illustrate the behaviour of the method when there is no degeneracy. When the image noise is very low as for the top graph, it can be seen that the number of points and homography fitting error dominate the score for the non-degenerate section. On the other hand, when the image noise is high (bottom graph), fundamental matrix error dominates the score. This behaviour requires correct selection of the weightings such as those given in the algorithm description.

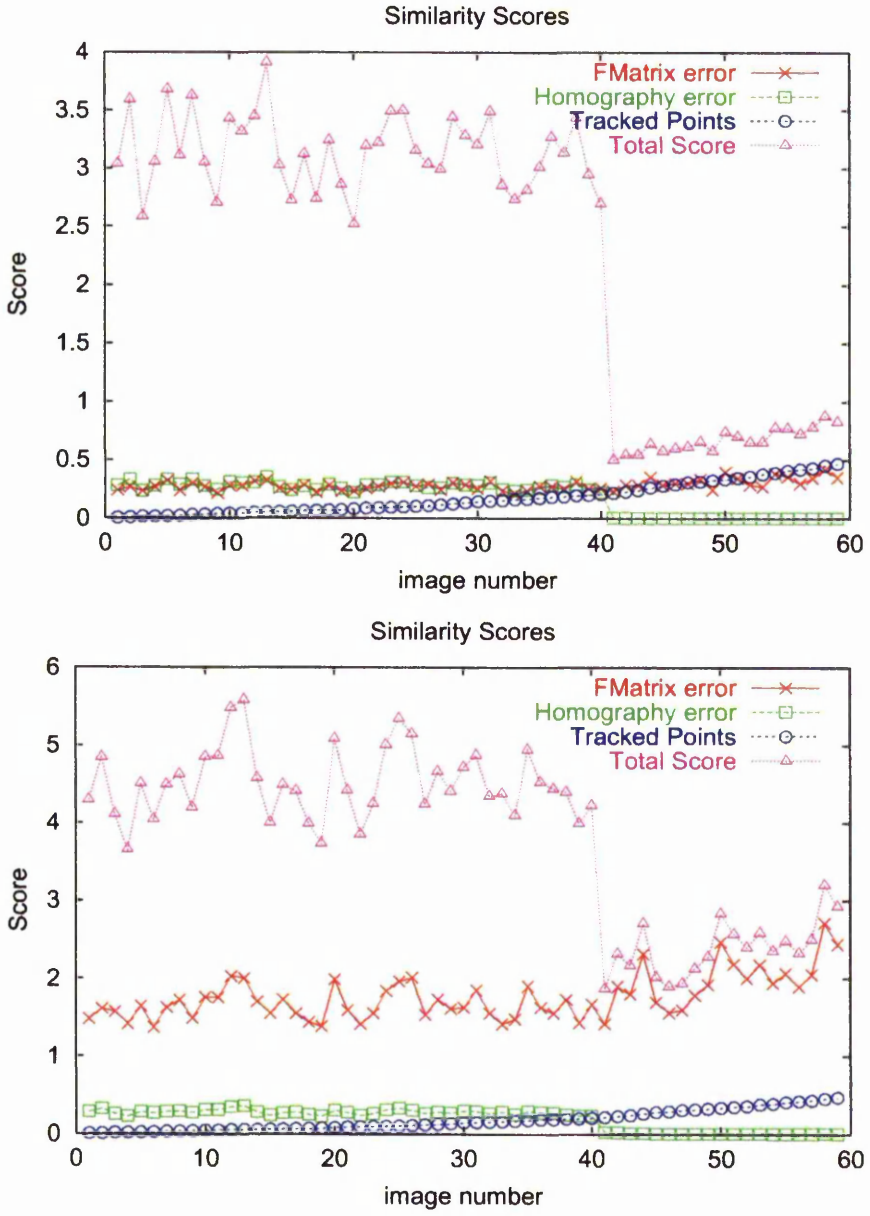

Figure 8.8: Frame selection scores for sequence with a large degenerate section (images 1 to 40). The top graph shows the scores for added noise of 0.4 pixels standard deviation and the bottom for 1.0 pixels.

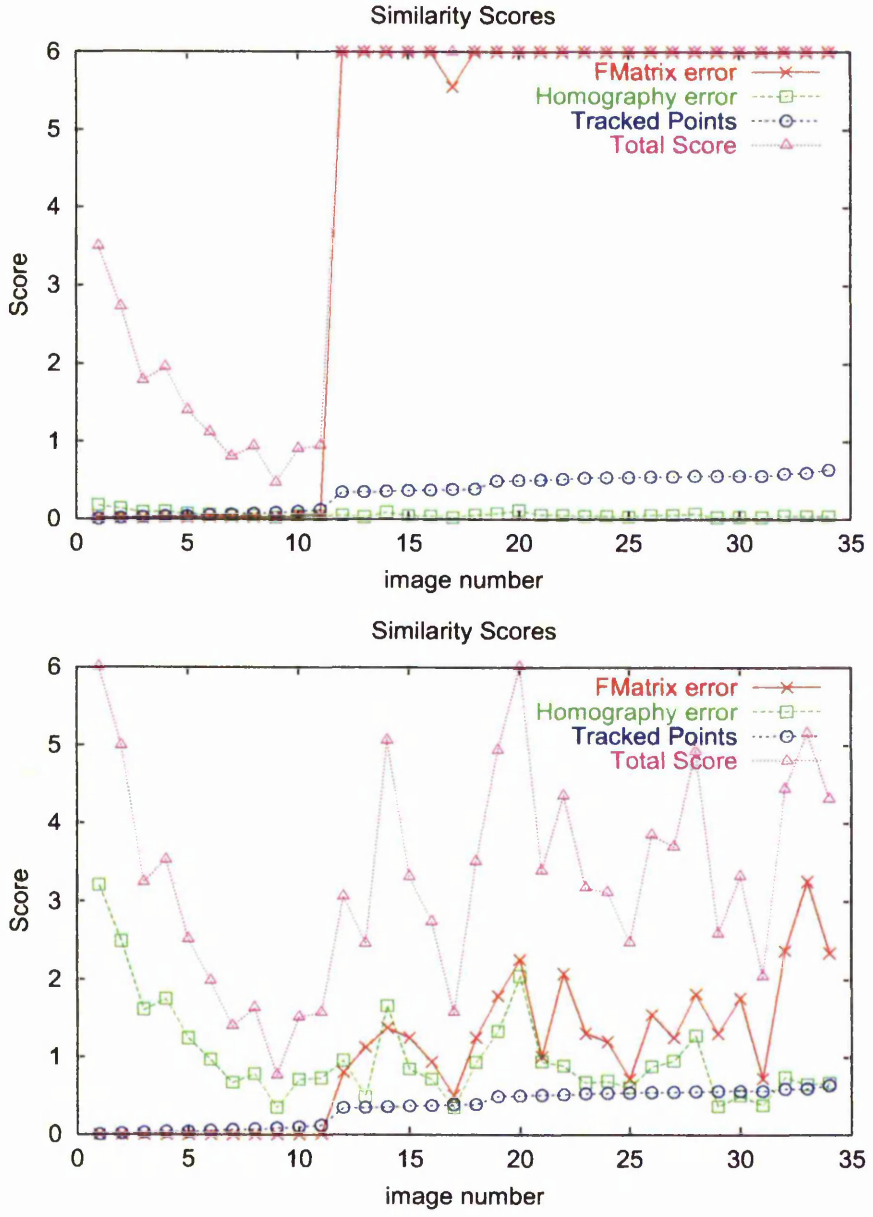

Figure 8.9: Frame selection scores for sequence with a large gap that breaks tracking. The top graph shows the method presented in this chapter, and the bottom graph a normalised method (see text for details).

It may seem a little odd that the method presented here uses absolute values for the error measure, rather than for example selecting some normalised criterion. After all, lack of normalisation means the weightings must be selected to match the particular implementations used for homography and fundamental matrix estimation. There is more than one reason for this.

Firstly, normalisation is not appropriate to the case in hand, since the larger the value the more necessary it is to make it as low as possible. For example if the epipolar error in fundamental matrix estimation is very large, then it is much more important that a pairing be selected in which the epipolar error is low, than a pairing for which there are a lot of matches. Similarly, if there is a very low re-projection error for fundamental matrices then it is more desirable to pick a pairing with more matches than it is to pick one with a relatively low error.

The second and most important reason for not using normalised error criteria is that a large variation in values is often made insignificant by the normalisation. For example, consider the sofa sequence in figure 10.3, page 211. This sequence has a very large gap in it at image 11 which renders matching highly inaccurate. Ideally the frame selection method should be robust enough to detect this and select a lot of images around it to reduce problems.

Figure 8.9 shows the frame selection graphs over the broken section using the presented method (top graph) and the same method, but this time with the errors from the fundamental matrix and homographies normalised so as to have a mean value of 1 (bottom graph). The normalisation smooths out the huge jump and subsequently selects a pairing over the jump at image 11 which reconstructs to a reprojection error of 1.418 pixels (selected images are 0,9,29). On the other hand, the non normalised version selects images around the jump (0,9,11,21). This time, the pairing over the jump has an error of 0.578 pixels and a much better reconstruction is produced. No matter what form of normalisation is used (e.g. median error) normalisation will always have the potential of removing large bumps in the data.

8.4.7 Discussion and Other Work

The idea of frame selection is by no means a new one. It is fairly apparent that, when handling video sequences, the amount of data needs to be greatly reduced. Methods have existed for some time for reducing the number of frames used in video compression. Similarly, the technique has previously been applied to structure and motion [Nis00], but in this case a

totally different approach to that taken here was used. For a start, feature matching between images occurs after rather than before frame selection. This is not going to offer the same facility for selecting high quality data for the subsequent reconstruction algorithms because the data is not yet available.

Furthermore, in the method of [Nis00], no explicit attempt was made to avoid degeneracy, and in fact it was even encouraged by not allowing frames to be selected between which the points had moved too far. This is not a good test because a large rotation will cause this effect but will not result in increased matching difficulty. Recall that rotation is not associated with perspective effects or occlusion and so matching is much easier for rotation.

Similarly, images were prioritised based on a sharpness measure which attempts to account for image blurring effects. These effects are fairly unimportant to structure and motion, since all that matters is that accurate matching can be achieved. It seems better to base the image selection on accuracy of reconstruction, rather than on better looking images. This is not to say selecting sharper images is not a good idea, particularly if textures are to be extracted, but this would be more relevant to later stages of processing than structure and motion.

Finally, the method in [Nis00] also includes a simple shot detection method (i.e. a change from filming one scene to filming another). In this work, the need for shot detection is assumed not to be present, but it would doubtlessly increase robustness to detect a complete failure of matching (as would occur at shot boundaries). Unlike the approach in [Nis00], for which matches are unavailable, this could probably be achieved by looking for a failure in matching, manifest as very few matches or as most matches being outlying to the fundamental matrix and homography.

The only other big advantage of the method in [Nis00] is that it works as a batch process rather than working from one end of the sequence to the other. This is mainly possible because the selection measure is much simpler (e.g. sharpness is a per image trait rather than degeneracy which is an inter image trait). However, it does not seem likely that the batch approach offers much to selection, except ensuring that the final image and first image is included in the reconstruction and allowing better balancing. However, using the matching method in this work, matches are available for the final image and so, if desired, the final image can always be added back into the reconstruction at later processing stages by using the resectioning technique of section 5.3.1, page 98.

One final possible improvement could be made, and that would be in changing the method used to detect degeneracy. It seems likely that some other measure than raw homography

fitting could be more appropriate. For example, the relative GRIC scores (see section 8.5 below) or some measure based on detecting multiple solutions for the fundamental matrix. For example, when using the method of 4.3.1, page 79 checking to see whether the eigenvectors associated with the two lowest eigenvalues both represent viable solutions (see [Tor95]).

Note that if there are multiple solutions for the fundamental matrix, degeneracy must exist because for more than one solution to occur, one or more parameters must be irrelevant or near to irrelevant to the minimisation criterion (hence the fundamental matrix is over parameterised). However, this will also detect non-critical forms of degeneracy.

8.5 Detecting and Handling of Degeneracy

So far, no consideration has been given to actually handling and detecting degeneracy, only to avoiding it. This comes down to selecting the most appropriate means of expressing the image motion after the frame selection - either a homography or a fundamental matrix. However, this requires a more rigorous approach than was used in avoiding degeneracy and so it is not easily possible to compare the number of outliers or error residuals directly since the more general model (the fundamental matrix) will always produce a lower residual and have more inliers.

To deal with this, it is necessary to use alternative measures. Some work exists for the detection and handling of degeneracy for motion estimation [TZM98, TFZ99, TFZ98]. For the purposes of this work, the approach based on GRIC presented in [TFZ98, TFZ99] was used.

The Geometric Robust Information Criteria [Tor97] or GRIC for short, is a robust model selection criterion. It is based on an extension of the existing AIC model selection criterion (see [Tor99] for details). It calculates a score function for each model to be tested (in this case homography and fundamental matrix fitting) that takes into account the inliers, outliers, the residuals, standard deviation of the errors and the relative number of parameters and dimensions of the models.

When the GRIC score indicates a homography to be the most appropriate fit, it becomes necessary to determine whether this is because of a pure rotation or a planar motion (i.e. image of only a single plane). This can again be evaluated using the GRIC score (see [TFZ98]). If a rotation is found, then the projection matrix can be identified using resectioning. This is handled at the start of hierarchical reconstruction, by absorbing the images undergoing

rotation into the two image pair reconstructions on either side (so the number of overlapping images is increased), and then reconstructing as normal, but with some of the starting sequences not being pairs

Planar degeneracies represent far more of a problem and are not handled in the current implementation. To handle such degeneracies, it is necessary to self calibrate the camera using the planar images [Tri98, MC00b, MC00a]. A more general discussion of how this might be achieved will be given later, in chapter 10, section 10.4.2.

In [TFZ98] an alternative approach to handling degeneracy was proposed which did not use model selection directly. Instead, this approach allows matching ambiguities for a point in one image that matches to more than one point in the next image to be resolved by propagating potential matches from both fundamental matrix and homography tracking. The ambiguity is then resolved later when a complete track is available. This is unnecessary in this work because matching is not guided by a full structure and motion model and because tracking rather than independent per image pair matching is performed.

8.6 Guided Matching for Merging Based Reconstruction

So far, only matching using consecutive images has been considered. However, at higher levels of processing, when a reconstruction is available, it becomes possible to use estimated cameras and structure to guide matching. Since the reconstruction method being used here is hierarchical, at each merge a pair of complete reconstructions are available to be matched to each other. This means that both reconstructions will have their own set of structure which will need to be matched into the other reconstruction thus both complicating and simplifying the matching.

Because merging involves two sequences, both with well established structure, there are two types of potential matches. Matches from structure to new features in the other sequence, and new matches from structure in one sequence to structure in the other sequence. This leads to two separate matching algorithms, one for identifying structure to structure matches, and one for identifying structure to feature matches.

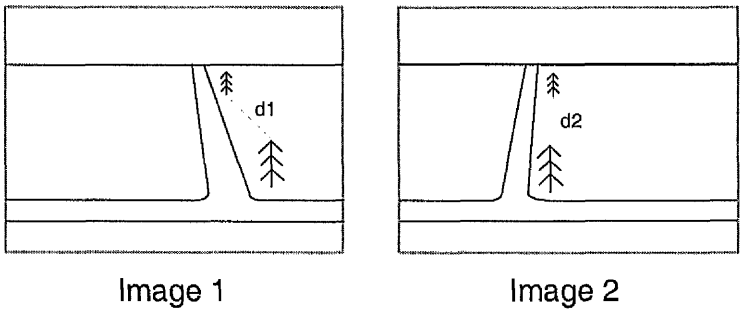

Figure 8.10: Illustration of the effect of parallax on distances between points in an image. Only parallax causes this effect, all other Euclidean transformations preserve length (by their very definition).

8.6.1 Determining Similarity Between Image Pairs

Before continuing, it will first be necessary to define a method for determining the suitability of a pair of images for matching. This method will have use in both forms of matching, either to select the best image pair for correlation or to avoid the overheads induced by attempting to match between unsuitable image pairs.

Because it will be used for determining the potential for matching between image pairs using the affine or translation motion model, the similarity measure to be presented here has been designed as far as possible to be sensitive to disruption of the affine image motion model. Since it is very difficult to actually be sensitive to the full affine model without parameter estimation, an assumption is made that the image motion is Euclidean (not metric since scaling is a by product of translation) i.e. an affine model, as in equation 8.2, but where D is orthogonal (a rotation).

In order to be sensitive to motion parallax, the measure is based on the difference in distance between images of the same structure in the two different images. This is illustrated in figure 8.10 where motion parallax has made the same measurement (distances d_1 and d_2) different This distance measure will be invariant to Euclidean transformations of the image plane caused by rotation or planar motion, but not to motion parallax caused by camera translation (this can also cause scaling). It follows that the bigger the difference the worse the potential match.

Naturally, it is not efficient to perform this distance test for every possible pairing of points. It would however, not be effective either because matching relies on similarity in

only a small matching window. Only localised distortion is of interest, and so the distance test is only performed for each point and its nearest neighbour.

So, onto the algorithm itself. Firstly, all structure from the reconstruction that is connected with features in the first image is identified and then projected into the second image. At this stage, it is important that any points that project outside the bounds of the second image are discarded from further processing. Furthermore, if less than a certain percentage of points (20% in the authors implementation) can be projected into the second image then the similarity measure result is set to 0.

The outcome of this is a set of pairings between points in both images. To actually calculate the measure, the distance from every point in the first image to the nearest neighbouring point is found, and the pairings used to find the distance between the same two points in the second image. The ratio of these distances is then used as the measure of similarity. However, because it is not relevant whether the two points are separated by a smaller or larger distance, the ratio is always of the form:

$$\frac{\text{smallerdistance}}{\text{largerdistance}}$$

To get the complete similarity score, the ratios for all points are summed, and then divided by the number of points to give a final score between 0 and 1. It is also a good idea to multiply this score by the fraction representing the percentage of points that were projected into the second image. This helps weight against matching into images for which there is little common structure, but does have the drawback that it makes the error measure considerably more arbitrary. The major drawback with this measure is that, although it is invariant to rotation and translation of the image plane, it is not invariant to distortion caused by camera internal parameters. It is notable though that with good quality cameras this distortion is rarely sufficiently significant to cause an upset.

When merging, this similarity measure can be run for every possible pairing of images in the first sequence with images in the second sequence. The result is a matrix of values giving the similarity between the images. This matrix of values will be useful in the matching schemes to be presented in the following sections.

Other Approaches to Frame Selection

The technique in the previous section has been designed to help tackle two different problems. Firstly, to detect if a camera has returned to view something it has viewed earlier and to

select image pairings that best conform to the assumptions underlying correlation (and hence resolve problems of which image pairs to match).

Whilst the second of these problems has received very little attention elsewhere, the first problem of detecting for a returning camera has been addressed. If the sequence is calibrated then this can be solved for by using a distance measure between cameras. However, such a method can not be used with projective sequences, and alternative approaches have been developed (e.g. [Saw98]).

Implementation Details

The above algorithm can be very slow if it is not implemented efficiently. It is very important to optimise wherever possible, particularly if one image is to be compared to many others at the same time. In this situation, the common image can have all re-projections pre-calculated and then ordered to help find nearest neighbours.

However, the algorithm can still be very slow if a brute force approach is taken to determining nearest point pairs. A simple scheme is to first sort the list of all points by x coordinate, and then to find the nearest neighbour for each point, a search is carried out up and down the list from the point which has the nearest x coordinate. This search progresses through the list recording the best result so far, until the search turns up a point that has an absolute difference in x coordinates which is greater than the minimum distance found so far. Such points mark the boundary at which it is impossible to get a closer point given the points are ordered on x coordinate. This approach is efficient enough to calculate similarity pairings for more image pairs than could be reasonably used in a reconstruction.

8.6.2 Structure to Structure Matching

The first type of matching that will be addressed is between structure in one sequence and structure in the other sequence. Because, for the case in hand, the sub-sequences are projective, the only effective means of determining similarity between potential structure matches will be to use the projections of the structure in the images. If sub-sequences are Euclidean then distance between structure in 3 space can be used and a simpler algorithm produced.

It is fairly safe to say that two items of structure are similar, if all projections of the structure in one sequence are within a certain confidence limit of the projections of potentially matching structure in the other sequence. So, given an item of structure in sequence one

$\mathbf{X}$ that has been observed in images k, an item of structure in sequence two $\mathbf{X}'$ observed in images j, and a homography aligning the coordinate frames H, then for the structure $\mathbf{X}$,$\mathbf{X}'$ to match:

$$\forall k \; d_E^2 \left(P_k H \mathbf{X}', P_k \mathbf{X}\right) \leq 5.99 \left(\sigma_2\right)^2 \tag{8.5}$$

and

$$\forall j \; d_E \left(P_j' H^{-1} \mathbf{X}, P_j' \mathbf{X}'\right) \leq 5.99 \left(\sigma_1\right)^2 \tag{8.6}$$

The standard deviations σ_1 and σ_2 should be calculated to be the standard deviations of the relevant error measure above, for all the known matches between structure used to calculate the initial aligning homography H. For example, these could be matches inferred from common features in an overlapping image. Note that the error criteria above are the same as are used in a least-squares sense to calculate the merging homography (see equation 7.3 on page 124). Consequently, the constant 5.99 corresponds to the same 95% confidence limit.

However, these criteria are not ideal in themselves, because structure is often tracked for many many images and can drift. This is particularly relevant because the sort of matches this scheme turns up are tracks broken because of occlusion or reinstated because part of a scene has come back into view. Subsequently, it is not unlikely for a very good match to fail some of the above tests and so, rather than reject a match if any of the projection tests should fail, a match is accepted provided that at least n% of all projections in a particular sequence pass ($n = 80\%$ in the authors implementation). For balance, this means n% of the tests in equation 8.5 must pass as well as n% of the tests in equation 8.6.

Similarly, to ensure that structure in sequence one matches accurately, it is insisted that, for each particular sequence, at least 3 projections match. By enforcing this, it is possible to avoid the need for correlation because 3 projections are sufficient to accurately constrain the position of the structure. This is one of the big advantages of this matching scheme, that it avoids correlation and hence the need for the actual images.

Using the above selection criterion, all structure in sequence one can be compared one by one to all the structure in sequence two. Because the criterion is symmetric, there is no need to reverse the process to match structure in sequence two to sequence one. The net result will be a set of potential matches for each item of structure in sequence one to item(s) of structure in sequence two.

Although it is not necessary to insist that one item of structure in sequence one matches one item of structure in sequence two it is necessary to insist that none of the matches are inconsistent. This means none of the projections of the matching structure from the same

sequence should project to different features in the same image. If multiple candidates are found, the best solution is to reject the potential match with the least number of projections passing the selection criterion in equations 8.5 and 8.6. Note that this check is also necessary, even if there are not multiple candidates, since a new match may contradict a previous match.

Implementation Details

The above algorithm is completely workable in itself, but it is very important to note the explosion in computational effort required by matching all projections of structure in one sequence to all projections of structure in the other sequence. As such, it is very important to ensure efficient implementation.

