Recovering Feature and Observer Position by Projected Error Refinement

Gareth S. Bestor

Technical Report #1381

August 1998

RECOVERING FEATURE AND OBSERVER POSITION BY PROJECTED ERROR REFINEMENT

by

Gareth S. Bestor

A dissertation submitted in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

(Computer Sciences)

at the

UNIVERSITY OF WISCONSIN - MADISON

1998

Everything should be made as simple as possible, but not simpler.

- Albert Einstein

Abstract

Recovering three-dimensional information from images is a principal goal of computer vision. An approach called *Structure From Motion* (SFM) does so without imposing strict requirements on the observer or scene. In particular, SFM assumes camera motion is unknown and the scene is only required to be static. This thesis describes a new SFM technique called *Projected Error Refinement* that computes the positions of feature points (i.e., *structure*) and the locations of the camera or observer (i.e., *motion*) from a noisy image sequence. The technique addresses limitations of existing SFM techniques that make them unsuitable except in controlled environments; the approach presented in this thesis models perspective projection, allows unconstrained camera motion, deals with outliers and occlusion, and is scalable. This new technique is recursive and thus is suitable for video image streams because new images can be added at any time.

Projected Error Refinement views SFM as a geometric inverse projection problem, with the goal of determining the positions of the cameras and feature points such that the *projectors* defined by each image optimally intersect (projectors are the lines of projection specifying the direction of each feature point from the camera's optical center). This is expressed as a global optimization problem with the objective function minimizing the mean-squared *angular projection error* between the solution and the observed images. Occlusion is dealt with naturally in this approach because only visible feature points define projectors that are considered during optimization - occluded features are ignored. The technique models true perspective projection and is scalable to an arbitrary number of feature points and images. Projected Error Refinement is non-linear and uses an efficient *parallel iterative refinement* algorithm that takes an initial estimate of the structure and motion parameters and alternately

refines the cameras' poses and the positions of the feature points *in parallel.* The solution can be refined to an arbitrary precision or refinement can be terminated prematurely due to limited processing time. The solution converges rapidly towards the global minimum even when started from a poor initial estimate. Experimental results are given for both 2D and 3D perspective projection using real and synthetic images sequences.

Acknowledgments

My graduate studies at the University of Wisconsin-Madison can best be described by the words of John Lennon - "Life is what happens to you while you're busy making other plans." Starting with only a vague idea of getting a Ph.D., I formed ties with people along the way that will forever influence my life, and not just academically. My thanks must first go to my advisor, Professor Chuck Dyer, for introducing me to computer vision and for his guidance over the years in my pursuit of research in this field which continues to fascinate me. I am also grateful for his financial support which allowed me to focus on finishing my degree. My thanks also go to Professor Mike Bleicher for turning me into more of a mathematician and geometer than I ever anticipated. I would have floundered long ago were it not for his willingness to answer my constant barrage of questions. I am also deeply thankful to Professor Nicola Ferrier for her enthusiasm and assistance in refining my thesis. Her insight into Structure From Motion was invaluable and our discussions were the most rewarding and productive of my graduate career and a constant source of inspiration. Most of all I would like to thank her for saying "I think you're smart" when I needed it most. My thanks also go to Professors Jude Shavlik and Vadim Shapiro for taking the time to serve on my thesis committee.

I am most fortunate to have two families. First, I would like to thank my parents, Tom and Ellie, for nurturing my inquisitiveness and exposing me to computers at a young age, which sealed my fate. I am also indebted to my 'surrogate' family Don Vedder, Tekla, Rita and Woody Wlodarczyk, who accepted me into their lives the day I arrived in Madison, fed me, gave me a couch to sleep on, and taught me everything I needed to know about volleyball, zucchini and groundhogs. Without their constant distractions I would have finished my degree

much sooner, and been the worse off for it. Some of the blame must also go to my good friend and partisan Michael Giddings, who introduced me to mountain biking, kayaking, NeXT and welding. My sincerest thanks go to Susan Dinan for sharing the best and worst periods of graduate school. I cannot imagine those years without her being there. I am also forever grateful to Mary Janowiak, who opened my eyes to worlds I would never otherwise have known and from whom I have learned so much. I most cherish the constant friendship and support of Tara Treichel over the last few years. For reasons I will never understand she put up with me during the final and hardest stages of my Ph.D., when everything else became secondary but when I most needed the support of others. Finally, this thesis to dedicated to Mark Winfield, who always believed in me.

Table of Contents

List of Figures

List of Tables

Chapter 1
Introduction

Vision is our most important sense and provides us with the richest source of data about our environment. One of the primary goals of a vision system is to determine the structure of the environment and to locate the observer within it. The difficulty in visually determining scene structure and observer location is that optical projection is a destructive transformation - the image projected onto the retina or camera is purely two-dimensional and contains no explicit three-dimensional information. The inverse projection problem, namely recovering 3D structure from 2D images, is fundamentally ill-posed and additional assumptions or constraints must be made for the problem to become well posed. The remarkable proficiency of the human visual system is evidence that inverse projection is not a hopeless cause.