Firstly, it should be noted that all projections of sequence two structure in sequence two images, i.e. $P'_k\mathbf{X}'$ and sequence one structure in sequence one images, i.e. $P_k\mathbf{X}$ can be pre-calculated. Similarly, the projection of each item of structure in every image in the other sequence can be pre-calculated i.e. $P_kH\mathbf{X}'$ and $P'_jH^{-1}\mathbf{X}$.

Even with this pre-calculation, there is the problem that very large numbers of point projections still need to be checked. Consequently, great improvements in speed can be made when checking the projections of a potential structure match, by stopping if any projection fails the projection test very badly. Severe failure is defined as any point projecting outside the 99.9% confidence limit, i.e. $13.82\sigma_2^2$ or $13.82\sigma_1^2$.

With these two speed enhancements, it is quite possible to merge extraordinarily large sequences. For example, on a Pentium II 300Mhz computer, merging two sequences of 32 images each with about 1500 points having approximately 8000 projections, matching can be completed in about 1/2 second resulting in 175 new structure to structure matches.

Despite these improvements, for even larger sequences it can become necessary to be more selective about which structure is used. In these cases, matching can only be attempted between structure that projects into certain images in the sub-sequences. These images can be selected using the matrix of similarity scores to pick only images that have at least one similarity score with an image in the other sequence that is above a certain threshold. This means that if a row or a column of the similarity matrix does not contain a score above a certain threshold (0.5 in the authors implementation) the image associated with that row or column should be disregarded. Only structure that has projections in the relevant images should then be tested.

One final note on the nature of structure to structure matching is that, given the constraint on the number of projections that must match, there is little point in attempting this sort of matching if there are fewer than 3 non-overlapping images in each sub-sequence. In general, few new matches will be obtained unless there are at least 5 non-overlapping images and plenty of occlusion. Essentially, this matching scheme is only really useful in scenes with a lot of occlusion or for extremely long sequences where the camera returns to view previous sections of the scene.

8.6.3 Structure to Feature Matching

As well as looking for structure to structure matches, it is also possible if desired to look for structure to feature matches (this should be done after structure to structure matching). Unlike the structure to structure matching, this sort of matching task is not symmetric and must be performed from sequence one to sequence two and from sequence two to sequence one, meaning that it can only be applied to smaller collections of images. For ease of interpretation, the scheme will be presented for the specific case of matching structure in sequence one to features in sequence two. The opposite matching scheme is simply obtained by swapping the sequences.

This form of matching is best performed after the merge of the two sequences. The merged subsequence can easily be considered as being two separate sub-sequences the same as those prior to merging. All structure common to both sub-sequences is ignored for this form of matching. The process then selects potential matches to features based on the projection of structure from sequence one into sequence two. To do this, all structure in sequence one that is not already matched into sequence two is projected into each image in sequence two, and then an image-based similarity score (similar to the one used for initial feature tracking) is calculated between the projected feature and some observed projection of the structure in sequence one. This still leaves some major problems to be addressed. If the sequence is large, this matching method is going to be intractable and secondly it is necessary to select the best pairing of images between which to do the match.

Selecting Images to Match

If the sub-sequences being merged are large, then matching all structure from sequence one into every image in sequence two is going to result in far too many correlations to be practical. Instead, matching is only attempted for structure that is visible in certain images

in sequence one.

To handle this, the matrix of similarity scores is used to select all image pairings between sequence one and sequence two images that have a similarity score above a certain threshold (0.3 in the authors implementation). Consideration is also given to memory usage, since the images from one sub-sequence will all need to be held in memory (or else some fairly complex caching scheme adopted). As such the images are selected so that at most the best n images will be used.

Then, for all the structure in sequence one observed in the qualifying first images, the structure is projected from sequence one to all relevant qualifying images of sequence two (all those pairings above the given threshold). If the projection is within the image, then it is added to the set of potential structure to feature matches.

Note that it is possible different pairings of images will produce the same potential structure to feature match, and so duplicates need to be detected and removed prior to full matching.

Correlate Potential Feature Matches

In this sub-section, the matching of an item of structure in sequence one into an image in sequence two will be considered. There are two main problems associated with this. The first is determining which pairing of images the correlation should be performed over. The second is matching for the inevitably wide baselines.

The first problem is that a potentially matching item of structure $\mathbf{X}$ in sequence one will project into a number of images in sequence one, for example images 1,2,3,5,6,9,10. Now all these points potentially match to the new feature in sequence two, but to correlate all of them would be both ineffective and time consuming. Ideally, the image that is closest to the image the new feature is in should be used where closest is defined by the pairing that best satisfies the motion model used for correlation.

So, how do we go about selecting which image(s) to correlate? For a simple algorithm, the closest image in the sequence could be used, in this case image 10. However, especially for long sequences, this is no guarantee. As an alternative, it is proposed to use the matrix of similarity scores defined in section 8.6.1. This matrix can be very simply used to find the pairing between the images in sequence one for which the feature has been observed, and the image in sequence two that maximises the similarity score.

The second problem that can now be addressed is how to match the points, given that the

baseline is very large. Again, the methods proposed in [TS94]) can be used, but the motion model varied according to the similarity score. The Newton-Raphson style minimisation, utilising image gradients is used again if the score is above a certain threshold (0.6 in the authors implementation) otherwise a normalised zero mean cross correlation. To initialise this model, it is given a translation relevant to the predicted features in both images (i.e. the re-projection). Even across very large baselines this can produce a very effective matching scheme, even if the pure translation is not a very good approximation to image motion. However, without a doubt it seems highly probably significant improvements could be made by adapting the gradient descent algorithm to use an affine image motion model.

This matching is done for each image in the second sub-sequence, working from the first image to last image of that sequence. The reason for this ordering is so that any matches found in sequence two can then be matched to later potential matches. For example, if structure from sequence one was matched into image 10 of sequence two, the next match into image 11 would involve a correlation between images 10 and 11. The net result is that if a point is picked up again in the second sub-sequence it is tracked using subsequent images.

8.7 Results

To give a basic illustration of the effectiveness of this matching scheme, the graphs in figure 8.11 show the results for tracking across a video sequence involving an orbital motion of a scene (the sequence illustrated in figure 7.15, page 157). The automatic selection scheme has selected 31 images which were then tracked using the methods in this chapter. It is fairly clear that the additional matching has resulted in much longer tracks.

8.8 Summary

This chapter has presented a scheme to track points across very large video sequences taken using a hand-held camera, as well as for practical detection of degeneracy in those sequences. Initially, this involves a scheme for tracking points using consecutive image pairings, followed by detection and selection of ideal image pairings with which to start the reconstruction process. A scheme is then presented to enable further matching, guided using the reconstruction.

Brief results were also given to illustrate the effectiveness of matching. Far more examples of the effectiveness in practical use will be given in chapter 10 where this tracking scheme will be used to produce a complete system for reconstruction.

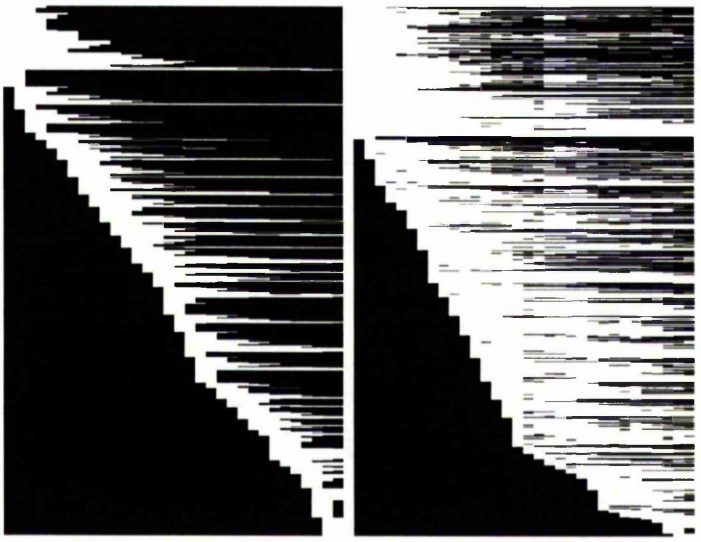

Figure 8.11: Graph of points tracked into images for normal tracking (left) and with extra matching during matching (right). White regions on the graphs show the point (y axis) is tracked into the given image (x axis).

Chapter 9

Rectification of Image Pairs

9.1 Introduction

Rectification is a process used to facilitate the analysis of a stereo pair of images by making it simple to enforce the constraints offered by the epipolar geometry. The process of rectification achieves this by making all matching epipolar lines coincident and parallel with an image axis. Many stereo algorithms assume this simplified form because subsequent processing becomes much easier if differences between matched points will be in one direction only. In this work, this will find use in aiding the problem of dense point matching.

In the past, stereo images were primarily rectified using optical techniques such as those discussed at length in [Sla80]. However, in more recent times, these have been replaced by software techniques that model the geometry of optical projection by applying a single linear transformation to each image. This effectively rotates both cameras until both image regions are due the same plane. [HG93] provides an example of this sort of technique which assumes that the camera(s) taking the images have been calibrated. In [Har95c], an alternative approach is given which assumes no camera calibration, and in [CTB92, LTSI96] a hardware implementation of planar rectification is proposed.

This approach of applying a single linear transformation is often referred to as planar rectification. It has significant advantages in that it is mathematically simple, fast and preserves image features such as straight lines. Unfortunately, planar rectification is not general and, if there is a large forward component in the camera movement it may produce unbounded, large or badly warped images. In the past, this was not a problem because stereo vision was usually performed using stereo rigs with near parallel cameras. However, recent advances in uncalibrated stereo vision, such as this work or [FZ98b, KPG98], have

focused more on hand-held sequences in which forward movement can frequently be present.

To deal with these restrictions, a cylindrical rectification technique was proposed in [SMI97] which used a separate transformation for each epipolar line. However, the technique was complex, omitted many implementation details and worked largely in 3D. A later work, [PKG99] overcame most of these problems, by using the tools of oriented projective geometry to perform a similar nonlinear rectification without using 3D.

In this chapter, a new general rectification technique is presented which further improves on these techniques. Firstly, it uses existing matches between the images, e.g. those used to calculate the epipolar geometry, to determine a linear transformation that makes epipolar lines coincident and minimises perspective distortion effects such as motion parallax between the images. Secondly, because epipolar lines are coincident, they can then be made parallel to an axis by applying the same nonlinear transformation to both images. Using the same transformation overcomes a problem with existing general techniques where the application of different nonlinear transformations to each image results in matching image features being warped differently in each image. Finally, it simplifies the approach of [PKG99] as well as addressing a number of unmentioned implementation details. It also handles a number of other minor problems overlooked by previous work, such as sub pixel coordinates and infinite epipoles.

9.2 Background

Before continuing to the rectification method, it will be appropriate to introduce and review some additional theory. This will set the scene for the new rectification method.

9.2.1 Oriented Projective Geometry

This chapter will rely on some concepts of oriented projective geometry. The reader is referred back to sections 3.7 (page 69) and 2.4.2 (page 36) for a proper description. In particular, the notion that matches between an image can be restricted to half epipolar lines instead of full epipolar lines will prove important. This simplifies the rectification procedure and prevents problems with matches occurring that refer to points behind one of the cameras.

9.2.2 Using a Single Homography for Rectification

It will now be shown how it is not always possible to keep rectified images bounded when rectifying using a single homography. From the axiom defining parallelism, two lines are considered parallel if they meet at the same point at infinity. This means that, for the case of a finite convex image region in two dimensional projective space, all epipolar lines passing through the image must meet at the same point on the line at infinity $\mathbf{l}_\infty$ for the lines to be parallel and the image to be rectified. This leads to the conclusion:

Proposition 1 If I represents a finite image region and $\mathbf{e}$ a point, then there exists a homography T taking $\mathbf{e}$ to infinity that keeps I finite if and only if $\mathbf{e}$ is not contained in I

This is proved in [Har95c] by considering that, if the transformed region I is to remain finite, $T^{-1}(\mathbf{l}_\infty)$ should be a line that does not intersect with I. Since I is presumably convex (images are usually rectangular), and it is possible when rectifying to make any line passing through $\mathbf{e}$ into $\mathbf{l}_\infty$, it follows that it is always possible to select a line that does not intersect I if $\mathbf{e}$ is not inside I.

9.2.3 Homographies Compatible with a Fundamental Matrix

As seen in section 3.6.1, page 65, it is possible to define an inter image homography that maps points on a particular world plane between two image planes. Such homographies can be used to perform linear planar rectification by applying one to each image so as to make all epipolar lines coincident and parallel (see [Har95c]). However, because making epipolar lines parallel can result in unbounded and badly warped images (see [Har95c] and section 9.2.2), only homographies which make epipolar lines coincident, but not necessarily parallel, will be considered here. These will be termed compatible homographies.

More concisely, given a fundamental matrix F for an image pairing and a match between the images $\mathbf{x} \leftrightarrow \mathbf{x}'$, a compatible homography H will transfer $\mathbf{x}'$ so that the resultant point $H\mathbf{x}'$ lies on the corresponding epipolar line $F\mathbf{x}'$. The result of applying this homography to all the points in one image will by definition be a pair of images that have coincident epipolar lines and hence the same epipole, i.e. $H\mathbf{e}' \simeq \mathbf{e}$ and $H^{-1}\mathbf{e} \simeq \mathbf{e}'$. The set of homographies that are consistent with the geometry of a particular image pair can be obtained from the fundamental matrix as: (see section 3.6.3 on page 67 for more details)

$$H \simeq [\mathbf{e}']_\times F - \mathbf{e}'\mathbf{a}^T \tag{9.1}$$

where $\mathbf{a}$ is an arbitrary 3 vector such that $\det H \neq 0$. Note that this means there is a 3 parameter set of homographies that are compatible with the fundamental matrix. Since compatible homographies can be considered as transforming points on a world plane between two different images, these 3 parameters can be considered to represent a plane in the scene. In image terms, these 3 extra parameters amount to defining a one dimensional projective transformation that is applied along all the epipolar lines in an image.

Most rectification techniques make no attempt to select the free parameters $\mathbf{a}$ in equation 9.1 using any principled manner. In [Har95c] rectification was improved by selecting these parameters so as to minimise perspective distortion between the images. In this work it is proposed to alter this approach so it may be applied to generalised rectification (by removing the lines becoming parallel constraint) to produce a number of improvements. Previous generalised rectification methods applied different versions of these free parameters to each epipolar line in each image. Subsequently, distortion arises which causes features not to look the same in both images for reasons other than perspective or photometric effects. In this work, the use of a compatible homography reduces these problems because the same parameters are applied to matching epipolar lines. However, it does not solve the problem totally, because different parameters are still applied to non matching epipolar lines.

One final point of note is that a compatible homography is a point to point mapping and so must enforce orientation. This is because, assuming the matches are correct, the mapping must map between the correct half epipolar lines because no points will match to incorrect half epipolar lines. Consequently, there is no need to enforce orientation explicitly as in the method of [PKG99].

9.3 General Rectification

The rectification method presented here comes in two stages. First, a compatible homography is selected so as to minimise distortion due to perspective effects in some supplied set of matches, and then applied to one image to make all matching epipolar lines coincident. Epipolar lines are then made parallel to an image axis by parameterisation of both images with polar coordinates centred on the epipole. Note that, because a compatible homography has been used on one image, the same nonlinear epipolar alignment process can be used for both images and so problems of inconsistent image warping avoided. On input, the rectification process expects to be provided with two rectangular images as well as a fundamental matrix and a set of point matches such as those used to calculate the fundamental matrix.

9.3.1 Determining a Compatible Homography

This section will address the method by which a compatible homography is obtained. This method uses known matches and the fundamental matrix between the images to attempt to find a compatible homography that minimises inter image distortion due to perspective effects.

To do this, an attempt is made to find the homography compatible with the fundamental matrix F that transfers points $\mathbf{x}'_i$ in image two as close as possible to their matches in image one $\mathbf{x}_i$. This will find the best fitting plane for all the observed points (remember a homography transfers points on a world plane between images), and so if the image is then warped to make the found plane exhibit no parallax, perspective effects should be reduced. Assuming the point matches have been identified subject to a Gaussian distributed error, the following least-squares criterion should be minimised for n points:

$$\min_H \sum_{i=1}^{n} d_E^2 \left(\mathbf{x}_i, \left([\mathbf{e}']_\times F - \mathbf{e}'\mathbf{a}^T\right)\mathbf{x}'_i\right)$$

where d_E is Euclidean distance and the compatible homography is parameterised as in equation 9.1. Replacing Euclidean distance with algebraic distance, and $[\mathbf{e}']_\times F$ with H, the result is two linear equations $k \in (1,2)$ in terms of $\mathbf{a}$ per point match $\mathbf{x} \leftrightarrow \mathbf{x}'$:

$$\left(\mathbf{x}'_k\mathbf{e}'_3 - \mathbf{e}'_k\right)\mathbf{x}^T\mathbf{a} = \left(\mathbf{x}'_k\mathbf{h}_3^T - \mathbf{h}_k\right)\mathbf{x}$$

where subscripts indicate the nth item in a vector and $\mathbf{h}_n$ the nth row of H. Stacking these equations gives a linear system of the form $X\mathbf{a} = \mathbf{b}$ which can be solved using any standard linear least-squares technique.

Whilst this linear algorithm is effective, it does come with the major problem that, even if the matches conform to the epipolar geometry, they can still be incorrect because the epipolar geometry only constrains matches to lie on a line. Consequently, some form of robust solution must be found. One approach is to use a random sampling method to minimise a robust Huber function $\rho(x)$ of the residuals x (σ is the robust standard deviation), i.e:

$$\rho(x) = \begin{cases} x^2 & x^2 < 3.84\sigma \\ 3.84\sigma & x^2 \geq 3.84\sigma \end{cases} \qquad \sigma = 1.4826\left[1 + \frac{5}{n-p}\right]\underset{i}{\text{median}}\,|r_i|$$

for n observations and a parameter space of dimension p, $p = 3$ for this case (see [RL87] for full details). This can be achieved by taking m minimal samples ($m = 300$ in my implementation) of 3 points and using them to find a solution $\mathbf{a}_n$. The solution $\mathbf{a}_n$, which

minimises the Huber function just outlined is then accepted as the best solution. Outliers are rejected using a 95% confidence limit and the robust standard deviation σ of the best solution i.e. $x^2 \geq 3.84\sigma$. Finally, the calculation can be repeated using the linear method.

Even if no point matches are outlying, the robust approach is still advisable because depth variation may mean that some points, although correctly matched will be so far off the best fit plane they will skew the selected plane very badly. This might occur if there is a very large amount of depth variation in an image or a very dominant plane. This is based on the assumption that it is desirable to find the best fit plane in general that will result in the most points matching well, and not to allow small insignificant image regions to skew the fit.

9.3.2 Unbounded Images

Since a linear transformation is being used, it is possible for the compatible homography to result in an unbounded image. This occurs in the extremely unlikely situation of the epipole in first image being infinite, and the epipole in the second image being within the image. In this case, the compatible homography will cause the epipole in the second image to be mapped to infinity, causing an unbounded second image. Fortunately, such a degeneracy is easily handled by swapping the images so that the homography maps points from image one (infinite epipole) to image two (epipole in image).

Near this degeneracy, problems will also occur with large images. In order to deal with this, the whole technique is modified by swapping the images the homography transfers between so that points are transferred to the image with an epipole closest to the image centre. For simplicity from this point onwards this swapping will be assumed and the image which has the compatible homography applied to it will be considered to be image two of the pair.

Applying to The Image

After determination, the compatible homography can be used to warp all points in the second image to the first image plane as $H\mathbf{x}'_i$, thus making all epipolar lines coincident, giving both images the same epipole and orienting the epipolar geometry. The nonlinear mapping of epipolar lines to the rectified image using polar coordinates will then be exactly the same for both images.

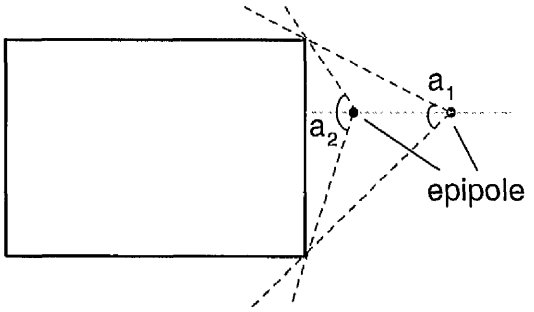

Figure 9.1: As can be seen, if the epipole is moved nearer to the image edge then the angle range covered by the image will increase. When it is on the edge it will represent a maximum angle range of π radians.

9.3.3 Rectifying the Images

After the compatible homography has been applied to one image, rectification can proceed so as to make epipolar lines parallel with the x axis. This is achieved by parameterising all image points in terms of polar coordinates centred on the epipole. Subsequently, each rectified point is described by a y coordinate given by an associated polar angle $-\pi \leq \theta \leq \pi$, and an x coordinate given by the distance of the rectified point from the epipole.

Recalling section 9.2.1, it should be noted that it is necessary to consider only positive distances from the epipoles. Points at negative and positive distances do not belong to the same half epipolar line and so cannot match to the same half epipolar line in the other image.

Before reparameterisation can proceed, it is first necessary to find the common bounds of the rectified images in polar coordinates. This amounts to identifying the range of epipolar lines common to both images as well as the maximum and minimum distance of points from the epipole for both images. Once bounded, the rectified images can then be built up line by line, with the distance between consecutive epipolar lines selected individually so as to avoid pixel compression. The output image is then created as the region that bounds the reparameterised image.

Finding the Common Region

Before finding the epipolar lines common to both images, it is best to first identify the extreme epipolar lines for both images. If it is assumed that the input images are rectangular, maximal epipolar lines are guaranteed to pass through the image corners and the angle range

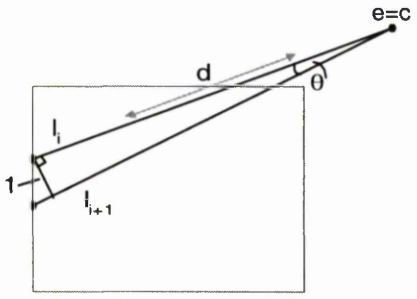

Figure 9.2: Determining the minimum distance between consecutive epipolar lines so as to avoid pixel loss

spanned by the image is guaranteed to be at most π radians (see figure 9.1). Consequently, the maximal corners can be found by determining the polar angle for each corner, and selecting the maximum and minimum corners so that the total angle range is less than π radians. Because of the restriction on the angle range, this can easily be achieved by first normalising all the angles, so that one angle is 0 radians, and then using normal minimum and maximum.

Since a compatible homography is available, all the image corners from image two can be transferred to image one prior to finding the maximal corners. The common region is then found as the second maximum and second minimum angle, such that the image spans less than π radians. In effect, this means that given minimum angles n,n' and maximum angles x,x' for both images, the second minimum sn and maximum sx are given by:

$$sn = \begin{cases} MIN(n,n') & \|n-n'\| > \pi \\ MAX(n,n') & \|n-n'\| <= \pi \end{cases} \qquad sx = \begin{cases} MAX(x,x') & \|x-x'\| > \pi \\ MIN(x,x') & \|x-x'\| <= \pi \end{cases}$$

This scheme will fail if an epipole is within an image, because in that case the relevant image will cover an angle range of 2π radians. Fortunately, this is an easily handled anomaly. If the epipole is within one image, then minimum and maximum angles can simply be set to the bounds of the other image. If the epipole is within both images, the maximum and minimum angles can be set to $-\pi$ and π.

Selecting the Epipolar Lines To Rectify

The next step is to build a table that will be used to transfer epipolar lines to and from different scan lines in the rectified image. To do this, the process starts from one extreme

epipolar line, assigns it the rectified line $y = 0$ and associates it with the relevant angle. Subsequent epipolar lines $y = n$ are then found by taking a small angle step from the previous epipolar line so that there is no pixel compression within the region of the epipolar line intersecting the image. The worst case pixel will always be situated at the furthest distance from the epipole, i.e. the image edge opposite to the epipole. Figure 9.2 shows how the angle step θ can be calculated very simply as $\theta = arctan\left(\frac{1}{d}\right)$ where d can be found by intersecting the epipolar line with the image.

Note that when this table is built up, each epipolar line from both images can be unrectified and intersected with the image. From this, the maximum and minimum distance from the epipole can be found. Subsequently, the maximum and minimum distance anywhere in both images can be found, and both output images completly bounded.

9.3.4 Rectifying and Unrectifying points

Unfortunately, avoiding pixel compression means it is necessary to rectify and unrectify points using look-up tables. Each y coordinate of the rectified image can be associated with an angle to enable unrectification, and vice-versa to enable rectification. However, this makes it difficult to rectify any point and to unrectify points with a subpixel y coordinate. In order to perform such operations, it is necessary to interpolate the look-up tables. For this reason, it is best to represent unrectified epipolar lines by their polar angles and interpolate the angles.

9.3.5 Resampling the Image

The image can be resampled very efficiently. For each line of the first image, the maximum and minimum distance of points from the epipoles can be unrectified, to give an epipolar line segment. Pixel sized steps from one extent of this segment to the other can be taken, and the output row of the rectified image can be built up. The same scheme can be used for the other image, but with the different maximum and minimum distances for the relevant image. Then, as the epipolar line in image one is being worked along in pixel sized steps, it is transferred into image two using the compatible homography.

9.3.6 Infinite Epipoles

Images with infinite epipoles will in fact not work with the above technique because all distances will become infinite, and all angles will be the same. Fortunately, they can easily be detected as an image with an effective angle range of 0 radians. Note that an infinite epipole in the second image is irrelevant because the use of a compatible homography means that points from the second image are transferred into the first image. Consequently, only an infinite epipole in the first image need be detected. For this case, the rectification can simply apply the compatible homography and rotate both images so that the epipole lies on the x axis. Although this is an exception case, it is extremely easy to detect and handle.

9.4 Examples

Figures 9.3 and 9.4 give a qualitative feel for the effects of rectification on some example images. For all the scenes, the fundamental matrix was first estimated using the techniques of chapter 4.

Figure 9.3 illustrates the rectification of an image pair produced with a camera undergoing a mainly forward movement, the sort that would result in an unbounded image if planar rectification were used. Notice how the epipole has been mapped to a line and how this has resulted in disparities which are purely horizontal making this a usable rectification. The other stereo pair in figure 9.4 was taken using a more conventional near parallel camera movement and can be seen to have produced very little image distortion.

9.5 Conclusion

This chapter has presented a simple and fast algorithm for rectification of any stereo image pair without the need for any calibration. Compared to previous techniques, it has reduced inter image distortion caused by the nonlinear rectification as well as providing a simpler and more complete set of implementation details. It has also provided a means of minimising image distortion due to perspective effects if some pre-determined matches are provided (such as might have been used to calculate the epipolar geometry). Overall, the result is a more general technique that produces images that are easier to match.

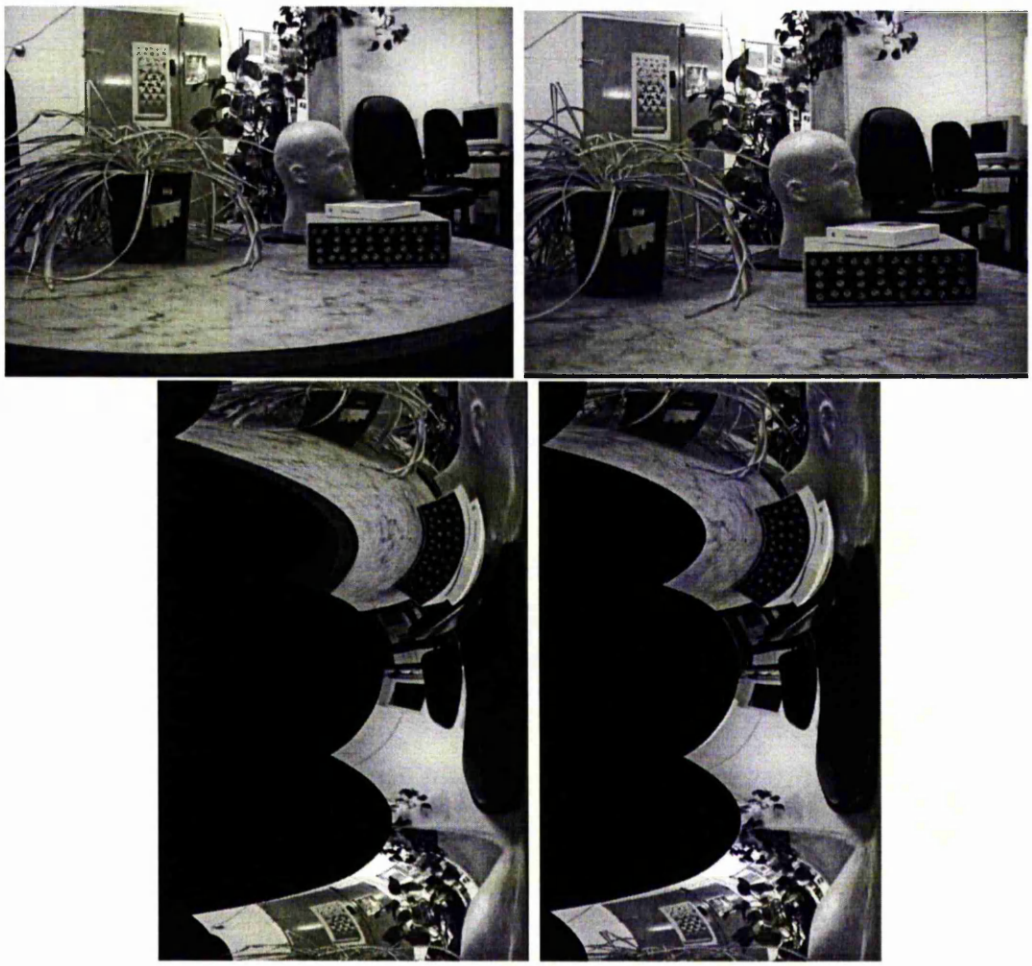

Figure 9.3: Forward movement image pair before (top) and after (bottom) rectification

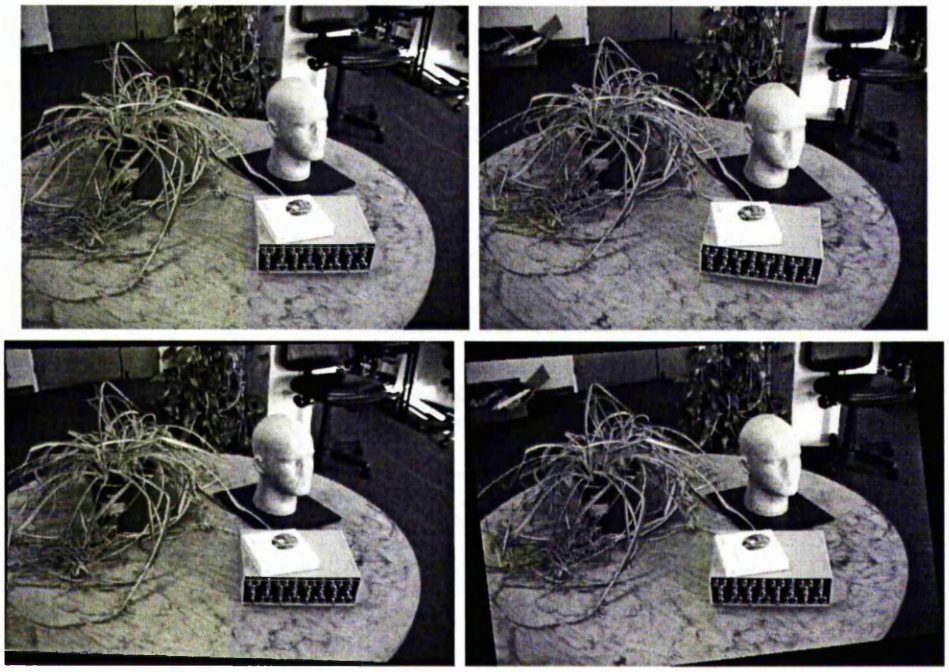

Figure 9.4: Near parallel image pair before (top) and after (bottom) rectification

Chapter 10

Models from Video Sequences

10.1 Introduction

This chapter will attempt to gather together the fairly varied collection of techniques presented in the rest of this work and combine them with existing state of the art techniques to create a complete reconstruction system. This system is general, in that it can take any video sequence without any real restrictions on the form except that certain degeneracies be avoided. These degeneracies are primarily allowing a single plane to fill the view which causes problems with projective reconstruction, and orbital motion which causes the self-calibration to fail.

10.2 Overview

The system takes as input a complete video sequence. In theory, any camera that conforms well to the full perspective model could be used for this purpose provided that the aspect ratio is close to one and there is little or no skew. The reconstruction then proceeds in a number of nicely separate modular stages:

- *Matching:* To start the whole process off, matching is performed between each consecutive image to produce pairwise sets of matches. During this process, an effort is made to match existing features already matched across the previous pair. The result of this is a large number of features many of which will be tracked across a great many images.

- *Image Selection:* In order to deal with the potentially inhibiting number of images and to avoid degeneracy, overlapping pairs of images are selected so as to minimise degeneracy and maximise accuracy. This sparse image sequence then forms the basis for the reconstruction.

- *Structure and Motion:* The image pairings are then used to build a set of reconstructions, one for each consecutive pair of images. These are robustly merged with further point matching to produce a set of projective cameras and a sparse collection of structure points.

- *Self-Calibration:* The next stage attempts to self calibrate the camera. This is achieved using natural constraints as well as certain assumptions about the camera, for example that it has zero skew and principal point in the centre of the image. At the end of this process, all structure and cameras are upgraded to a metric form.

- *Dense Correspondence:* At this stage, an effort is made to match every point in each image with the consecutive image. These correspondences are performed for every consecutive pair in the sparse image sequence and then chained together to produce large sets of points tracked for many images.

- *Model Construction:* The points from the dense correspondence are used to create depth maps (containing the distance of each point from the camera) which are then projected into the images and used to triangulate a mesh. Due to time constraints this part of the system is still a little primitive.

10.3 The Complete System

Now an overview has been given, it is appropriate to give a more detailed description of each of the stages involved in the reconstruction.

10.3.1 Matching

Before any reconstruction can begin, it is first necessary to find the same world feature in different images. In this system, point features only are matched between consecutive images and chained together to produce longer tracks. The process works sequentially from the first image to the last image and involves two main steps. Firstly, images are rejected

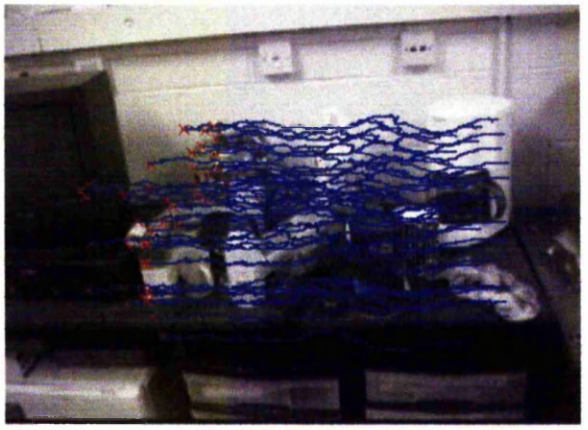

Figure 10.1: An example of feature tracking. The crosses indicate features in the image and the blue lines the trail taken by the points over the previous 120 images.

totally if the consecutive pairing is too similar and then point features are matched between consecutive images.

Image Rejection

Because a hand-held video sequence is being used, it is frequently the case that consecutive images show so little difference that there is no point in even trying to match between them. If this is found to be true then one of the two images can safely be permanently discarded and further processing proceed without it.

Detecting such similarity is a very simple and efficient process. Basically normalised zero mean cross correlation with a large window size (21x21 in my implementation) is used to obtain a correlation score between every pixel (x, y) in image 1 and the corresponding pixel at (x, y) in image 2. The average correlation score is taken from these and if it is found to be greater than some threshold (0.97 in my implementation) the second image of the pair is rejected. This can be achieved extremely efficiently [Sun97] using box filtering [McD81] to perform the cross correlation as described in appendix F.

Point Tracking

The point tracking algorithm of chapter 8 is next used to track points across as many images as possible. To avoid repetition the discussion of the point matching method used in this system (due to [TS94]) will not be repeated. The basic method finds features in the first image which are likely to be easy to match, and then attempts to track these into the second

image. New features are then found in the second image to replace any that could not be tracked and the second image becomes the new first image with the whole process repeating until there are no more images.

Figure 10.1 shows an example set of tracks, these are tracks across 120 images with the trails representing the path of the points. Note that this is purely the response of the feature tracker and includes no outlier rejection.

10.3.2 Image Selection

Because a video sequence can contain many thousands of images, the next stage selects a smaller set of images covering the whole sequence. This serves a two fold function, firstly to reduce the computational load and secondly to allow selection of images so that the camera motion between the images is sufficiently large and non degenerate to allow accurate camera determination. Without this stage, geometry estimation can still produce reconstructions with good re-projection error, but the reconstructions will usually be sufficiently inaccurate to make later stages (particularly self-calibration) fail.

The image selection is performed using the techniques of chapter 8, and attempts to find pairings of images that minimise a certain similarity score based on maximising the number of matches and minimising both the reconstruction error and degeneracy. The result is a sparse set of images and the pairwise geometry of these images in the form of the fundamental matrix.

10.3.3 Structure and Motion

Now a sparse set of images and the associated pairwise image relations are available, they are used to build a set of reconstructions, one for each consecutive pair of images. These are then robustly merged with outlier removal and further point matching to produce a set of projective cameras and a sparse collection of structure points.

This topic was treated extensively in chapter 7, and so only the specific solution used will be detailed here. Starting from the selected image pairs, the fundamental matrix governing the geometry of each pair is used to produce a reconstruction using the techniques of chapter 5. For best results a bundle adjustment is run on the pairwise reconstructions.

Each consecutive pair of images is then robustly merged to produce a reconstruction for the resultant triplet of images. Pairs of triplets are then merged robustly so as to keep two

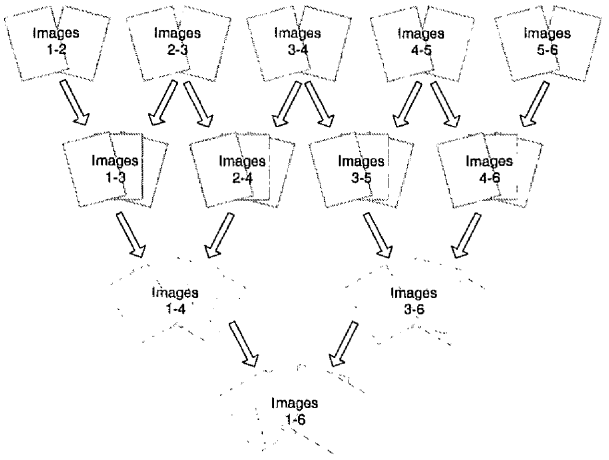

Figure 10.2: Hierarchical merging used for reconstruction

images overlapping at all times and the merging repeats hierarchically as shown in figure 10.2. A bundle adjustment is performed after every merge.

To handle extremely large sequences in a computationally efficient manner, if the number of images in a sub-sequence gets above 60 then the merging proceeds sequentially instead of hierarchically. Bundle adjustment is not performed during the sequential merging, but is delayed until the complete sequence is available.

Throughout the merging process, a continual effort is made to find more matches. After each merge, this will take all points in one sub-sequence that are not already tracked into the other sub-sequence and attempt to find matches for them using the newly calculated structure and cameras. This takes the form of a guided search as described in section 8.6, page 184, and although not essential can help a great deal if parts of the scene go in and out of view either due to occlusion or returning to view something seen earlier. For very long sequences the extra matching is highly advisable.

10.3.4 Self-Calibration

The next stage attempts to self calibrate the sparse cameras. This is achieved by using natural constraints to relate the equivalent metric cameras to the projective ones. In particular the constraint that the absolute conic (see section 2.4.4, page 39) remains invariant to the Euclidean component of a camera matrix.

If this constraint is used in combination with certain assumptions about the camera, such as zero skew and principal point at the centre of the image then linear constraints can

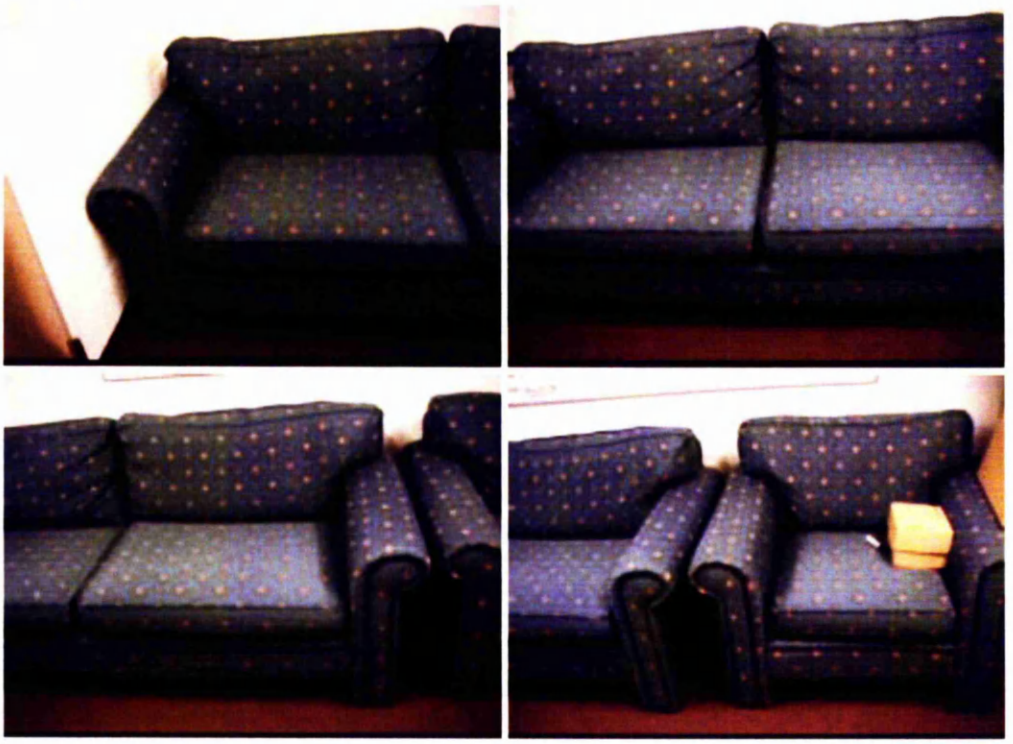

Figure 10.3: Selected images from the sofa sequence of 327 images (includes first and last image).

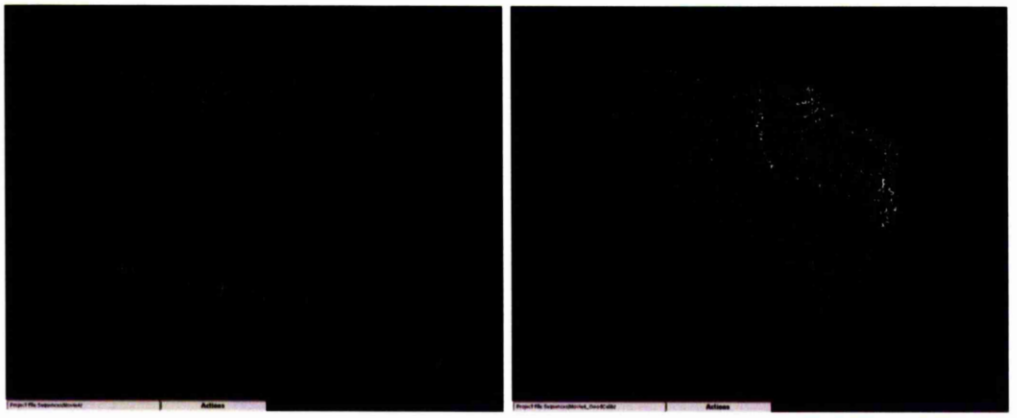

Figure 10.4: Two views of the calibrated sparse set of points and cameras generated from the sofa scene. Cameras are marked with cones. Note how the one camera has been incorrectly reconstructed due to a trade off between focal length and movement along the optical axis.

Figure 10.5: Selected images from the box sequence. This includes the first and last images.

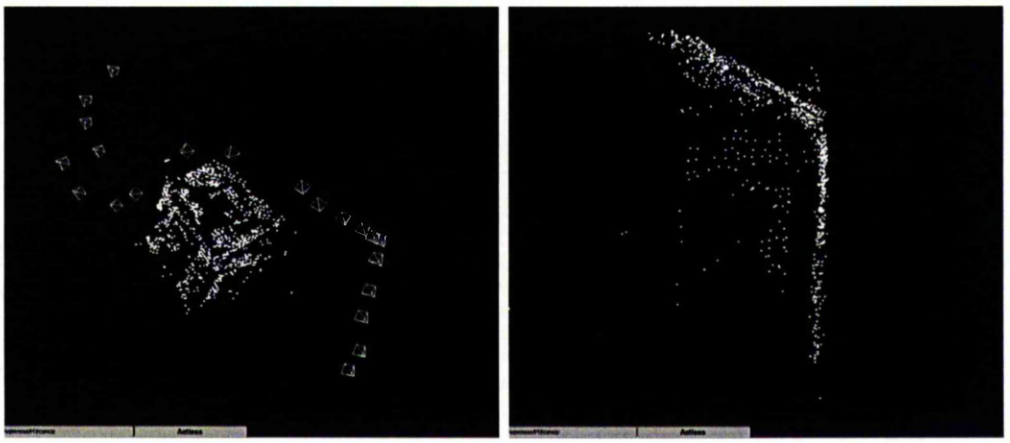

Figure 10.6: Two views of the sparse set of calibrated points and cameras generated from the box scene.

be found on the projective form of the absolute conic. Once the location of the absolute conic has been found in the projective reconstruction it can be reduced to canonical form, upgrading the structure and cameras to a metric form. This is achieved using the entirely automatic algorithm of [PKG97, Pol99].

Due to space constraints the self-calibration algorithm will not be detailed here because it does not represent an original contribution of this work. For the interested reader, appendix E gives a detailed description of the algorithm as well as a little background on the self-calibration problem.

Figures 10.4 and 10.6 show novel views of the sparse points and cameras obtained after self-calibration for the two video sequences shown in figures 10.3 and 10.5 respectively. It can be seen that the points are relatively noise free and accurately determined as are the cameras. Note that the two video sequences shown in figures 10.3 and 10.5 cover the complete video sequence, and include first and last frames. A more complete set of images from the sofa sequence can be found in appendix G.

Practical Notes

Self-calibration is far from being an ideally solved problem. There exist numerous degeneracies for the form of self-calibration used here. See [Stu97] for a complete analysis for constant intrinsic parameters and [Stu99, Pol99] for varying focal length. More recent work on degeneracy can be found in [Kah99]. The most common form of degeneracy likely to dominate a large sequence is orbital motion or planar scenes. The algorithm also requires a

very good reconstruction.

Personal experience with non degenerate sequences has drawn to light two main problems with self-calibration. The first of these is a tendency to exchange changes of focal length with movement along the optical axis. This is probably because a change of focal length differs only by minor second order differences from a movement along the optical axis. Bearing all this in mind, it probably explains the success of self-calibration methods which allow focal lengths to vary. Such approaches can account for the aforementioned errors without damaging the reconstruction of the plane at infinity.

In the projective reconstruction, it follows that this trade off of movement against focal length is caused by inaccurate reconstruction of epipoles (i.e. camera centres). The frame selection method outlined in section 10.3.2 can help a great deal with this problem, as can a constrained bundle adjustment enforcing consistent focal lengths (but this is at the cost of allowing focal length to vary). Figure 10.4 illustrates an example failure, where the far right camera should be at a roughly similar distance from the sofa but clearly is not. In this example, because it is viewed from a novel point and the structure still looks metric, the plane at infinity has been correctly located even if the cameras have not. It follows that allowing varying focal lengths has saved the metric properties of the reconstruction.

The second major problem with self-calibration is projective drift. This is more prevalent in very long sequences in which later images do not contain features tracked from earlier images. The result is a tendency of the projective coordinate frame to drift slightly, meaning that there will no longer be a consistent plane at infinity for the self-calibration. Similar drift has been observed in closed sequences (see [FZ98b]) where the last and first camera positions should be the same, but are not in the reconstruction.

This problem can be significantly alleviated by using the merging based approach to reconstruction presented in this work which provides better balancing of error. However, it is by no means solved and to cope with it in a fail safe manner it would be necessary to eliminate the projective drift entirely.

10.3.5 Dense Correspondence

The reconstruction thus far has only made use of a sparse set of easily matchable points. At this stage, an effort is made to match every point in each image with the consecutive image. The same sparse image set used for structure and motion calculation is used for this purpose. The dense set of pairwise matches this produces are then robustly combined to

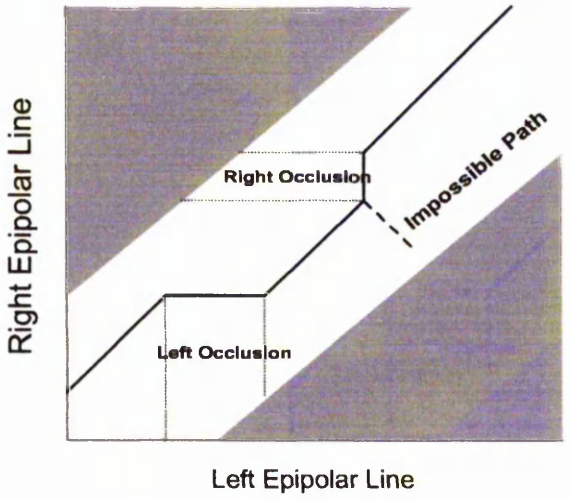

Figure 10.7: Dense correspondence as a path search

Figure 10.8: Disparity map (left) generated from the pentagon image pair (right).

create longer more reliable tracks.

Rectification

Perhaps the most important constraint that can be imposed when attempting to match between image pairs is that offered by the epipolar geometry. This can imposed in a computationally efficient manner by first rectifying the pairs of images. To this end, the completely general rectification method of chapter 9 is employed to make all matching epipolar lines coincident and parallel to the x axis. The advantages of using this particular rectification method are that it is applicable to any epipolar geometry and that unlike other general methods it minimises distortion due to perspective effects.

Figure 10.9: Depth map (left) generated from the pair of images (right) of a cluttered desk. The depth map is from the point of view of the middle image.

Figure 10.10: Depth map (left) generated from the pair of images (right) of a sofa and chair. The depth map is from the point of view of the middle image.

Dense Correspondence for Image Pairs

The approach to pairwise dense correspondence taken here is an extended version of the method presented in [CHRM96]. This method attempts to match individual pixels subject to extra constraints in addition to that offered by the epipolar geometry. In particular, uniqueness is enforced so that given a match in one image it may only match one point in the other image. As a by product of this, the so called ordering constraint is also enforced, so that given a particular match at (x, y) in image 1, the next match at $(x + 1, y)$ must either be occluded or at a greater disparity. If it is at less disparity then it will match something that could have been matched to earlier in the scan line.

The original method in [CHRM96] matches on a per pixel basis, but to increase robustness the starting point for the implementation presented here was a normalised zero mean cross correlation (as in [Fal94]). This 'box matching' is performed between all points (x, y) in the first image and all points $(x + d, y)$ at a certain disparity d in the second image. Such matching can be performed efficiently using a box filter (see appendix F) to perform all correlations for a range of disparities $d_{min} \leq d \leq d_{max}$. When colour images are available further improvements were obtained by summing the correlations for each of the red, green and blue channels. Similar colour based improvements to block matching have been used before [Kos93].

Matching is then performed so as to match each pixel in the first image with a corresponding pixel in the second image which maximises the correlation score. This is re-cast as a path search problem for each epipolar line, so that the ordering and uniqueness constraints can be imposed (as in [CHRM96]).

Figure 10.7 illustrates this path search problem. The white band in this figure indicates the region of potential matches (defined by the disparity range $d_{min} \leq d \leq d_{max}$), and the line through this region, an example path built from left to right. The constraints are enforced by insisting that only three types of step may be taken to create a path. This can be a diagonal step corresponding to a match, a horizontal step corresponding to an occlusion in the left image or a vertical step corresponding to an occlusion in the right image. A step along the path as indicated by the dotted line would result in potentially non-unique matches and so is not allowed.

Each type of step is assigned a particular cost. A step associated with a match is assigned a cost the same as the correlation score and a step associated with an occlusion is assigned a certain fixed cost. Dynamic programming is then used to find the path with the highest

score.

To increase efficiency and improve the accuracy a pyramid scheme was used ([Fal97]). In this scheme both images are low pass filtered and then scaled down by a power of 2. Matching is then performed at the lower resolution, and the results scaled and interpolated so as to restrict the matching at the next level of resolution. By using the same correlation window size at each resolution the matching window effectively decreases in size, allowing a larger window matching to provide an initial guess before refinement of finer details with less accurate and smaller correlation windows.

When matching is complete, the resulting disparity maps are median filtered so as to remove isolated points and to smooth the disparity map. Some example disparity maps and depth maps can be seen in figures 10.8 to 10.10, with occlusions and unmatched areas marked in red.

The first figure, 10.8 shows a disparity map for a fairly standard and accurate test pair of images often used to asses correspondence algorithms. The next figure 10.9 shows a depth map for a pair of very detailed images and illustrates the sort of scene for which dense correspondence is not ideal due to the large number of distinct depth changes (at object boundaries). The final figure 10.10 shows a much more suitable scene without the sharp edges.

Dense Correspondence for Multiple Views

The pairwise disparity maps are usually fairly inaccurate because the epipolar constraint only constrains the match for a point to lie on lines. To help overcome this problem and to greatly increase the accuracy, matches from the disparity maps are chained together to create longer matches.

A different approach was taken to previous methods ([KPG98]). Rather than convert all disparity maps into depth maps (i.e. distance from the camera rather than disparity) and then refine these depth maps by linking up and down using the matches, an effort is made to reconstruct a set of points from all the disparity maps.

To do this, a set of tracks is initialised using the first disparity map. Using each subsequent disparity map the tracks are extended by rectifying the point from the second image of the last pair into the first image of the new pair and using the interpolated disparity value. At each stage a new set of tracks are added for any points which were not used in extending previous tracks. All points are constantly reconstructed and then tracking stopped if their

re-projection error falls outside a confidence limit. Rejected points are added to the complete set of points only if they have been tracked for 4 or more images.

The advantage of this approach is that it is viewpoint independent and does not require calibration. However, it can be slower and does produce data which is harder to work with.

10.3.6 Model Construction

To produce a model for viewing purposes, the large point set obtained from dense correspondence is projected into a particular image to produce a depth map. The resultant depth map is then smoothed using a median filter that preserves edges (a cross shaped window rather than a box shaped window) and small gaps in the depth map filled by interpolation.

A triangular mesh is then fitted to the depth map, by placing large squares of a certain edge size (e.g. 6 pixels) over the depth map. These squares are then halved into triangles and the depth map used to compute the 3D coordinates of the vertices. To prevent problems with blurring of edges, if a triangle is found to have a surface normal pointing away from the camera it is split into two until this is not the case. Any triangles of unit edge size that still point away are removed entirely.

This method whilst fairly primitive can produce visually pleasing results, particularly if the resultant models are texture mapped as shown in figures 10.11 and 10.12. These two particular models have been selected since figure 10.11 contains only nice smooth objects and is ideal for this approach to model construction, whereas figure 10.12 is a low quality greyscale sequence with many distinct edges which illustrates the ineffectiveness of dense correspondence on low quality images or at obtaining accurate edges.

10.4 Further Improvements

Due to the limited time available for this project much has not been addressed. In particular, there are many existing improvements to the techniques used which could be added to the system as well as other problems which yet need to be addressed. A brief summary of some of these are given here.

10.4.1 Feature Matching

It is fairly safe to say that point matching for video sequences is to all intensive purposes practically solved. The system presented in this work as well as previous systems have

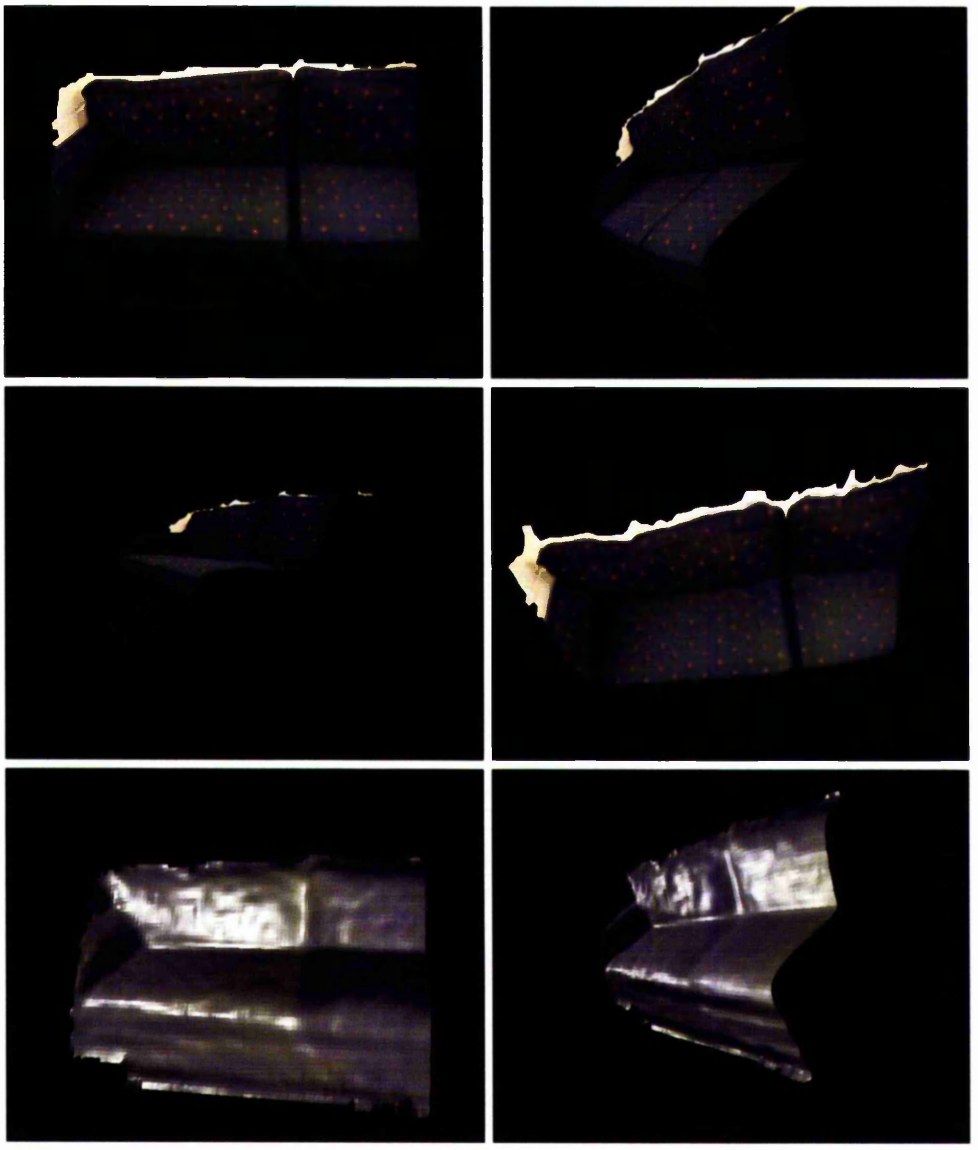

Figure 10.11: Four textured novel views (top) of the model generated from the sofa sequence and two un-textured (bottom). This sequence is well suited to the dense correspondence process and looks good even from very different viewpoints far from those visible in the sequence. A comprehensive selection of the images from the sofa sequence can be found in appendix G.

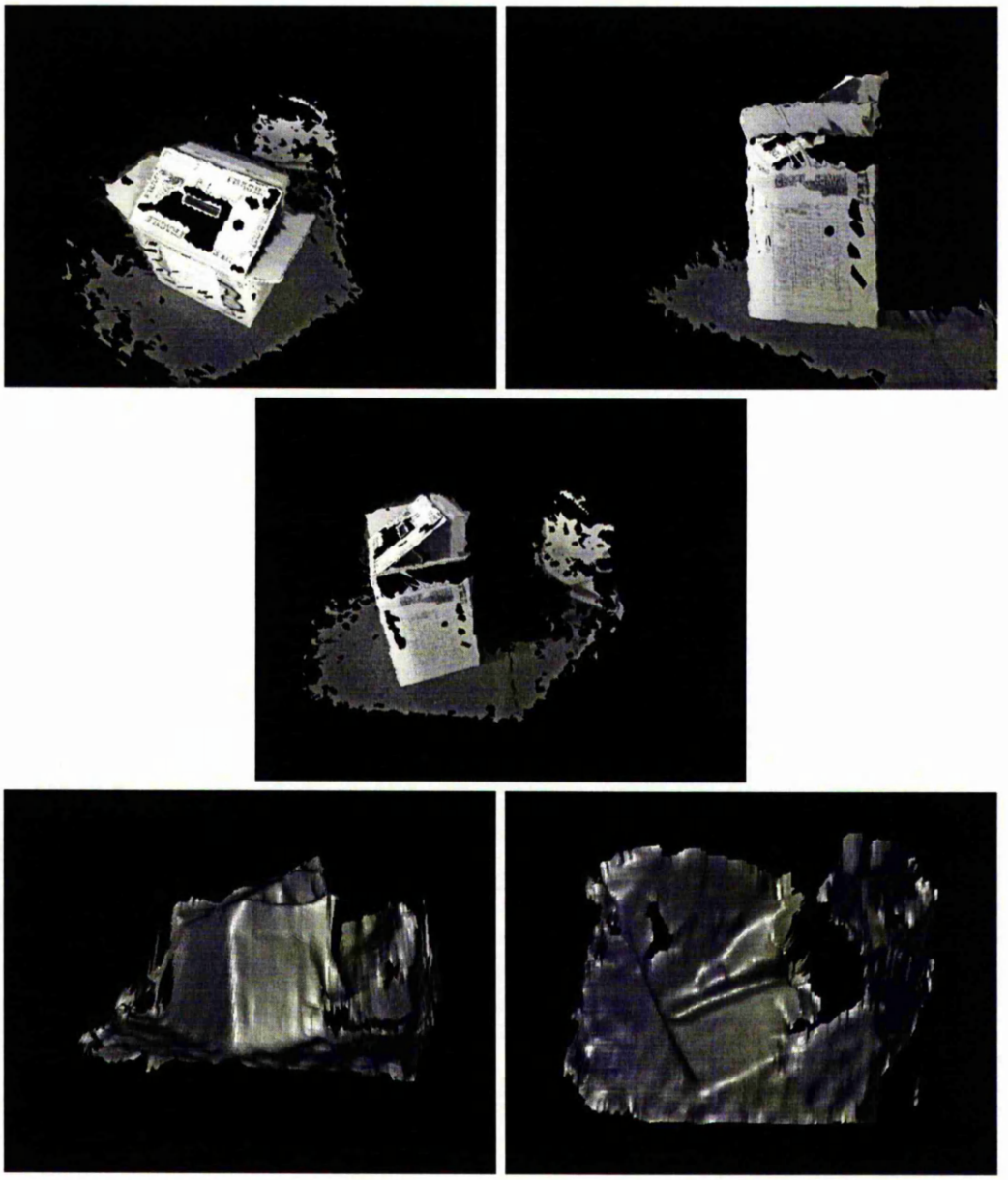

Figure 10.12: Three textured novel views (top) of the model generated from the box sequence and two un-textured (bottom). Note how the sharp edges are not recovered especially accurately and how the lack of texture in the images (particular for the floor) has resulted in large sections not being recovered. In particular notice the incorrect reconstruction of the empty space under the lid of the box. A sample of the images used to generate this sequence can be found in figure 10.5.

illustrated the suitability and generality of points for reconstruction and tracking. However, there is still plenty of room for improvement by extending the system to use lines [CCZ96], curves [SZ00] or even conics and edges. Not only can such features be very accurately located, but they can be very useful for identifying object boundaries when constructing models.

10.4.2 Degeneracies for Structure and Motion

Recalling section 8.1.3, page 165 there are two critical forms of motion, rotation and imaging planar scenes which will not allow determination of a camera. Since detection and handling degeneracy due to rotation is accounted for in the system developed for this work, only planar motion remains to be handled. This is in theory possible, since as discussed in section 8.1.3, page 166 a camera imaging a planar scene can be reconstructed if the structure in the scene is metric.

Fortunately self-calibration algorithms for planar scenes do exist and techniques are available for fixed [Tri98, MC00b] or varying [MC00a] focal lengths. Consequently, calibration of the camera in the planar section can be used to determined cameras and structure using the techniques in [Fau93]. To do this requires at the least 3 views for a very restricted camera model or 4 if the focal length is allowed to vary. Since video sequences are being used it is extremely unlikely that a plane which dominated the view in at least one image cannot be tracked for many more than 4 images.

Detection of planar motion is already included in the system, and so to deal with it the only necessary extension would be the addition of the self-calibration and reconstruction algorithms. For greater robustness it would also be a good idea to implement a homography based feature tracker suitable for tracking points on the plane causing the degeneracy for as many images as possible. Naturally, this discussion does rely on the relatively untested effectiveness of the planar self-calibration algorithms to be feasible.

10.4.3 Self-Calibration

Whilst projective reconstruction can be achieved very reliably for very long video sequences, self-calibration is still fairly unreliable. The use of the image selection system and new merging based projective reconstruction proposed in this work enables sufficiently good reconstruction for self-calibration to work most of the time. However, it is still not very reliable for very large sequences and is very unstable. In particular, it is prone to replacing camera movement along the optical axis with changes in focal length as well as to projective drift.

It is likely much could be done to alleviate these problems. In particular, allowing the user to provide certain scene constraints such as parallel lines, vanishing points and known structure or angles should the automatic methods fail. It also seems likely that further work on self-calibration to detect and capitalise on degeneracies (such as pure rotation) could yield some benefits. It would also be interesting to integrate the self-calibration into the hierarchical scheme so that when possible, merging could be performed between two calibrated sequences. By making the reconstructions metric as early as possible problems of projective drift could be dealt with because the reconstruction would be metric and so could not drift in a projective manner.

10.4.4 Dense Correspondence

Dense correspondence can still be very unreliable for many image pairs. In particular, images with lots of depth variation or little texturing rarely work well, if at all. It seems likely that much more could be achieved along the lines of dense correspondence. The implementation used in this work is not completely state of the art, and there do exist further slight improvements to the type of approach, particularly the disparity map interpolation offered in [Fal94]. This method attempts to make disparity edges and luminance edges line up and would greatly reduce the blurring of edges (corona effect) due to the use of block based disparity estimation. Also of use would be the addition of the assumption that depth discontinuities are associated with intensity discontinuities [BT98] that would allow untextured areas to be handled.

Very good results in dense correspondence have also been obtained by performing matching using many images (e.g. [CHRM96, NMSO96]). Although this needs to be performed on a per pixel basis, using as many as 19 or 20 images has been shown to produce very high accuracy. It would be interesting to pursue this approach since suitable images can usually be obtained from a video sequence, and it is likely results would be better than imposing the consistency after matching by linking the disparity maps.

One approach of interest is to extend dense correspondence so that it uses correlation scores from image triplets rather than pairs to initialise the correspondence. This is possible when using planar rectification because all points of the same disparity belong to the same plane. Consequently, the third image can be warped by a homography for each depth to align this plane and the correspondence performed. This is still possible for general camera motions because the method presented in this work maintains this property (unlike previous

general methods). However, it is further complicated because each scanline needs to be transformed separately. Unfortunately due to time constraints there was no chance to finish pursuing this (except the new rectification algorithm).

10.4.5 Model Construction

Since the model construction method in this work is so simple there is doubtlessly much room for improvement on this front. For example, attempting to fit thin plate splines to create surfaces and then matching these surfaces between images to get more accurate positioning. Also, textures could be extracted to sub pixel accuracy by interpolating the visible texture regions in all images

It would also be interesting to attempt to extend the method to be less viewpoint dependent. Although this could be achieved by producing a set of models for each viewpoint and then merging them, it is probably not the best approach. Such a method would be highly redundant and so instead it would probably be better to produce a model using one viewpoint then project it into another view point, fill in the gaps and refine where possible, then project into yet another view point. Provided care is taken to detect one surface overlaying another this would probably be far simpler.

Chapter 11

Conclusion

11.1 Summary

This work has developed a complete system for producing models from video sequences, with emphasis on enhancing certain aspects of the reconstruction process.

Firstly, problems of effective image acquisition were resolved by a method to select suitable frames for reconstruction from a video sequence. The use of a video sequence makes feature matching very easy and accurate as well as ensuring a good distribution of images are available. This approach avoids problems due to the overwhelming quantities of data and serious inaccuracies caused by geometry estimation for small baselines.

Whilst feature matching is a very difficult problem it can provide only a starting point for reconstruction. At the heart of most uncalibrated reconstruction systems is a method for projective reconstruction. Although existing approaches to projective reconstruction are very effective, this work has further extended them to produce significant improvements. New robust merging based algorithms were proposed and shown on both synthetic and real data to produce large improvements over existing methods.

Finally, these and numerous other improvements were incorporated into a complete system for 3D reconstruction from video sequences. This entirely automatic system is capable of producing realistic models with no human interaction. It should be noted that this system also serves to demonstrate the effectiveness of the new techniques, and is arguably not a major contribution in itself (models have been produced from images using dense correspondance techniques for decades).

The modular nature of the whole reconstruction process means that all the contributions of the work are generally applicable. For example, the video frame selection process does

not have to be applied to projective reconstruction, and rectification has many uses other than dense correspondence.

11.2 Discussion

Whilst totally automated model building does now seem possible, there are still many limitations and practical problems. The process can benefit a great deal from user interaction and further improvements to the techniques. Due to the many and varied aspects of reconstruction a discussion relating to each of the main areas will now be given:

11.2.1 Feature Matching

Point matching can now be considered a practically solved problem, particularly for video sequences. With video sequences, it has long been known that the very small difference between frames allows matching to be performed using a very simple model of image motion. This is not true for sparse matching (i.e. a collection of images separated by arbitrary and larger baselines), where a more complete motion model needs to be used to guide the matching process.

Experience with this work has led me to believe that a very great deal of complexity can be avoided by using video sequences instead of sparse image sets. The reasons for this lie in the feature tracking. Tracking features using very simple motion models avoids a great deal of the complexity required for geometry guided matching. Primarily, it avoids the need for model selection (i.e. selection between homographies and fundamental matrices), avoids problems with low frequency repeated structure, simplifies image acquisition, and greatly increases the accuracy and length of tracks.

Combining these simplifications with the image selection process results in an extremely effective and robust method. In my experience, unless effort was made to introduce planar degeneracies, projective reconstruction never failed to produce extremely good results even with very poor quality sequences. However, all this is not to say that further improvements are not still possible, particularly in handling and even capitalising on degeneracies (e.g. tracking with a homography), in matching other forms of structure such as lines, conics or even edges or even by using different robust methods (for example, see [LPT00] where a simplex based approach is shown to be more effective than RANSAC).

11.2.2 Structure and Motion

The problem of producing cameras and structure is a very long-standing problem that has received a great deal of attention. There is no one ideal solution, whether working with calibrated or uncalibrated cameras, but instead a plethora of techniques from which the most appropriate must be selected for the task in hand. This work has successfully generalised a method of reconstruction so that it can be adapted to most problems that require a full projective camera model.

In my view, the application of projective geometry to modelling the image formation process presents an effective way of approaching the reconstruction problem when only projective (or image based) concepts are required (e.g. for point matching or rectification). It allows all the often unnecessary constraints required to make a space metric to be ignored (e.g. rotation matrices), and greatly simplifies both the theoretical and practical application of geometry to the reconstruction problem. It is simpler to consider only those properties of the object space that have been preserved in the image than to add in all the details that have been lost. This usually results in increased accuracy and ease of computation.

Because of the increase in accuracy, even in situations where camera calibration is to be performed, a projective interpretation can prove very useful for stages where calibration is not needed (e.g. to match points). Even if calibration is required at some stage, projective reconstruction can provide a useful means of boot strapping the whole process or verifying the accuracy of the calibrated results.

However, as will be discussed in the next section, the lack of effective calibration algorithms for upgrading projective structure to metric makes general use of projective reconstruction very difficult. Good results have been achieved producing metric reconstructions directly using only a very rough approximation of camera calibration (a guess) as a starting point (e.g. [TM91, LTCP01]). This consideration means that in many cases it is not worth attacking the unusal and un-intuitive (it is however mathematially simpler) projective approach since it offers little practical benefit in situations where metric reconstructions are required.

11.2.3 Self-Calibration

Despite extensive work on the problem of self-calibration, a robust and practical solution remains to be found for upgrading projective structure to metric. Although very effective solutions have been obtained for specialised situations such as stereo rigs or restricted camera

of camera self-calibration is now very well understood, although it is still likely that some improvements will be found by addressing it further.

However, if pure self-calibration is ever to become generally reliable, it is likely that a more pragmatic approach will probably be necessary that focuses more on the where and when of applying the calibration rather than the calibration method itself. For example, the idea of including self-calibration into the merging based reconstruction so as to avoid problems of projective drift, or of using some form of random sampling to select cameras from which to calibrate (the latter was suggested by Simon Gibson and is not the author's idea).

An alternative would not be to attempt calibration after reconstruction, but to take approximations to the camera calibration (e.g. a guess) and use this to initialise a fully metric reconstruction (e.g. [TM91, ?]). This method can be surprisingly robust, and certainly more robust than the existing general self calibration algorithms.

11.2.4 Model Building

This section covers the entire process of model building - a problem that has been resolved in this work by using a dense correspondence approach. The drawback with dense correspondence is that it provides a low degree of accuracy. It particularly tends to fail at detecting sharp edges, and suffers badly from problems with objects that are viewed at odd angles. Although the later problem can probably be dealt with by some form of registration using all views.

In the author's opinion the dense correspondence approach to modelling still needs a fair bit of work. In particular, integrating the multiple view linking into the actual matching process itself has been shown to produce dramatic improvements and would doubtlessly be invaluable. Also producing the models from the dense set of points needs a great deal of work. Ideally a method that only infers surfaces from continuous regions in images and then links these together in a viewpoint independent manner is required (e.g. [MK00] has made a start on this approach). This is probably better than building a set of models from each viewpoint, and then merging these together - an approach designed for different forms of acquisition where different data is available (e.g. volumetric data).

Perhaps it might even be possible to resolve some of the matching problems by converting the dense correspondence problem across relatively wide baselines into a tracking problem across consecutive frames of a video sequence. In this way, disparities between images can

be relied upon to be very small, and so searching greatly reduced. This seems particularly appropriate if the matching is performed across many images rather than just a pair.

Of course, dense correspondence does not present the only solution to the modelling problem. A lot could be done in producing semi automated CAD tools, which could be guided by user interaction to fit models to the images. This is an especially good idea since most of the guided user interaction could be done prior to calibration. User provided shapes such as conics and squares can then be used to greatly improve the reliability of a self-calibration.

Bibliography

[AH88] N. Ayache and C. Hansen. Rectification of images for binocular and trinocular stereovision. In *International Conference on Pattern Recognition (ICPR88)*, pages 11–16, Rome, Italy, 1988.

[AP95] A. Azarbayejani and A. Pentland. Recursive estimation of motion, structure and focal length. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(6):562–575, June 1995.

[Arm96] M. N. Armstrong. *Self-Calibration from Image Sequences.* PhD thesis, University of Oxford, 1996.

[AS98] S. Avidan and A. Shashua. Threading fundamental matrices. In *5th European Conference on Computer Vision (ECCV98).* Springer - Verlag, June 1998.

[BD94] T. Blaszka and R. Deriche. Recovering and characterizing image features using an efficient model based approach. Technical Report 2422, INRIA, November 1994.

[Bea78] P. R Beaudet. Rotationally invariant image operators. In *International Conference on Pattern Recognition (ICPR78)*, pages 579–583, 1978.

[BGK98] M. Bober, N. Georgis, and J. Kittler. On accurate and robust estimation of fundamental matrix. In *Computer Vision and Image Understanding*, volume 72, pages 39–53, 1998.

[Bro58] D. C. Brown. A solution to the general problem of multiple station analytical stereo triangulation. Technical Report 43, RCA Data Reduction Technical Report, Patrick Air Force base, Florida, 1958.

[BT98] S. Birchfield and C. Tomasi. Depth discontinuities by pixel-to-pixel stereo. In *6th International Conference on Computer Vision (ICCV98)*, pages 1073–1080, Bombay, India, 1998.

[BTZ96] P. A. Beardsley, P. H. S. Torr, and A. Zisserman. 3D model aquisition from extended image sequences. In *4th European Conference on Computer Vision (ECCV96)*, pages 683–695, 1996.

[BYX82] P. J. Burt, C. Yen, and X. Xu. Local correlation measures for motion analysis: a comparative study. In *IEEE CPRIP*, pages 269–274, 1982.

[BZM97] P. A. Beardsley, A. Zisserman, and D. W. Murray. Sequential updating of projective and affine structure from motion. *International Journal of Computer Vision*, 23(3):235–260, 1997.

[Car94] S. Carlsson. Multiple image invariance using the double algebra. In J. L Mundy, A. Zisserman, and D. Forsyth, editors, *Applications of Invariance in Computer Vision*. Springer-Verlag, 1994. Volume 825 of Lecture Notes in Computer Science.

[CB88] B. Charnley and R. Blissett. Surface reconstruction from outdoor image sequences. *Image and Vision Computing*, 6(2):87–90, February 1988.

[CCZ96] J. C. Clarke, S. Carlsson, and A. Zisserman. Detecting and tracking linear features efficiently. In *7th British Machine Vision Conference (BMVC96)*, 1996.

[CGVC00] G. Chesi, A. Garulli, A. Vicino, and R. Cipolla. On the estimation of the fundamental matrix: a convex approach to constrained least-squares. In *European Conference on Computer Vision (ECCV2000)*, 2000.

[CHRM96] I. J. Cox, S. L. Hingorani, S. B. Rao, and B. M. Maggs. A maximum likelihood stereo algorithm. *Computer Vision and Image Understanding*, 63(3):542–567, May 1996.

[Cri99] A. Criminisi. *Accurate Visual Metrology from Single and Multiple Uncalibrated Images*. PhD thesis, University of Oxford, Dept. Engineering Science, December 1999. D.Phil. thesis.

[CTB92] P. Courtney, N. A. Thacker, and C. R. Brown. A hardware architecture for image rectification and ground plane obstacle detection. In *11th International Conference on Pattern Recognition (ICPR92)*, pages IV:23–26, The Hague, Netherlands, 1992.

[CWC90] N. Cui, J. J. Weng, and P. Cohen. Extended structure and motion analysis from monocular image sequences. In *International Conference on Computer Vision (ICCV90)*, pages 222–229, Osaka, Japan, 1990.

[CZZF96] G. Csurka, C. Zeller, Z. Zhang, and O. D. Faugeras. Characterizing the uncertainty of the fundamental matrix. In *Computer Vision and Image Understanding*, volume 68, pages 18–36, October 1996.

[DF90] R. Deriche and O. D. Faugeras. 2d-curves matching using high curvature points: applications to stereovision. In *10th International Conference on Pattern Recognition (ICPR90)*, volume 1, pages 240–242, 1990.

[DF93] R. Deriche and O. D. Faugeras. A computational approach for corner and vertex detection. *International Journal of Computer Vision*, 10(2):101–124, 1993.

[DS83] J. E Dennis and R. B Schnabel. *Numerical Methods for Unconstrained Optimization and Nonlinear Equations.* Englewood Cliffs, NJ: Prentice-Hall, 1983.

[DTM96a] P.E. Debevec, C.J. Taylor, and J. Malik. Modeling and rendering architecture from photographs: A hybrid geometry and image based approach. Technical Report UCB//CSD-96-893, U.C. Berkley, CS Division, January 1996.

[DTM96b] P.E. Debevec, C.J. Taylor, and J. Malik. Modeling and rendering architecture from photographs: A hybrid geometry and image based approach. In *Proceedings ACM SIGRAPH*, pages 11–20, 1996.

[Fal94] L. Falkenhagen. Depth estimation from stereoscopic image pairs assuming piecewise continuous surfaces. In *European Workshop on Combined Real and Synthetic Image Processing for Broadcast and Video Productions*, Hamburg, Germany, 1994.

[Fal97] L. Falkenhagen. Hierarchical block-based disparity estimation considering neighborhood constraints. In *Proceedings International Workshop on SNHC and 3D Imaging*, 1997.

[Fau92] O. Faugeras. What can be seen in three dimensions with an uncalibrated stereo rig. In G. Sandini, editor, *2nd European Conference on Computer Vision (ECCV92)*, pages 563–578. Springer-Verlag, Santa Margherita Ligure, Italy, 1992.

[Fau93] O. Faugeras. *Three-Dimensional Computer Vision.* MIT Press, 1993.

[FB81] M. A. Fischler and R. C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.

[FLR+95] O. Faugeras, S. Laveau, L. Robert, G. Csurka, and C. Zeller. 3-d reconstruction of urban scenes from sequences of images. Technical Report 2572, INRIA, 1995.

[FP86] W. Förstner and A. Pertl. *Photogrametric Standard Methods and Digital Image Matching Techniques for High Precision Surface Measurements.* Elsevier Science Publications, 1986.

[FP97] O. Faugeras and T. Papadopoulo. A nonlinear method for estimating the projective geometry of three views. Technical Report 3221, INRIA, 1997.

[FQM92] O. D. Faugeras, Luong. Q., and S. Maybank. Camera self-calibration: Theory and experiments. In *2nd European Conference on Computer Vision (ECCV92)*, pages 321–334, 1992.

[FZ98a] A. W. Fitzgibbon and A. Zisserman. Automatic 3D model acquisition and generation of new images from video sequences. In *Proceedings of European Signal Processing Conference (EUSIPCO '98), Rhodes, Greece*, pages 1261–1269, 1998.

[FZ98b] A. W. Fitzgibbon and A. Zisserman. Automatic camera recovery for closed or open image sequences. In *5th European Conference on Computer Vision (ECCV98)*, pages 311–326. Springer-Verlag, June 1998.

[GHH01] S. Gibson, T. J. Howard, and R. J. Hubbold. Flexible image-based photometric reconstrucion using virtual light sources. In *Eurographics*, Manchester, UK, September 2001.

[GVL89] G. H. Golub and C. F. Van Loan. *Matrix Computations.* The John Hopkins University Press, 1989.

[HÅ97] B. A. Heyden and K. Åström. Euclidean reconstruction from image sequences with varying and unknown focal length and principal point. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR97)*, pages 438–443. IEEE Computer Society Press, 1997.

[Har92] R. I. Hartley. Estimation of relative camera positions for uncalibrated cameras. In *2nd European Conference on Computer Vision (ECCV92)*, pages 579–587, 1992.

[Har93] R Hartley. Cheirality invariants. In *DARPA Image Understanding Workshop*, pages 743–753, 1993.

[Har94a] R. Hartley. Lines and points in three views: a unified approach. In *DARPA Image Understanding Workshop*, pages II:1009–1016, Monterey, 1994.

[Har94b] R. I. Hartley. Euclidean reconstruction from uncalibrated views. In J.L.Mundy, A.Zisserman, and D.Forsyth, editors, *Applications of invariance in computer vision*, pages 237–256. Springer-Verlag, 1994.

[Har94c] R. I. Hartley. Projective reconstruction from line correspondences. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR94)*, pages 903–907, 1994.

[Har95a] R. Hartley. In defence of the 8-point algorithm. In *5th International Conference on Computer Vision (ICCV95)*, pages 1064–1070, Boston, M.A., 1995. IEEE Computer Society Press.

[Har95b] R. I. Hartley. A linear method for reconstruction from points and lines. In *5th International Conference on Computer Vision (ICCV95)*, pages 882–887, 1995.

[Har95c] R. I. Hartley. Theory and practice of projective rectification. Technical Report 2538, INRIA, April 1995.

[Har97] R. I. Hartley. Lines and points in three views and the trifocal tensor. *International Journal of Computer Vision*, 22(2):125–140, March 1997.

[Har98] R. I Hartley. Computation of the quadrifocal tensor. In *5th European Conference on Computer Vision (ECCV98)*, volume 1, pages 20–35. Springer - Verlag, 1998.

[HBS99] A. Heyden, R. Berthilsson, and G. Sparr. An iterative factorization method for projective structure and motion from image sequences. *Image and Vision Computing*, 1713:981–991, 1999.

[Hey95] A. Heyden. Reconstruction from image sequences by means of relative depths. In *5th International Conference on Computer Vision (ICCV95)*, pages 1058–1063. IEEE Computer Society Press, 1995.

[HG93] R. I. Hartley and R. Gupta. Computing matched epipolar projections. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR93)*, pages 549–555, New York, 1993.

[HGC92] R. I. Hartley, R. Gupta, and T. Chang. Stereo from uncalibrated cameras. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR92)*, pages 761–764, 1992.

[Hor90] B. K. P. Horn. Relative orientation. *International Journal of Computer Vision*, 4(1):59–78, January 1990.

[HS88] C. G Harris and M. J. Stephens. A combined corner and edge detector. In *Fourth Alvey Vision Conference*, pages 147–151, 1988.

[HS94] R. I. Hartley and P. Sturm. Triangulation. In *American Image Understanding Workshop*, pages II:957–966, 1994.

[HS97] R. I. Hartley and P. Sturm. Triangulation. *CVIU*, 68(2):146–157, November 1997.

[Hub81] P. J Huber. *Robust Statistics*. John Wiley & Sons, New York, 1981.

[HZ00] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521623049, 2000.

[Jac97] D. W. Jacobs. Linear fitting with missing data: Applications to structure from motion and to characterizing intensity images. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR97)*, pages 206–212. IEEE Computer Society Press, 1997.

[JAP99] T. Jebara, A. Azarbayejani, and A. P. Pentland. 3d structure from 2d motion. *IEEE Signal Processing Magazine - 3D And Stereoscopic Visual Communication*, 16(3), 1999.

[Kah99] F. Kahl. Critical motions and ambiguous euclidean reconstructions in auto-calibration. In *International Conference on Computer Vision (ICCV99)*, pages 469–476, Kerkyra, Greece, 1999.

[Kos93] A. Koschan. Dense stereo correspondence using polychromatic block matching. In D. Chetverikov and W. Kropatsch, editors, *5th International Conference on Computer Analysis of Images and Patterns (CAIP93)*, pages 538–542, Budapest, Hungary, 1993.

[KPG98] R. Koch, M. Pollefeys, and L.V. Gool. Multi viewpoint stereo from uncalibrated video sequences. In *5th European Conference on Computer Vision (ECCV98)*, pages 55–71. Springer-Verlag, 1998.

[KR82] L. Kitchen and A. Rosenfeld. Grey-level corner detection. In *Pattern Recognition Letters*, pages 95–102, 1982.

[KS83a] M. Kenall and A. Stuart. *The Advanced Theory of Statistics.* Charles Griffin and Company, London, 1983.

[KS83b] M. Kendall and A. Stuart. *The Advanced Theory of Statistics.* Charles Griffin and Company, London, 1983.

[KS98] N. K. Kutulakos and S. M. Seitz. A theory of shape by space carving. In *6th International Conference on Computer Vision (ICCV98)*, 1998.

[Lav96] Stéphane Laveau. *Géométrie d'un systèm do N caméras. Théorie, estimation et applications.* PhD thesis, L'école polytechnique, May 1996.

[LDFP93] Q. T. Luong, R. Deriche, O. D Faugeras, and T. Papodopoulo. On determining the fundamental matrix: analysis of different methods and experimental results. Technical Report 1894, INRIA, 1993.

[LF96a] S. Laveau and O. Faugeras. Oriented projective geometry for computer vision. In *4th European Conference on Computer Vision (ECCV96)*, pages 147–156, 1996.

[LF96b] Q. T. Luong and O. D. Faugeras. The fundamental matrix: Theory, algorithms, and stability analysis. In *International Journal of Computer Vision*, volume 17, pages 43–75, January 1996.

[LH81] H. C. Longuet-Higgins. A computer algorithm for reconstructing a scene from two projections. *Nature*, 293:133–135, September 1981.

[LPT00] A. J. Lacey, N. Pinitkarn, and N. A. Thacker. An evaluation of the performance of ransac algorithms for stereo camera calibration. In *12th British Machine Vision Conference (BMVC2001)*, 2000.

[LTCP01] A. J. Lacey, N. A. Thacker, P. Courtney, and S. B. Pollard. Tina 2001: The closed loop 3d model matcher. 2001.

[LTSI96] A. R. Lane, N. A. Thacker, L. Seed, and P. A. Ivey. A generalized computer vision chip. *Real-Time Imaging*, 2:203–213, 1996.

[LV93] Q. T Luong and T Vieville. Motion of points and lines in the uncalibrated case. Technical Report RR-2054, INRIA, 1993.

[LV94] Q. T. Luong and T. Vieville. Canonic representations for the geometries of multiple projective views. In *3rd European Conference on Computer Vision (ECCV94)*, pages A:589–599, 1994.

[MC00a] E. Malis and R. Cipolla. Multi-view constraints between collineations: application to self-calibration from unknown planar structures. In *European Conference on Computer Vision (ECCV2000)*, 2000.

[MC00b] E. Malis and R. Cipolla. Self-calibration of zooming cameras observing an unknown planar structure. In *International Conference on Pattern Recognition (ICPR2000)*, 2000.

[McD81] M. J. McDonnell. Box-filtering techniques. *Computer Graphics and Image Processing*, 17:65–70, 1981.

[MF92] S. Maybank and O. Faugeras. A theory of self-calibration of a moving camera. *International Journal of Computer Vision*, 8:123–151, 1992.

[MGDP94] T. Moons, L.Van Gool, M.Van Diest, and E. Pauwels. Affine reconstruction from perspective image pairs. In J. L. Mundy, A. Zisserman, and D. Forsyth, editors, *Applications of invariance in computer vision*, pages 297–316. Springer-Verlag, 1994.

[MH00] S. Mahamud and M. Hebert. Iterative projective reconstruction from multiple views. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR2000)*, volume 2, pages 430 – 437, June 2000.

[MK00] D. D. Morris and T. Kanade. Image-consistent surface triangulation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR2000)*, volume 1, pages 332–338, 2000.

[MM95] P. F. McLauchlan and D. W. Murray. A unifying framework for structure and motion recovery from image sequences. In *5th International Conference on Computer Vision (ICCV95)*, pages 314–320, 1995.

[MP79] D. Marr and T. Poggio. A computational theory of human stereo vision. In *Royal Society of London*, volume B-204, pages 301–328, 1979.

[MT96] R. Mohr and B. Triggs. Projective geometry for image analysis. can be obtained from the web, September 1996.

[Nis00] D. Nistér. Frame decimation for structure and motion. In *European Conference on Computer Vision (ECCV2000)*, 2000. Volume 2018 of Lecture Notes in Computer Science, p 17.

[NMSO96] Y. Nakamura, T. Matsura, K. Satoh, and Y. Ohta. Occlusion detectable stereo - occlusion patterns in camera matrix. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR96)*, pages 371–378, 1996.

[Nob88] J. A. Noble. Finding corners. *Image and Vision Computing*, 6:121–128, May 1988.

[OK85] Y. Ohta and T. Kanade. Stereo by intra- and inter-scanline search using dynamic programming. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 7, pages 139–154, March 1985.

[pbESI] produced by Eos Systems Inc. Photomodeller. On Web. http://www.photomodeler.com/.

[pbR] produced by RealViz. Image processing factory. On Web. http://www2.realviz.com/.

[Pea01] K. Pearson. On lines and planes of closest fit to systems of points in space. *Philosophical Magazine Series 6*, 2:559, 1901.

[PF98] T. Papadopoulo and O. D. Faugeras. A new characterization of the trifocal tensor. In *5th European Conference on Computer Vision (ECCV98)*, 1998.

[PKG97] M. Pollefeys, R. Koch, and L. Gool. Self calibration and metric reconstruction in spite of varying and unknown internal camera parameters. Technical Report KUL/ESAT/MI2/9707, Katholieke Universiteit Leuven, 1997.

[PKG99] M. Pollefeys, R. Koch, and L. Van. Gool. A simple and efficient rectification method for general motion. In *Proc. International Conference on Computer Vision*, pages 496–501, Corfu (Greece), 1999.

[PMF85] S. B. Pollard, J. E. W. Mayhew, and J. P Frisby. PMF: A stereo correspondence algorithm using a disparity gradient limit. *Perception*, 14:449–470, 1985.

[Pol99] M. Pollefeys. *Self-calibration and metric 3D reconstruction from uncalibrated image sequences.* PhD thesis, K.U.Leuven, 1999.

[PZ98] P. Pritchett and A. Zisserman. Wide baseline stereo matching. In *6th International Conference on Computer Vision (ICCV98)*, pages 754–760, January 1998.

[Qua94] L. Quan. Invariants of 6 points in 3 uncalibrated images. In J.O.Eckland, editor, *3rd European Conference on Computer Vision (ECCV94)*, pages 459–469. Springer-Verlag, 1994.

[Qua95] L. Quan. Invariants of 6 points and projective reconstruction from 3 uncalibrated images. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 17, pages 34–46, January 1995.

[RC98] S. Roy and I. J. Cox. A maximum-flow formulation of the n-camera stereo correspondence problem. In *6th International Conference on Computer Vision (ICCV98)*, pages 492–499, Bombay, India, January 1998.

[RGH80] T. W Ryan, R. T. Gray, and B. R. Hunt. Prediction of correlation errors in stereo-pair images. *Optical Engineering*, 19(3):312–322, 1980.

[RL87] P. J Rousseeuw and A. M Leory. *Robust Regression and Outlier Detection.* Jogn Wiley & Sons, New York, 1987.

[SA90] M. E. Spetsakis and Y. Aloimonos. Structure from motion using line correspondences. *International Journal of Computer Vision*, 4(3):171–183, 1990.

[Sam82] P. D Sampson. Fitting conic sections to 'very scattered' data: An iterative refinement of the bookstein algorithm. *Computer Graphics and Image Processing*, 18:97–108, 1982.

[Sam88] P. Samuel. *Projective Geometry.* Springer-Verlag, 2nd edition, 1988.

[Saw98] Robust video mosaicing through topology inference and local to global alignment. In H. S. Sawhney, S. Hsu, and R. Kumar, editors, *5th European Conference on Computer Vision (ECCV98)*, volume 1407 of *Lecture Notes in Computer Science.* Springer, 1998.

[Sha95] A. Shashua. Algebraic functions for recognition. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 17, pages 779–789, August 1995.

[SK51] J. G. Semple and G. T. Kneebone. *Algebraic Projective Geometry.* Oxford; Clarendon 1951 (Oxford science publications), 1951.

[SKZ99] Heung-Yeung Shum, Qifa Ke, and Z. Zhang. Efficient bundle adjustment with virtual key frames: A hierarchical approach to multi-frame structure from motion. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR99)*, pages 538–543, June 1999.

[Sla80] C. C Slama, editor. *Manual of Photogrammetry.* American Society of Photogrammetry, Falls Church, Va, fourth edition, 1980.

[SMI97] R. Sébastien, J. Meunier, and J. C. Ingemar. Cylindrical rectification to minimize epipolar distortion. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR97)*, pages 393–399, 1997.

[Spa94] G. Sparr. A common framework for kinetic depth, reconstruction and motion for deformable objects. In *3rd European Conference on Computer Vision (ECCV94)*, 1994.

[Spa96] G. Sparr. Simultaneous reconstruction of scene structure and camera locations from uncalibrated image sequences. In *13th International Conference on Pattern Recognition (ICPR96)*, pages 328–333, 1996.

[ST96] P. Sturm and B. Triggs. A factorization based algorithm for multi-image projective structure and motion. In *4th European Conference on Computer Vision (ECCV96)*, pages II:709–720, 1996.

[Sto91] J. Stolfi. *Oriented Projective Geometry, A Framework for Geometric Computations.* Academic Press, Inc., 1250 Sixth Avenue, San Diego, CA., 1991.

[Str94] Streilein94. Towards automation in architectural photogrammetry: Cad-based 3d-feature extraction. *ISPRS Journal of Photogrammetry & Remote Sensing*, 49(5):4–15, October 1994.

[Stu97] P. Sturm. Critical motion sequences for monocular self-calibration and uncalibrated euclidean reconstruction. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR97)*, pages 1100–1105. IEEE Computer Society Press, 1997.

[Stu99] P. Sturm. Critical motion sequences for self-calibration of cameras and stereo systems with variable focal length. In *10th British Machine Vision Conference (BMVC99)*, pages 63–72, Nottingham, 1999.

[Sun97] C. Sun. A fast stereo matching method. In *Digital Image Computing: Techniques and Applications*, pages 95–100, 1997.

[SW00] A. Shashua and L. B. Wolf. On the structure and properties of the quadrifocal tensor. In *European Conference on Computer Vision (ECCV2000)*, pages 710–724, 2000.

[SZ98] C. Schmid and A. Zisserman. The geometry and matching of curves in multiple views. In *5th European Conference on Computer Vision (ECCV98)*, pages 394–409. Springer-Verlag, June 1998.

[SZ00] C. Schmid and A. Zisserman. The geometry and matching of lines and curves over multiple views. *International Journal of Computer Vision*, 40(3):199–233, 2000.

[SZH00] F. Schaffalitzky, A. Zisserman, and P. H. S. Hartley, R. I.and Torr. A six point solution for structure and motion. In *European Conference on Computer Vision (ECCV2000)*. Springer-Verlag, June 2000.

[TFTR98] T. Tommasini, A. Fusiello, E. Trucco, and V. Roberto. Making good features track better. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR98)*, pages 178–183, Santa Barbara, CA, June 1998. IEEE Computer Society Press.

[TFZ98] P. H. S. Torr, A. W. Fitzgibbon, and A. Zisserman. Maintaining multiple motion model hypotheses over many views to recover matching and structure. In *6th International Conference on Computer Vision (ICCV98)*, pages 485–491, January 1998.

[TFZ99] P. H. S. Torr, A. W. Fitzgibbon, and A. Zisserman. The problem of degeneracy in structure and motion recovery from uncalibrated image sequences. *International Journal of Computer Vision*, 32(1):27–44, August 1999.

[TK92] C. Tomasi and T. Kanade. Shape and motion from image streams under orthography: A factorization approach. *International Journal of Computer Vision*, 9(2):137–154, November 1992.

[TM91] N. Thacker and J. E. W. Mayhew. Optimal combination of stereo camera calibration from arbitrary stereo images. *Image and Vision Computing*, 9:27–32, 1991.

[TMHF00] Bill Triggs, Philip McLauchlan, Richard Hartley, and Andrew Fitzgibbon. Bundle adjustment – A modern synthesis. In B. Triggs, A. Zisserman, and R. Szeliski, editors, *Vision Algorithms: Theory and Practice*, LNCS, pages 298–375. Springer Verlag, 2000.

[Tor95] P. Torr. *Motion Segmentation and Outlier Detection.* PhD thesis, Department of Engineering Science, University of Oxford, 1995.

[Tor97] P. H. S. Torr. An assessment of information criteria for motion model selection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR97)*, pages 47–52, 1997.

[Tor99] P. H. S. Torr. Model selection for structure and motion recovery from multiple images. Technical Report MSR-TR-99-16, Microsoft Research, One Microsoft Way, Redmond, WA 98052, USA, March 1999.

[Tri87] H. P. Trivedi. Estimation of stereo and motion parameter using a variational principle. *Image and Vision Computing*, 5(2):181–183, May 1987.

[Tri95] B. Triggs. The geometry of projective reconstruction i: Matching constraints and the joint image. *5th International Conference on Computer Vision (ICCV95)*, pages 338–343, 1995.

[Tri97] B. Triggs. Auto-calibration and the absolute quadric. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR97)*, pages 609–614. IEEE Computer Society Press, 1997.

[Tri98] B. Triggs. Autocalibration from planar scenes. In *5th European Conference on Computer Vision (ECCV98)*, 1998.

[Tri00] B. Triggs. Plane + parallax, tensors and factorization. In *European Conference on Computer Vision (ECCV2000)*, 2000.

[TS94] C. Tomasi and J. Shi. Good features to track. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR94)*, pages 593–600, June 1994.

[TV96] J. Tarel and J. Vezien. Camcal v1.0 manual a complete software solution for camera calibration. Technical Report 196, INRIA, September 1996.

[TVPG99] T. Tuytelaars, M. Vergauwen, M. Pollefeys, and L.V. Gool. Image matching for wide baseline stereo. In *International Conference on Forensic Human Identification*, 1999.

[TZ97] P. H. S. Torr and A Zisserman. Robust parameterization and computation of the trifocal tensor. *Image and Vision Computing*, 15:591–607, 1997.

[TZ98] P. H. S. Torr and A. Zisserman. Robust computation and parameterization of multiple view relations. In *Proc. 6th International Conference on Computer Vision, Bombay, India*, pages 727–732, January 1998.

[TZ00] P. H. S. Torr and A. Zisserman. MLESAC: A new robust estimator with application to estimating image geometry. *Computer Vision and Image Understanding*, 78:138–156, 2000.

[TZM98] P. H. S. Torr, A. Zisserman, and S. Maybank. Robust detection of degenerate configurations for the fundamental matrix. *Computer Vision and Image Understanding*, 71(3):312–333, September 1998.

[WA92] T. S. Weng, J. Huang and N. Ahuja. Motion and structure from line correspondences: Closed-form solution, uniqueness and optimization. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 14, pages 318–336, March 1992.

[Wei99] M. A. Weiss. *Data Structures and Algorithm Analysis in C++*. Florida International University, 2nd edition, 1999.

[ZBR95] A. Zisserman, P. Beardsley, and I. Reid. Metric calibration of a stereo rig. In *IEEE Workshop on Representation of Visual Scenes, Boston*, pages 93–100, 1995.

[ZDFL94] Z. Zhang, R. Deriche, O. Faugeras, and Q. T. Luong. A robust technique for matching two uncalibrated images through the recovery of the unknown epipolar geometry. Technical Report 2273, INRIA, May 1994.

[ZDFL95] Z. Zhang, R. Deriche, O. Faugeras, and Q. Luoung. A robust technique for matching two uncalibrated images through the recovery of the unknown epipolar geometry. *Artificial Intelligence*, 78:87–119, 1995.

[ZF92] Z. Zhang and O. Faugeras. *3D Dynamic Scene Analysis.* Springer - Verlag, 1992.

[Zha97] Z Zhang. Parameter estimation techniques: A tutorial with application to conic fitting. *Image and Vision Computing*, 15(1):59–76, January 1997. Also INRIA research report No.2676.

[Zha98] Z. Zhang. Determining the epipolar geometry and its uncertainty: A review. *International Journal of Computer Vision*, 27(2):161–195, March 1998. Also available as INRIA research report No.2560.

[ZX97] Z. Zhang and G. Xu. A general expression of the fundamental matrix for both projective and affine cameras. In *Fifteenth International Joint Conference on Artificial Intelligence (IJCAI97)*, pages 1502–1507, 1997.

Appendix A

Nonlinear refinement

For many of the methods in this thesis, nonlinear least-squares refinement is absolutely essential. For many problems (particularly those with smooth functions), the Levenberg-Marquardt (abbreviated to LM) iteration algorithm is widely accepted as being the most successful algorithm, and this appendix will aim to provide a very brief outline of the method. The outline will only be sufficient to develop an understanding of the algorithm's practical form, and for a more complete description the reader is referred to more concise books on the subject, such as [DS83].

A.1 Newton Iteration

Given a function $\mathbf{y} = f(\mathbf{x})$, a measured value $\hat{\mathbf{y}}$ for $\mathbf{y}$ and an initial estimated value $\mathbf{x}_0$ for $\mathbf{x}$, Newton iteration attempts to find the vector $\hat{\mathbf{x}}$ that most nearly satisfies this functional relation $\mathbf{y} = f(\mathbf{x})$. It does this by continually refining the estimate under the assumption that the function f is locally linear. More precisely, there will be an error ϵ_0 associated with the initial estimate such that:

$$\hat{\mathbf{y}} = f(\mathbf{x}_0) + \epsilon_0 \tag{A.1}$$

Assuming that f is locally linear it can be approximated at $\mathbf{x}_0$ by:

$$f(\mathbf{x}_0 + \boldsymbol{\Delta}) = f(\mathbf{x}_0) + J\boldsymbol{\Delta} \tag{A.2}$$

where $J = \frac{d\mathbf{y}}{d\mathbf{x}}$ is the mapping represented by the Jacobian matrix. Setting $\mathbf{x}_1 = \mathbf{x}_0 + \boldsymbol{\Delta}$ and substituting into equation A.2 leads to $\hat{\mathbf{y}} - f(\mathbf{x}_1) = \hat{\mathbf{y}} - f(\mathbf{x}_0) - J\boldsymbol{\Delta}$. Identifying with equation A.1 gives $\hat{\mathbf{y}} - f(\mathbf{x}_1) = \epsilon_0 - J\boldsymbol{\Delta}$. As such, minimising the error $\|\hat{\mathbf{y}} - f(\mathbf{x}_1)\|$ is

equivalent to minimising:

$$\|\epsilon_0 - J\Delta\| \tag{A.3}$$

Solving for the unknown Δ in equation A.3 is a linear minimisation that can be solved by the method of normal equations. The minimum occurs when $J\Delta - \epsilon_0$ is perpendicular to the row space of J, leading to the so called *normal equations* $J^T(J\Delta - \epsilon_0) = 0$ which simplify to $J^T J\Delta = J^T\epsilon_0$. Finally, the normal equations can be solved by any appropriate means, such as Gaussian elimination and the recovered Δ used to update the estimated parameter vector $\mathbf{x}_0$.

To sum up, the solution is found by stepping in the direction of the functions gradient taking successive approximations according to the formulae (under the assumption the function is locally linear):

$$\mathbf{x}_{i+1} = \mathbf{x}_i + \Delta_i$$

where Δ_i is the solution to the normal equations:

$$J^T J\Delta_i = J^T\epsilon_i$$

Unfortunately, this iteration procedure can often fail to converge to the required least-squares solution $\hat{\mathbf{x}}$, or can get stuck on a local minimum value. In some situations it may not even converge at all. The behaviour and success of the algorithm depends very heavily on the initial estimate $\mathbf{x}_0$.

A.2 Levenberg-Marquardt Iteration

The Levenberg-Marquardt iteration method (often abbreviated to LM) is a slight variation on the Newton iteration method. The normal equations $N\Delta = J^T J\Delta = J^T\epsilon$ are replaced by the *augmented normal equations* $N'\Delta = J^T\epsilon$, where $N'_{ii} = (1+\lambda)N_{ii}$ and $N'_{ij} = N_{ij}$ for $i \neq j$. The value λ is given an initial value, typically 10^{-5}, and the augmented normal equations solved as for Newton iteration. If the value of Δ leads to a reduction in the error, then the increment is accepted and λ is divided by 10 before the next iteration. On the other hand, if Δ leads to an increase in error, then λ is multiplied by 10 and the augmented normal equations are solved again. This process repeats until a value of Δ is found which gives a decreased error. This repeated solving, using different values for λ, constitutes one iteration of the LM algorithm.

Appendix B

Bundle Adjustment

The bundle adjustment [Bro58] is a well known and very well established method for providing a nonlinear refinement of all structure and all cameras in a scene (see [Sla80, TMHF00, Har92, SKZ99]). The basis of bundle adjustment is to find the least-squares solution that minimises the re-projection error:

$$\sum_{ij} d_E^2 \left(P_i H \mathbf{X}_j, \mathbf{x}_j^i \right) \tag{B.1}$$

for all cameras P_i in image i, 3D structure $\mathbf{X}_j$ and associated 2D image features $\mathbf{x}_j^i$. This equation is nonlinear, involving unknowns for both structure and cameras as well as an unknown scale factor that has to be eliminated by dividing through. For projective cameras, in general the best that can be done to minimise the exact error measure in equation B.1 is to refine a supplied initial solution using a gradient descent technique such as Levenberg-Marquardt (abbreviated to LM) or Newton iteration (see appendix A for a detailed description).

Whilst it would be quite straightforward simply to use the error measure in equation B.1 in a conventional LM implementation, it would unfortunately not be practical because of the size of the problem involved. For example, consider a normal scene involving 40 images with 2000 points. This leads to $40 * 11 + 2000 * 3 = 6440$ unknowns. As would be expected, it is not tractable to solve for that many unknowns using normal methods. Fortunately a solution is still feasible, because the Jacobian matrix for the problem has a special sparse block structure. This leads to a similar sparse block structure for the normal equations used in LM or Newton iteration (see appendix A for a description of LM). If the sparsity is properly exploited, it is possible to obtain an enormous simplification in the solution of the normal equations.

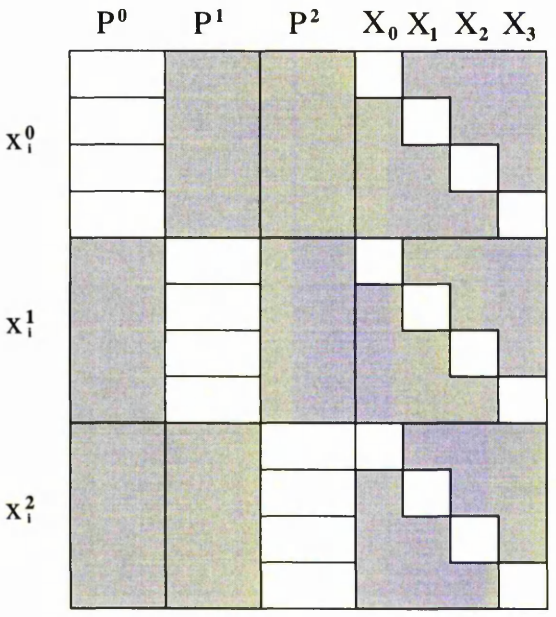

Figure B.1: Graphical illustration of the sparse Jacobian matrix for bundle adjustment

To illustrate the sparsity, consider the two different types of independent parameters, cameras P_i and structure $\mathbf{X}_j$. Altering a camera P_i will alter the re-projection error for all points in the same image i as the camera, and altering 3D structure $\mathbf{X}_j$ will lead to a change of re-projection error in all projections of that point $\mathbf{x}_i^j$. This means that the matrix of partial derivatives of the dependent parameters with respect to the independent parameters has a particular sparse structure as shown in figure B.1. In the figure, the grey regions indicate areas that are invariably filled with zeros and white areas indicate areas with varying value.

The case illustrated is three cameras and four points visible in all images. If a point were not visible in certain images then the relevant rows would be missing, and if a camera were fixed then the corresponding columns would be missing (for example the first camera P_0 might be fixed to $(I|0)$). Given that the Jacobian J has a special sparse structure, so do the normal equations $J^T J\Delta = J^T\epsilon$ as illustrated in figure B.2

If the form of the normal equations is examined a little more closely, it can be seen that it is possible to give individual formulae for each of the blocks in the normal equations. Given N images and M points then $\frac{d\mathbf{x}_j^i}{dP_i}$ for $j \in \{1,\ldots,11\}$ can be defined as the $M \times 11$ matrix of partial derivatives of the image points $\mathbf{x}_j^i$ with respect to the matrix of camera parameters P_i. Also, $\frac{d\mathbf{x}_j^i}{d\mathbf{x}_j}$ for $i \in \{1,\ldots,4\}$ can be defined as the $N \times 4$ matrix of partial derivatives of the projected image points $\mathbf{x}_j^i$ with respect to 3D structure $\mathbf{x}_j$. Finally, given $\epsilon\left(\mathbf{x}_j^i\right)$ as the re-projection error from equation B.1, we may write:

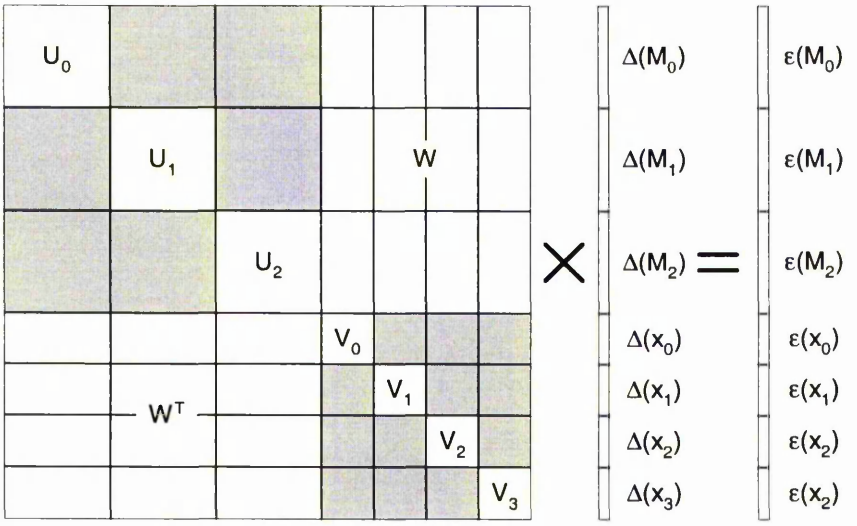

Figure B.2: Graphical illustration of the sparse normal equations for bundle adjustment

$$
\begin{aligned}
U_i &= \textstyle\sum_j \frac{dx_j^i}{dP_i}^T \frac{dx_j^i}{dP_i} \\
V_j &= \textstyle\sum_i \frac{dx_j^i}{dx_j}^T \frac{dx_j^i}{dx_j} \\
W_{ij} &= \frac{dx_j^i}{dP_i}^T \frac{dx_j^i}{dx_j} \\
\epsilon\left(P_i\right) &= \textstyle\sum_j \frac{dx_j^i}{dP_i}^T \epsilon\left(u_j^i\right) \\
\epsilon\left(x_j\right) &= \textstyle\sum_j \frac{dx_j^i}{dx_j}^T \epsilon\left(u_j^i\right)
\end{aligned}
\tag{B.2}
$$

And so, the normal equations $J^T J\Delta = J^T\epsilon$ may be written in the form:

$$
\begin{pmatrix} U & W \\ W^T & V \end{pmatrix}
\begin{pmatrix} \Delta\left(P\right) \\ \Delta\left(X\right) \end{pmatrix}
=
\begin{pmatrix} \epsilon\left(P\right) \\ \epsilon\left(X\right) \end{pmatrix}
$$

where matrix U, V and vectors $\epsilon\left(P\right)$, $\epsilon\left(X\right)$ are made up of sub blocks as given in equation B.2, and $\Delta\left(P\right)$, $\Delta\left(X\right)$ also naturally decompose into sub blocks.

If it is assumed that V is invertible (this is reasonable when using Levenberg-Marquardt - see below) and multiply each side of the normal equations on the left by:

$$
\begin{pmatrix} I & -WV^{-1} \\ 0 & I \end{pmatrix}
$$

and work through, the following equation is obtained:

$$
\begin{pmatrix} U - WV^{-1}W^T & 0 \\ W^T & V \end{pmatrix}
\begin{pmatrix} \Delta\left(P\right) \\ \Delta\left(X\right) \end{pmatrix}
=
\begin{pmatrix} \epsilon(P) - WV^{-1}\epsilon(X) \\ \epsilon(X) \end{pmatrix}
\tag{B.3}
$$

Equation B.3 can then be split into two parts to be solved separately. The top half gives:

$$U - WV^{-1}W^T\Delta(P) = \epsilon(P) - WV^{-1}\epsilon(X) \tag{B.4}$$

which presents a set of $N*11$ equations in $N*11$ unknowns and can be solved to give $\Delta(P)$. The resulting solution can then be substituted into the bottom half of equation B.3 which after rearranging gives:

$$\Delta(X) = V^{-1}\left(\epsilon(X) - W^T\Delta(P)\right) \tag{B.5}$$

allowing a simple solution for $\Delta(X)$. Because of the block-diagonal form of V, the equations B.4 may be computed efficiently using the quantities computed in B.2. In particular, the matrix $A = U - WV^{-1}W^T$ divides naturally into sub-blocks, where the (i,j) th sub-block is the matrix:

$$A_{ij} = \delta_{ij}U_i - \sum_k W_{ik}V_k^{-1}W_{jk}^T \tag{B.6}$$

Similarly, the vector $\mathbf{b} = \epsilon(P) - WV^{-1}\epsilon(X)$ also divides into blocks of the form:

$$\mathbf{b}_i = \epsilon(P_i) - \sum_j W_{ij}V_j^{-1}\epsilon(x_j) \tag{B.7}$$

It is worth noting that the matrix A and the vector $\mathbf{b}$ can both be computed directly without needing to compute and store the matrix J or the normal equations. The amount of computation required is linear in the number of points $\mathbf{x}_j$ involved, and also linear in the total number of observed points $\mathbf{x}_j^i$.

Also, the back substitution given in equation B.5 can be done block by block as follows:

$$\Delta(\mathbf{x}_j) = V_j^{-1}\left(\epsilon(\mathbf{x}_j) - \sum_i W_{ij}^T\Delta(P_i)\right) \tag{B.8}$$

This back substitution also requires computation time linear in the number of points involved. So far, the normal equations being solved are those from Newton Iteration. It is easy to extend this to Levenberg-Marquardt by augmenting the matrix J^TJ with the LM parameter λ. This is equivalent to augmenting the matrices U_i and V_j, a process that will help to ensure that the matrices V_j will be invertible even in degenerate cases when V_j is singular. This effect means it is not essential to avoid over parameterisation of the minimisation problem.

B.1 Implementation Details

The implementation can follow the above description in quite a straightforward manner. For a decent computational efficiency, it is first necessary to precompute all the sub-blocks

in equation B.2. This can be done quite easily without needing to store the matrices of partial derivatives. Everything else can then be achieved without making further demands on memory usage.

After this, all the V_j matrices can be inverted. Equations B.7 and B.6 can then be used to find $\mathbf{b}_i$ and A_{ij} before solving for $\Delta(X)$ using equation B.5. Finally this can be substituted into B.8 to give the solution for $\Delta(\mathbf{x}_j)$.

At this point it is worth mentioning a further and mainly implementation improvement to bundle adjustment, one not made explicit in any descriptions found by the author. If the bundle adjustment is to be used for 'natural' problems then there will usually be a large number of images and many points will only be tracked for a few images. Because of this, it is absolutely essential to ensure appropriate rows and columns are missing from the Jacobian. Also, it is absolutely essential to ensure that the appropriate W_{ij} matrices are not calculated or stored. If this is not done, then memory requirements quickly become out of control. For example, given 40 images with 1000 points tracked on average 6 images, then the memory required to store all 11×3 W_{ij} matrices would be $O(40 * 1000 * 11 * 3) = O(1320000)$, whereas if appropriate matrices are omitted it would be $O(6 * 1000 * 11 * 3) = O(198000)$, a difference of approximately $6\frac{1}{2}$ times the memory usage.

A further benefit of simply considering certain W_{ij} to be zero is that calculations involving multiplication with a zeroed W_{ij} can simply be ignored and set to 0. This occurs when solving using equation B.5, when calculating A_{ij} using equation B.6 or $\mathbf{b}_i$ using equation B.7, and finally when solving for $\Delta(\mathbf{x}_j)$ using equation B.8. Again, given a fairly 'natural' problem, this results in very significant performance increases.

Given an efficient implementation, it is quite feasible in terms of memory and time usage to solve systems involving hundreds of images with thousands of points. Without using the technique, such systems would be impossible to handle, for example 100 images with 2000 points would result in normal equations with dimension 7100×7100, which would clearly be impractical to solve using normal methods (such as Gaussian elimination).

B.2 Euclidean Bundle Adjustment

In this work, use is made of bundle adjustment, not only for projective reconstruction, but also for purposes of refining Euclidean reconstructions. The bundle adjustment algorithm remains largely unaltered, but it does become necessary to parameterise the camera projection matrices differently. To enforce a Euclidean frame, each camera matrix must be decomposed

in terms of a calibration matrix, a rotation matrix and a translation matrix.

Factorisation can be used to perform this decomposition. Considering the camera matrix as $P = [A|\mathbf{a}]$, A can be decomposed using RQ decomposition into an upper triangular matrix and an orthogonal matrix. Relating this to the Euclidean camera matrix $P \simeq [KR|| - KR\mathbf{t}]$, these two matrices represent the calibration and rotation. The translation matrix can then be obtained as $-A^{-1}\mathbf{a}$ since relating this to the Euclidean form gives $A = KR$ and $\mathbf{a} = -KR\mathbf{t}$, so this multiplication removes KR. Note that the supplied form of K used in this work assumes a right handed coordinate system, and so, if a left handed form is being used, all y coordinates must have their sign changed to prevent orientation problems with the cameras.

However, this still leaves the problem of how to parameterise the rotation matrix in a minimal manner. In [Hor90], it has been proposed to parameterise rotations using quaternions, but this has the disadvantage that a rotation is parameterised by 4 parameters instead of the minimal 3. Instead, experience has shown that Euler angles perform much better, provided care is taken to avoid problems with singularities.

In order to avoid singularities in the Euler representation, it was proposed in [Har94b] to represent rotations R_i as incremental with respect to a base rotation X_i as $R_i = X_i\Delta(\theta_i, \phi_i, \kappa_i)$. The base rotation X_i is taken as the initial guess, and $\Delta(\theta_i, \phi_i, \kappa_i)$ is the rotation represented by Euler angles. Initially, the rotation parameters θ_i, ϕ_i and κ_i are all set to 0 and subsequently Δ is the identity mapping. At the end of each LM iteration, the base rotation is reset to the new rotation as $X_i\Delta(\theta_i, \phi_i, \kappa_i)$ and θ_i, ϕ_i, κ_i are reset to 0.

Appendix C

Random Sampling for Robust Model Fitting

Random sampling algorithms are an approach to parameter estimation for data that may contain outliers. That is to say, the set of data from which parameters are to be estimated may contain items which do not fit the distribution of the selected error model at all. The basis of random sampling is to take minimal samples of the data and estimate the model using these. It is hoped that, one or more of the minimal samples will not contain any outliers and so will produce a valid solution. The model that produces the best result according to some criteria involving all the data is then kept as the best result.

For example, standard least-squares methods attempt to minimise $\sum_i r_i^2$, where the residual r_i can be defined as the difference between the ith observation and the fitted value. If errors in observed data conform to a Gaussian distribution then the global minimum of this function represents the maximum likelihood estimate. However, it is not uncommon for real data to be contaminated by outliers with large residuals that would be considered highly improbable in a Gaussian distribution. Because the function is squared, any of these large residuals will have a dominating effect on any estimated parameters if a least-squares method is used.

To deal with the problem of outliers, it is possible to minimise something more robust than the sum of the residuals squared, for example the median residual or some robust function such as a Huber function (this will be described in detail later in this appendix). Random sampling algorithms provide one means of performing this minimisation.

There have been a number of random sampling algorithms proposed in the literature. Primarily, these consist of least median of squares (LMedS) ([RL87]) which minimises the

median of the residuals, random sampling consensus (RANSAC) [FB81] which minimises the number of outliers, and maximum likelihood sample consensus (MLESAC) [TZ00] which minimises a Huber function (to be detailed later). To give a feel, both LMedS and MLESAC will now be described in detail. Note the versions here assume Gaussian distributed errors.

C.1 Least Median of Squares (LMedS)

As the name suggests, Least Median of Squares attempts to minimise the median residual. The median can easily be seen to be more robust than the mean by a simple example. Consider a set of numbers - 1,100,101,102,103 where 1 is considered to be an outlier. The mean of this set of numbers is 81.4 whereas the median is 101. This means the central value of the correct data has been estimated much more accurately by the median. So in LMedS, parameters are estimated by solving the following minimisation problem for a particular error measure function - r_i:

$$\min \text{median}_i r_i^2$$

Since this is a nonlinear minimisation, there is no straightforward formula that can be used to minimise it. Instead, it can be minimised by searching in the space of all possible parameters estimated from the data. Since this space is going to be far too large for an exhaustive search, only a randomly chosen subset can be analysed. The algorithm here is based on the algorithm in chapter 5 of [RL87]:

1. Initially m random sub-samples of p data items are drawn. p should be the minimum numbers of observations needed to calculate the given parameters.

2. For each sub-sample, indexed by j, the parameters $\mathbf{a}_j$ are determined.

3. For every $\mathbf{a}_j$ the median of the square residuals M_j is calculated with respect to the whole set of points, i.e.:

$$M_j = \text{median}_{i=1,\ldots,n} r_i^2$$

4. Finally, the estimate $\mathbf{a}_j$ for which M_j is minimal is retained and this forms the solution

This algorithm is now complete, except that we need to determine some sensible way of finding m – the number of sub-samples used. The idea is that we want to pick enough sub-samples to be certain that at least one of those sub-samples contains no outliers. So, if we

assume that we have a fraction ϵ of outliers and a sub-sample size of p, then the probability that at least one of the sub-samples is good is given by:

$$P = 1 - [1 - (1 - \epsilon)^p]^m \tag{C.1}$$

We can then rearrange this to determine m in terms of ϵ, p and P to get:

$$m = \frac{\log(1 - P)}{\log[1 - (1 - \epsilon)^p]}$$

It is then possible to determine a value for P by requiring it to be near 1 and setting a value of ϵ. For example, $P = 0.999$ and ϵ is set to 40 percent initially.

The LMedS technique works much better than least-squares when there are outliers, but it is very inefficient in the presence of Gaussian noise. To remedy this, after robust estimation outliers are normally removed and least-squares techniques used to produce a final estimate.

C.2 Maximum Likelihood Sample Consensus (MLESAC)

Unlike LMedS, which minimises the median, MLESAC attempts to minimise a robust Huber function of the residuals r_i:

$$\gamma\left(\frac{r_i}{\sigma}\right) = \begin{cases} r_i^2 & r_i < \rho \\ \rho^2 & r_i \geq \rho \end{cases} \tag{C.2}$$

where ρ is some threshold usually based on a confidence limit and σ is the standard deviation of the residuals. In this case, σ must be approximated with the robust standard deviation:

$$\sigma = 1.4826 \left[1 + \frac{5}{n - p}\right] \underset{i}{\text{median}} |r_i|$$

for n observations and a parameter space of dimension p (see [RL87] for full details). This function assigns outliers a fixed cost to reflect the notion that they probably arise from a diffuse or uniform distribution, the likelihood of which is constant. Inliers on the other hand conform to a Gaussian distribution, so are assigned the familiar cost used to find a maximum likelihood estimate.

Minimisation occurs using random sampling to select minimal sets of samples from which to produce an estimate for the model parameters. Each of these is evaluated against the whole data set using the sum of the Huber function for all items of data $\sum_i \gamma_i$. The one

which produces the minimal sum value is then accepted as the best solution. The number of samples to take can be arrived at using the same criteria as for LMedS, but on the whole MLESAC is a minimisation function and so it is best to set the number of samples to some suitably larger number than the minimal number of samples to ensure a sample of inliers. For fundamental matrix estimation this was set to 300.

Because a Huber function is being minimised, it is very reasonable after obtaining an estimate using random sampling to attempt an iterative nonlinear minimisation of the Huber function using a method such as those presented in appendix A. As with LMedS, it can also be a good idea after this stage to remove outliers and produce a solution using only the inlying data and a normal least-squares method.

Appendix D

Determining Triplet Geometry using only Six Points

When using robust random sampling algorithms (see appendix C), it is essential to develop a method for calculating the trifocal tensor that uses minimal data. Since the trifocal tensor depends on 18 parameters, it follows that at least 18 constraints will be necessary for the tensor to be determined. This can be provided by the projection of 6 world points into the three images, giving $3 * 6 = 18$ independent constraints.

The method given here for calculation of the trifocal tensor is based on the method given in [Qua94] for computing the structure of 6 points in 3 views. The basis of the technique is to simplify the problem by first aligning the points with the standard basis, and then solving for the remaining unknowns.

To align points with the basis, the six points are assigned projective world coordinates $(1,0,0,0)^T$, $(0,1,0,0)^T$, $(0,0,1,0)^T$, $(0,0,0,1)^T$,$(1,1,1,1)^T$ and $(X,Y,Z,W)^T$ where X,Y,Z,W are unknown. Similarly, the corresponding image points are assigned to the projective basis of each image, i.e. $(1,0,0)^T$, $(0,1,0)^T$, $(0,0,1)^T$, $(1,1,1)^T$, $(x_5,y_5,w_5)^T$ and $(x_6,y_6,w_6)^T$.

The transformations B^i for each image i to take points x_1, x_2, x_3, x_4 into this canonical frame can be efficiently calculated as:

$$B^i = [\lambda_1 x_1, \lambda_2 x_2, \lambda_3 x_3]$$

where:

$$\begin{pmatrix} \lambda_1 & \lambda_2 & \lambda_3 \end{pmatrix} = \begin{bmatrix} x_1 & x_2 & x_3 \end{bmatrix}^{-1} x_4$$

This leads to a simple set of equations for the projection of all six points in each image i:

$$\begin{bmatrix} 1 & 0 & 0 & 1 & x_5^i & x_6^i \\ 0 & 1 & 0 & 1 & y_5^i & y_6^i \\ 0 & 0 & 1 & 1 & w_5^i & w_6^i \end{bmatrix} \simeq \begin{bmatrix} \alpha^i & 0 & 0 & \delta^i \\ 0 & \beta^i & 0 & \delta^i \\ 0 & 0 & \gamma^i & \delta^i \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & X \\ 0 & 1 & 0 & 0 & 1 & Y \\ 0 & 0 & 1 & 0 & 1 & Z \\ 0 & 0 & 0 & 1 & 1 & W \end{bmatrix} \tag{D.1}$$

The problem is now to recover X, Y, Z, W and $\alpha^i, \beta^i, \gamma^i, \delta^i$ for each camera. From the projection of the last two points (last two columns) in equation D.1, it is possible to determine the values of the sixth space point and camera parameters in terms of the fifth and sixth image coordinates. Writing this as a linear system in terms of the unknown cameras gives:

$$\begin{bmatrix} w_5^i & 0 & -x_5^i & w_5^i - x_5^i \\ 0 & w_5^i & -y_5^i & w_5^i - y_5^i \\ w_6^i X & 0 & -x_6^i Z & w_6^i W - x_6^i W \\ 0 & w_6^i Y & -y_6^i Z & w_6^i W - y_6^i W \end{bmatrix} \begin{bmatrix} \alpha^i \\ \beta^i \\ \gamma^i \\ \delta^i \end{bmatrix} = 0$$

The 4×4 matrix on the left is rank 3 and so must have a determinant of 0:

$$\begin{aligned} \left(-x_5^i y_6^i + x_5^i y_6^i\right)(WX - YZ) + \left(x_6^i y_5^i - y_5^i w_6^i\right)(WY - YZ) + \\ \left(-x_6^i w_5^i + y_6^i w_5^i\right)(WZ - YZ) + \left(-x_5^i w_6^i + y_5^i w_6^i\right)(XY - YZ) + \\ \left(x_5^i y_6^i - y_6^i w_5^i\right)(XZ - YZ) = 0 \end{aligned} \tag{D.2}$$

This is true for all three images giving three linear constraints on the five unknowns: $(WX - YZ)$, $(WY - YZ)$, $(WZ - YZ)$, $(XY - YZ)$ and $(XZ - YZ)$. If the constraints from equation D.2 are stacked into a 3×5 matrix, this matrix will be of rank 3 provided the 3D points are in general position. The two dimensional null-space of this matrix may be recovered using an SVD to get $\mathbf{t}_1$, $\mathbf{t}_2$ as the two vectors spanning this null-space. What is sought is a vector $\mathbf{t} = (t_1, t_2, t_3, t_4, t_5)$ corresponding to the five unknowns in equation D.2. Rearranging D.2 in terms of the image coordinates of the fifth and sixth points gives:

$$\begin{bmatrix} x_5^i & y_5^i & w_5^i \end{bmatrix} \begin{bmatrix} 0 & t_5 - t_1 & t_1 - t_4 \\ t_2 & 0 & t_4 - t_2 \\ -t_3 & t_3 - t_5 & 0 \end{bmatrix} \begin{bmatrix} x_5^i \\ y_5^i \\ w_5^i \end{bmatrix} = 0 \tag{D.3}$$

It can be seen that the determinant of the matrix in equation D.3 is zero, giving the following constraint on the elements of $\mathbf{t}$:

$$t_1 t_2 t_5 - t_2 t_3 t_5 - t_2 t_4 t_5 = t_1 t_3 t_4 - t_2 t_3 t_4 - t_3 t_4 t_5$$

Since $\mathbf{t}_1 + \alpha\mathbf{t}_2 = \mathbf{t}$, this constraint gives a cubic polynomial in terms of the unknown scaling α which can be solved to give 1 or 3 real solutions, and hence one or three solutions for $\mathbf{t}$. Once $\mathbf{t}$ has been obtained, $(X, Y, Z, W)^T$ can be recovered as follows:

$$\frac{X}{W} = \frac{t_4 - t_5}{t_2 - t_3}\frac{Y}{W} = \frac{t_4}{t_1 - t_3}\frac{Z}{W} = \frac{t_5}{t_1 - t_2}$$

In the situation where $W = 0$, the sixth point is on the plane at infinity, and it is possible simply to rearrange the ordering of the points so that a different point is the sixth point. Given $(X, Y, Z, W)^T$, equation D.1 provides a set of linear constraints on the camera matrices $\left(\alpha^i, \beta^I, \gamma^i, \delta^i\right)$. The three camera matrices are then obtained as:

$$P^i = B^{-1}\begin{bmatrix} \alpha^i & 0 & 0 & \delta^i \\ 0 & \beta^i & 0 & \delta^i \\ 0 & 0 & \gamma^i & \delta^i \end{bmatrix}$$

and the tensor may be recovered directly using equation 4.18 described on page 93.

Appendix E

Self-Calibration

A lot of this work has focused on producing a 3D reconstruction of a scene that is defined subject to an arbitrary projective transformation. Whilst this is useful for many applications, in order to view, manipulate or measure the scene using existing hardware and software, it is usually necessary to upgrade this reconstruction to metric. Although the theoretical possibility of recovering intrinsic camera parameters has been known since the beginning of the 20th century, translating it into a working implementation has proved very difficult. As such, there is extensive work on the subject, too extensive to review here. Instead, only the method used in this work will be presented, and the interested reader is referred elsewhere for more details on the topic of self-calibration (for example [HZ00, Pol99, Arm96]).

E.1 Preliminaries

Key to all self-calibration algorithms is some means of relating Euclidean, projective and possibly affine cameras. To facilitate this, a brief overview of how cameras for different geometries can be related will be given. Unlike previous discussions on this subject (such as section 3.2.6 on page 49), this will make use of homogeneous notation to give a more concise description.

E.1.1 Projective Cameras

It is possible to decompose a projective camera matrix P_{Pn} for an image n as:

$$P_{Pn} \simeq [H_{1n}|\mathbf{e}_{1n}] \tag{E.1}$$

where H_{1n} is an inter image homography that transforms points on some reference plane between the first and nth image, and $\mathbf{e}_{1n}$ is the projection of the 1st cameras centre in the image n. Any reference plane is valid in equation E.1 provided that the same reference plane is used for all views and the following transformation can be applied at will to all cameras in order to change the reference plane:

$$\mathbf{T} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ \rho a1 & \rho a2 & \rho a3 & \rho \end{bmatrix} \tag{E.2}$$

where ρ represents an arbitrary change in scale. Given the decomposition of P_{Pn} in equation E.1, this means homographies for different planes are related as follows:

$$H'_{1n} \simeq H_{1n} - \mathbf{e}_{1n} \cdot \mathbf{a}^T \tag{E.3}$$

for $\mathbf{a} = (a1, a2, a3)$ and ρ absorbed into the arbitrary scaling. See section 3.6.1 on page 65 for more details on the relationships between inter image homographies and camera matrices.

E.1.2 Affine Cameras

The affine representation of a camera is the same as the projective one, but with the inter image homography H_{1n} due to the plane at infinity $H_{\infty 1n}$. Therefore, the camera matrices can now be decomposed as:

$$P_{An} \simeq [H_{\infty 1n} | \mathbf{e}_{1n}]$$

This corresponds to transforming the projective camera matrices as in equation E.3 using some particular $\mathbf{a}$ which makes the reference plane the plane at infinity. The same transformation can be applied to all cameras, and so in general affine cameras can be obtained from projective ones with the following transformation:

$$P_{An} \simeq P_{Pn} T$$

for some T as defined as in equation E.2 which makes the reference plane the plane at infinity.

E.1.3 Euclidean Cameras

To transform projective camera matrices into a Euclidean form requires that a full projectivity H be applied to all cameras P_{Pn} and structure $\mathbf{X}_{Ei}$ as:

$$P_{En} \simeq P_{Pn} H \qquad \text{(E.4)}$$
$$\mathbf{X}_{Ei} \simeq H^{-1} \mathbf{X}_{Pi}$$

The result will be Euclidean camera matrices. Since they are Euclidean, it follows that it should be possible to decompose the camera matrix as a non-Euclidean calibration matrix K_n and a Euclidean transformation giving the camera orientation:

$$P_{En} \simeq K_n \left[R | -R\mathbf{t} \right] \qquad \text{(E.5)}$$

Given $P_{En} = [A|\mathbf{a}] = [K_n R | -K_n R\mathbf{t}]$, this can be achieved by the use of RQ decomposition on A to decompose it into an upper triangular matrix (K_n) and an orthogonal matrix (R). Finally, $\mathbf{t}$ can be obtained as $-A^{-1}\mathbf{a}$.

E.2 Self-Calibration

Although the idea of self-calibration has been around for some time, the first real self-calibration algorithm is usually attributed to [FQM92]. Early works primarily considered the case of constant internal camera parameters and bundle adjustment like methods (for example [Har94b]). Later works studied specific camera motions such as pure rotation and translation [MGDP94, Har94c] or stereo rigs [ZBR95].

Later work by Triggs [Tri97] introduced the absolute dual quadric as a tool for self-calibration. This was later refined in the works of [HÅ97] and [PKG97]. The technique presented here is a refined version of the technique [PKG97] presented in [HZ00].

E.2.1 Absolute Dual Quadric

The basis of the technique presented here is that in space one degenerate dual (i.e. plane) quadric exists which is fixed under all Euclidean transformations. This quadric is called the absolute dual quadric (The dual of the absolute conic discussed briefly in section 2.4.4). It is usually written as Ω^* and represented with a 4x4 symmetric matrix of rank 3:

$$\Omega = \begin{bmatrix} I_3 & \mathbf{0}_3 \\ \mathbf{0}_3^T & 0 \end{bmatrix}$$

This is the dual of the quadratic point equation $x^2 + y^2 + z^2 = 0$ which is a circle of radius $\sqrt{-1}$ in the plane at infinity. Of particular importance is the projection of the absolute conic into the images to give the dual image of the absolute quadric ω_n^* in image n:

$$\omega_n^* = P_{En}\Omega^* P_{En}^T$$

Note that projective transformations H are applied to a dual quadric as $\Omega^* = H\Omega^* H^T$. If the decomposition of the Euclidean cameras as in equation E.5 is then substituted into this equation and worked through, it can be found that:

$$\omega_n^* \simeq \mathbf{K}_n \mathbf{K}_n^T$$

Note how the invariance of the absolute quadric to the Euclidean component of the camera matrices eliminates the rotation and translation component, leaving only the non-Euclidean camera calibration.

This projection of the quadric can be used to obtain constraints on the reconstruction by projecting the absolute dual quadric with projective cameras that are modified to be Euclidean (as in equation E.1):

$$\omega_n^{-}1 \simeq P_{Pn} H\Omega^* H^T P_{Pn}^T \tag{E.6}$$

This provides constraints on the calibrating homography H. Once H has been determined, it can then be applied to the cameras and structure as in equation E.4 to upgrade the reconstruction to Euclidean.

E.2.2 Nonlinear Method

Given the relation in equation E.6, it is possible to derive a non-linear equation in terms of the unknown calibration matrices and the unknown projective quadric $H\Omega^* H^T$. Both these should be parameterised in a minimal manner. $H\Omega_n^* H^T$ should be parameterised using a minimum of 8 parameters by imposing the symmetry, scale factor and rank 3 constraints. This can be achieved by simply setting Ω_{33} to 1 and calculating Ω_{44} from the rank 3 constraint $(\det\Omega = 0)$.

Similarly, the upper triangular calibration matrices can be parameterised using a minimal 5 parameters by setting K_{n33} to 1. However, this represents a practically inhibiting $8 + 5n$ unknowns where n is the number of images. To remedy this, certain assumptions can be made about the camera calibration matrix. In particular, with most high quality modern

Restriction on K	Constraint on ω^*	type	No. Constraints
zero skew	$\omega^*_{12}\omega^*_{33} = \omega^*_{13}{\omega_{23}}^*$	quadratic	n
principle point at origin	$\omega^*_{13} = \omega^*_{23} = 0$	linear	2n
zero skew and principal point at origin	$\omega^*_{12} = 0$	linear	n
fixed (unknown) aspect ratio with zero skew and principal point at origin for cameras i and j	$\frac{\omega^*_{i11}}{\omega^*_{i22}} = \frac{\omega^*_{j11}}{\omega^*_{j22}}$	quadratic	n-1
known aspect ratio r with zero skew and principal point at origin	$r^2\omega^*_{11} = \omega^2_{22}$	linear	n

cameras, it can safely be assumed that the skew is zero and that the camera centre is in the middle of the image.

Assuming the image coordinate system has been altered so that the camera centre is at $(0, 0)$, this means the calibration matrices K_n now have the simplified form:

$$K_n = \begin{bmatrix} f_n & 0 & 0 \\ 0 & f_n & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Given these parameterisation for $\mathbf{K}_n$ and Ω^*_n the following criterion can then be minimised using any standard nonlinear minimisation technique (such as Levenberg-Marquardt):

$$\min \sum_{k=1}^{n} \|\lambda_n K_k K_k^T - P_{Pk}\left(H\Omega^* H^T\right) P_{Pk}^T\|^2 \tag{E.7}$$

The unknown scale factors λ_n can be eliminated by using a matrix norm so that both $K_k K_k^T$ and $P_{Pk}H\Omega^* H^T P_{Pk}^T$ have a frobenius norm of 1.

E.2.3 Linear Method

Although a nonlinear minimisation has been presented, it does not represent a feasible self-calibration algorithm without some means of initialisation. This can be achieved, by re-examining the effect of the assumptions on camera form on the image of the absolute dual

quadric $\omega_n^* \simeq K_n K_n^T$:

$$\omega_n^* = \lambda \begin{bmatrix} f_n & 0 & 0 \\ 0 & f_n & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} f_n & 0 & 0 \\ 0 & f_n & 0 \\ 0 & 0 & 1 \end{bmatrix}^T = \begin{bmatrix} \lambda f_n^2 & 0 & 0 \\ 0 & \lambda f_n^2 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

This enables certain extra constraints to be imposed per image based on the form of the matrix rather than the full measure in equation E.7. In particular, it can be enforced that $\omega_{11}^* = \omega_{22}^*$ and that $\omega_{12} = \omega_{13} = \omega_{23} = 0$. These constraints can be transformed to constraints on the form of $P_{Pk}\left(H\Omega^* H^T\right) P_{Pk}^T$ to give 4 constraints per image.

Different forms of these constraints (as well as some nonlinear ones) are also available if different assumptions are made about the camera, and table E.2.3 gives a complete summary of these constraints. For more details see [HZ00]. On the whole, the best algorithm for the case in hand will be the one that makes the most assumptions it can about the camera form, given the camera being used. For the video sequences used in this work, it was found that known aspect ratio, zero skew and principal point at image centre could safely be assumed.

E.2.4 Alternative Nonlinear Method

In practise the linear approach does not allow all the constraints listed in table E.2.3 to be enforced. Subsequently, a nonlinear minimisation is carried out that enforces all the possible constraints. Note that there is little point in minimising the nonlinear criterion in equation E.7 because the quantity being minimised is meaningless anyway so it does not necessarily represent a better solution.

For best results, and to minimise something meaningful, a constrained Euclidean bundle adjustment (see section B.2, page 252) is performed in which the camera parameters are allowed to change, but are constrained sensibly. This Euclidean bundle adjustment is not strictly necessary to get a reasonable solution in many cases, but can help resolve problems in camera localisation, in particular the trade off between focal length and movement along the optical axis.

E.2.5 Upgrading to Metric

Once the projective form of the absolute dual quadric $H\Omega^* H^T$ is known, it becomes necessary to identify the projective transformation H that will take the conic to canonical form. This is achieved by decomposing the absolute dual quadric using SVD into the product of three

matrices UWV^T. Because the absolute dual quadric is symmetric $U = V$, and because it is rank 3 $W = (\lambda_1, \lambda_2, \lambda_3, 0)$ assuming the singular values are arranged in descending order. This leaves a decomposition of the form UWU^T. Note that, if the smallest singular value is not 0, it can be set to 0 to enforce the rank 3 constraint. A similar transformation can be determined using eigen decomposition of Ω^*, bearing in mind that singular values are the squared eigenvalues.

Appendix F

Cross Correlation and Box Filtering

F.1 Cross Correlation

At the core of many feature based matching methods is some method for obtaining the similarity of the features in a potential match - a correlation score. Consequently, it will be worth discussing the process of window-based cross correlation in detail, and in particular the zero mean normalised cross correlation.

Some details about cross correlation have already been alluded to in chapter 8. Of particular relevance is the discussion in section 8.1.1, p162 concerning the modelling of image effects due to camera motion with very simple models of image motion. Cross correlation is usually built around the simplest of these models: pure translation. This assumes that, in a localised region, all points in the matching region of another image will have undergone a constant translation (see figure F.1). This assumption is not always well founded, particularly if there is significant camera rotation or the regions being imaged exhibit a lot of motion parallax.

There are many different measures of correlation that can be applied using the cross correlation approach, but for this work, a zero mean normalised cross correlation score will be discussed. Specifically, this bases similarity between a pair of candidate matches $\mathbf{x}_1$ and $\mathbf{x}_2$, on the image intensities in a rectangular correlation window of size $(2n+1) \times (2m+1)$ centred on the features. The zero mean normalised correlation score is then calculated as

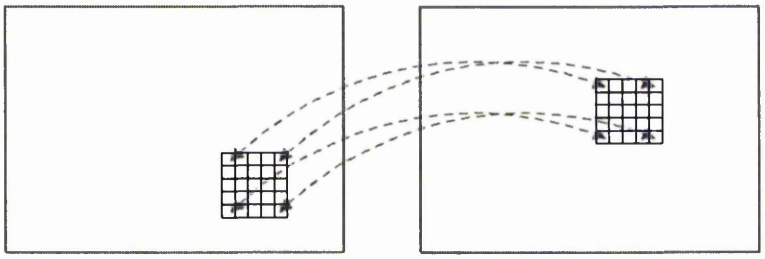

Figure F.1: The usual assumption applied to window based cross correlation of a pure translation being applied to the window in the first image to give the window in the second image

follows:

$$\text{Correl}\left(\mathbf{x}_1, \mathbf{x}_2\right) = \frac{\sum_{i=-n}^{n}\sum_{j=-m}^{m}\left[I_1\left(u_1+i, v_1+j\right) - \overline{I_1\left(u_1, v_1\right)}\right]\left[I_2\left(u_2+i, v_2+j\right) - \overline{I_2\left(u_2, v_2\right)}\right]}{\left(2n+1\right)\left(2m+1\right)\sqrt{\sigma_1^2\left(u_1, v_1\right)\sigma_2^2\left(u_2, v_2\right)}} \quad \text{(F.1)}$$

where $I_k\left(x, y\right)$ is a function giving the intensity of the pixel at position x, y in the kth image and:

$$\overline{I_k\left(u, v\right)} = \frac{\sum_{i=-n}^{n}\sum_{j=-m}^{m} I_k\left(u+i, v+j\right)}{\left(2n+1\right)\left(2m+1\right)}$$

$$\sigma_k\left(u, v\right) = \sqrt{\frac{\sum_{i=-n}^{n}\sum_{j=-m}^{m} I_k^2\left(u+i, v+j\right)}{\left(2n+1\right)\left(2m+1\right)} - \overline{I_k\left(u, v\right)}}$$

are the mean and standard deviation of the image intensities in the correlation window.

To sum up, this measure is a normalised summation of the squared differences between relatively corresponding points in a window centred on the potential match. The result will be a value between -1.0 and 1.0, with negative values indicating worse scores. The normalisation is very useful because, if a raw sum of squared differences is used, then for example a threshold on quality of matches needs to be adapted to the amount of variation in the images and to the number of bits per pixel used to represent the image.

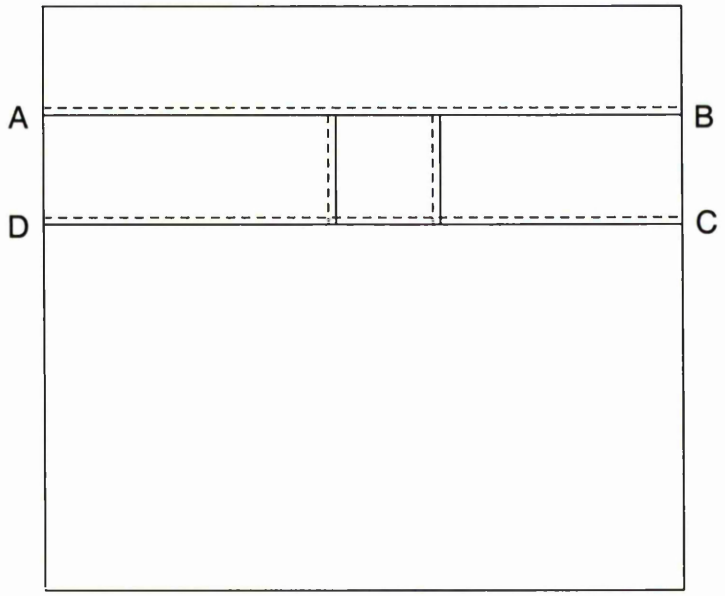

Figure F.2: Box filtering. A window is passed over the image (ABCD) and summation maintained for all columns in the window. As the window moves down, the rows indicated by dashes are added and removed. The same applies to movement along the rows.

F.2 Box Filtering

Box filtering is a very well established method for high speed digital filtering of data [McD81]. The process allows a $(2n+1) \times (2m+1)$ region around a point (u, v) to be summed for a function $f(d)$ as follows:

$$b(u,v) = \frac{\sum_{i=-n}^{n} \sum_{j=-m}^{m} f(I(u+i, v+j))}{(2n+1)(2m+1)}$$

Assuming images with integer values are used, both the numerator and denominator here can be evaluated using integer arithmetic, with the division rounded to the nearest integer for extremely fast operation. Since this form of filtering is frequently applied to every point (e.g. in an image), box filtering capitalises on the repeated summation for overlapping sections. For example, if a sum for the box centred at (u, v) is found then the sum for the box at $(u+1, v)$ will be exactly the same, but less the column at $u-n$ and plus the column at $u+n+1$. Naturally, the same applies to movement down columns.

To achieve this optimisation on both columns and rows, the filter is applied first across

and then down the image. Figure F.2 illustrates how the optimisation is achieved. Starting from the top, a window ABCD is placed over the image I centred on column v. A buffer IBUF(n) is maintained which contains the sum of the corresponding columns in the window ABCD. At the start of the filtering operation, IBUF is filled using a complete summation, but for subsequent rows, the window is moved down and IBUF updated by subtracting the top row $v-m$ and adding the new bottom row $v+m+1$ (as indicated by the dotted lines).

A similar process is used to maintain the complete value when moving the window along the row. In this case a single value ISUM is maintained that is the sum of the relevant IBUF entries. The box filter result is obtained from this as:

$$b(u,v) = \frac{ISUM}{(2n+1)(2m+1)}$$

As the box moves along the row to $(u+1,v)$, ISUM is updated by adding the next value at IBUF($u+n+1$) from ISUM and subtracting the old value at IBUF($u-n$).

This provides a very brief overview of box filtering. For more precise details, including a review of different image boundary handling strategies, the reader is referred to the work of [McD81].

F.2.1 Application to Cross Correlation

Applying box filtering to the zero mean normalised cross correlation discussed earlier is fairly straightforward and involves breaking the cross correlation equation F.1 into components involving summation. Each summation can then be handled using a box filter.

There are three summations in the equation which can be handled with a box filter. The first and fairly straightforward ones involve the calculation of mean and variance for each pixel in the images. On top of these, the numerator of the correlation function can then be separated into:

$$\text{Correl}(\mathbf{x}_1,\mathbf{x}_2) = \left(\sum_{i=-n}^{n}\sum_{j=-m}^{m} I_1(u_1+i,v_1+j)\,I_2(u_2+i,v_2+j)\right) - (2n+1)(2m+1)\,\overline{I_1(u_1,v_1)I_2(u_2,v_2)}$$

and the summation determined using a box filter over a special image created by multiplying both images.

For a more exacting description of how box filtering may be applied to cross correlation, the reader is referred to [Sun97]. The process is also hinted at in [McD81].

Appendix G

The Sofa Image Sequence

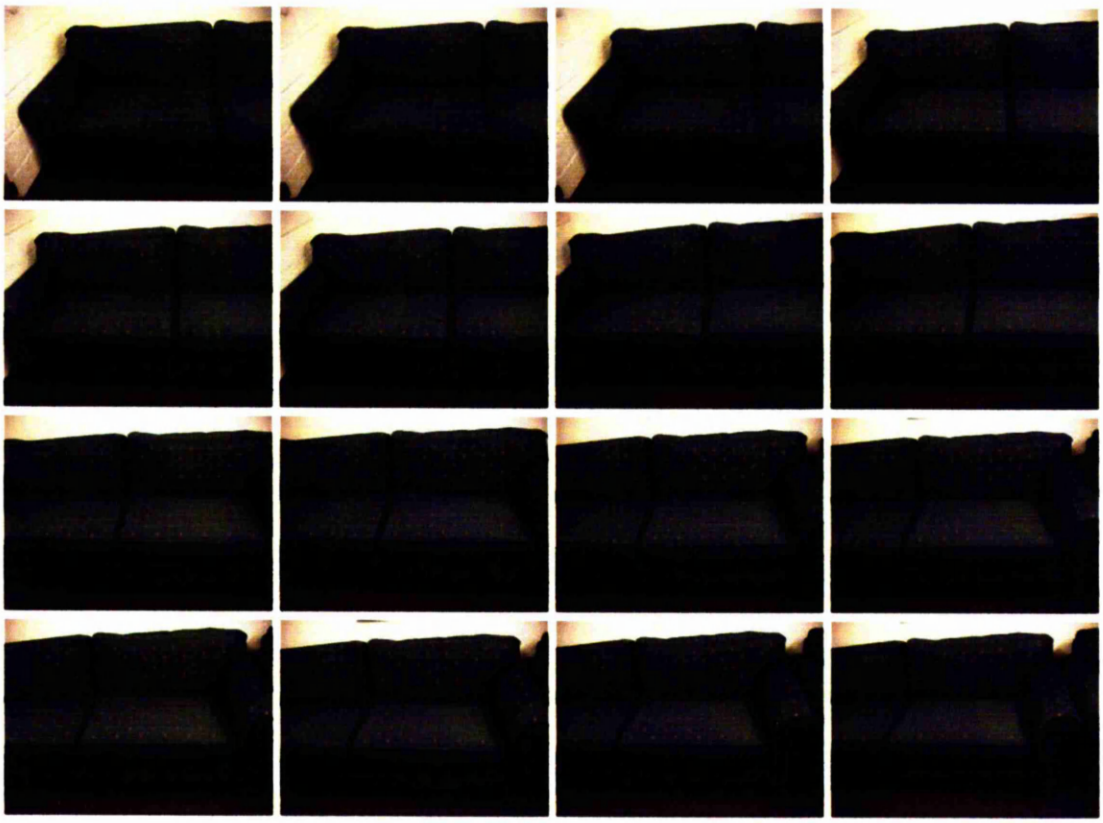

Figure G.1: The sofa sequence of 327 images. Sampled at roughly every 10 images and including the first and last frames. Images 0 - 170.

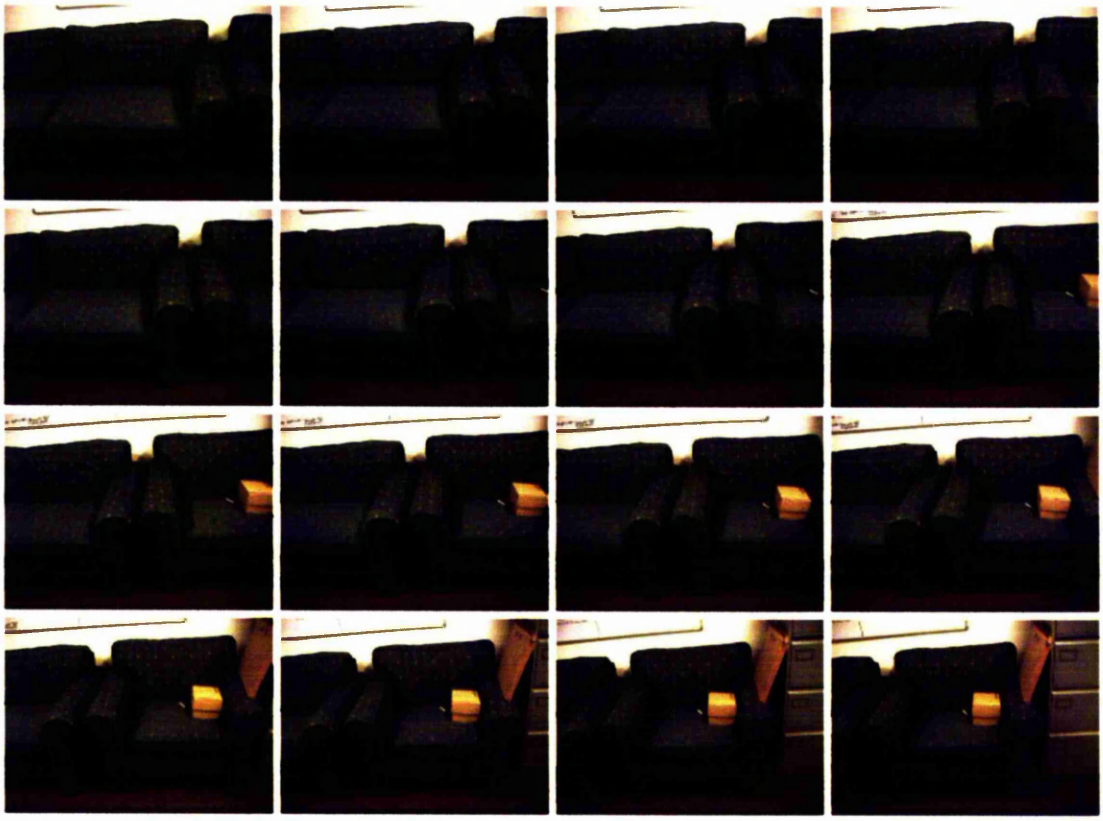

Figure G.2: The sofa sequence of 327 images. Sampled at roughly every 10 images and including the first and last frames. Images 180 - 327.

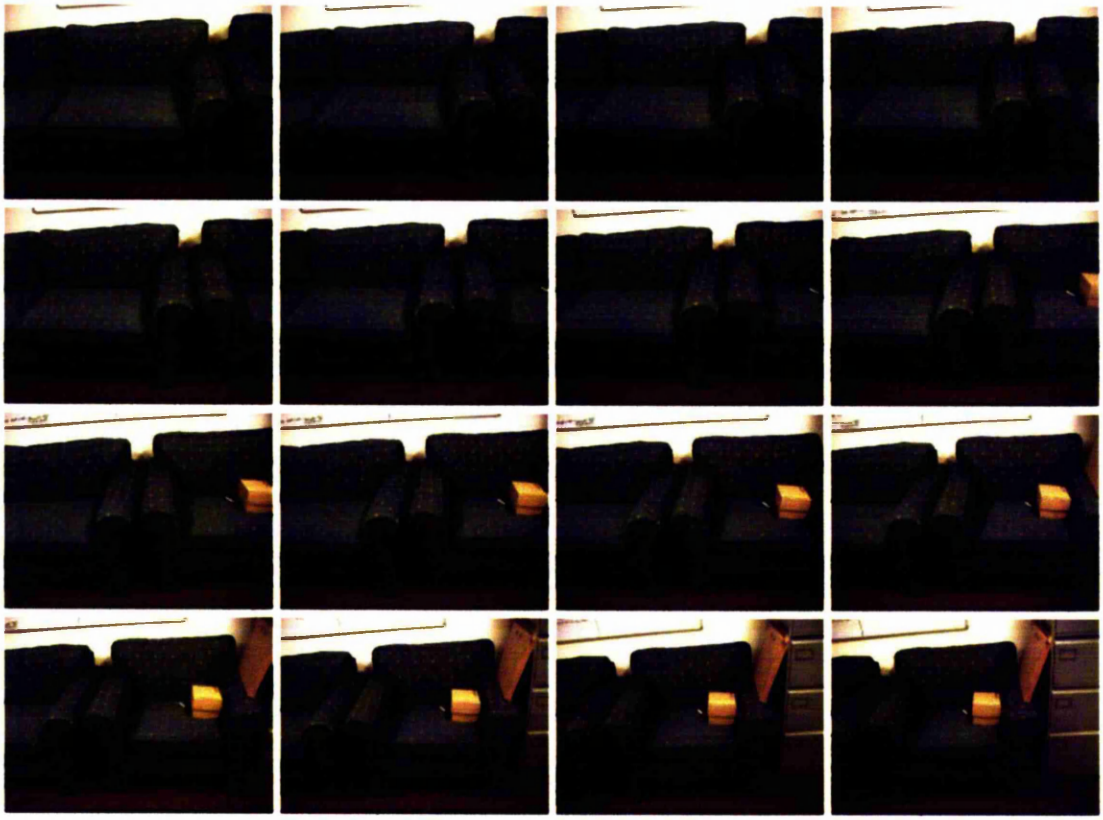

Figure G.2: The sofa sequence of 327 images. Sampled at roughly every 10 images and including the first and last frames. Images 180 - 327.