The assumptions made in computer vision provide a useful basis to distinguish different techniques. For example, stereo vision assumes the relative positions of two cameras is precisely known [12], whereas techniques that infer local shape from image contours presume physical properties of the scene such as smoothness [25], [69]. An approach called *Structure From Motion* (SFM) makes minimal assumptions about the camera and scene and does not require specialized hardware (Shariot and Price [49] and Ullman [67] summarize earlier work in SFM, and Oliensis [39] gives a critique of recent techniques). Ideally, SFM allows arbitrary camera motion and only requires that the scene is static. However, many SFM techniques, such as the Factorization Method and its derivatives, assume a non-perspective projection model and therefore are only accurate for specific camera motions and scene structures [59], [5], [42], [46], [70], [60]. This thesis describes a new SFM technique called *Projected Error*

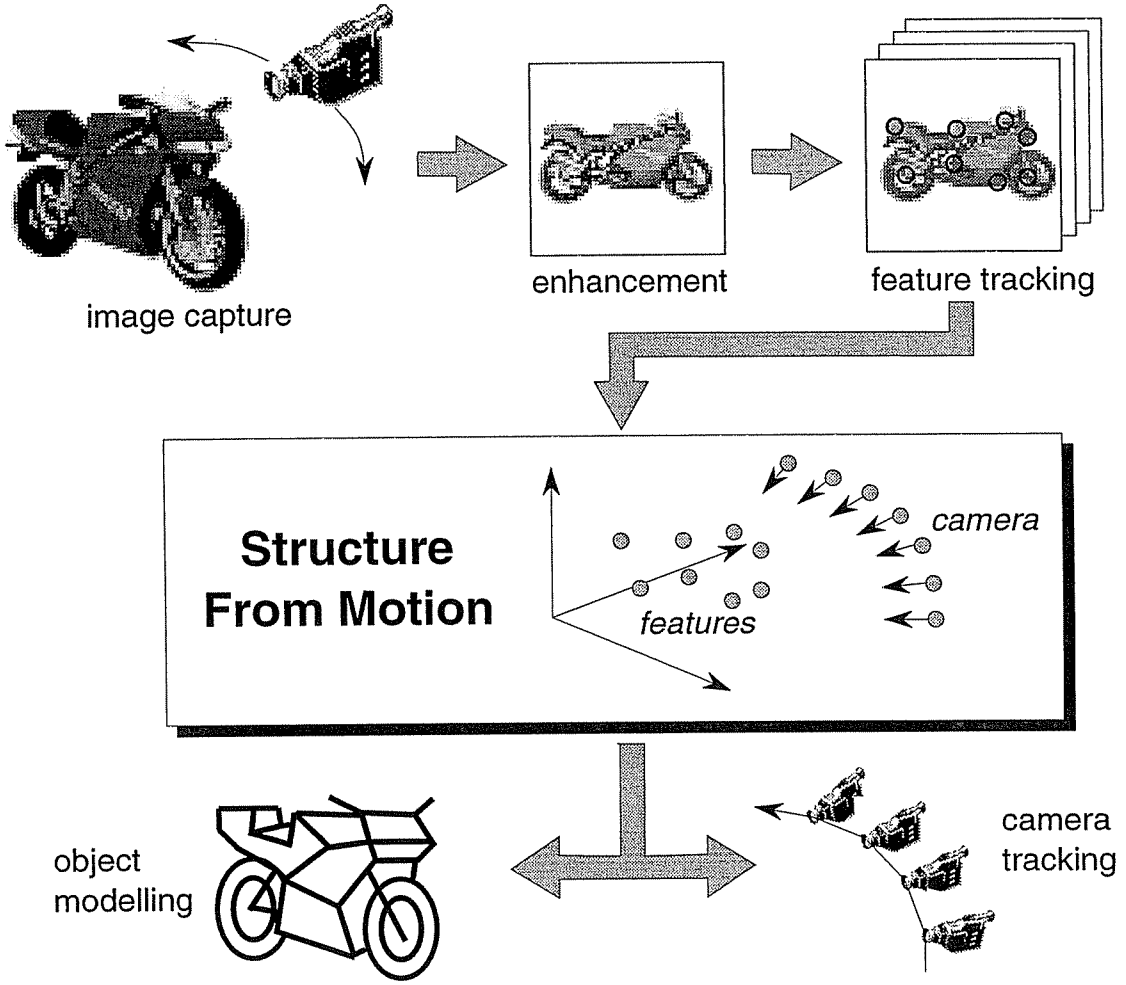


Figure 1.1: The vision pipeline includes multiple stages: image capture and digitization, image enhancement, feature detection and tracking, recovering 3D structure, i.e., *Structure From Motion*, surface fitting and object modelling, and camera tracking and navigation.

Refinement that recovers the positions of feature points and the locations of the cameras, and avoids some of the additional constraints imposed by other techniques on the inverse projection problem.

1.1 Problem Description and Assumptions

Recovering 3D structure and motion from images is a multi-stage process of which SFM is but one part, as shown in Figure 1.1. SFM occurs after images are captured, digitized and prominent features in each image identified and matched with the features in the other images.

The input to SFM is typically a list of features and their projected 2D positions in each image, and the output is the 3D positions of these feature points and the 3D locations of the cameras and the rotations of the images around the optical centers.[1] Surfaces may be subsequently fitted to the recovered points to construct a three-dimensional model of the scene for use in object recognition and computer graphics (e.g., wireframe models [3]). The recovered positions of the camera may be used for navigation and obstacle avoidance (e.g., motion planning [27]).

Structure From Motion techniques generally make the following assumptions:

- the extrinsic and possibly intrinsic parameters of the camera are unknown,
- the scene is static; called the *rigidity constraint,*
- multiple images are projected from different viewpoints,
- the features extracted from the images are primitives such as points or lines,
- the correspondence of features between the images is known; i.e., the *correspondence problem* is solved.

The most restrictive of these assumptions is the rigidity constraint. Rigidity is necessary for image consistency because the images are not captured at the same time, unlike say stereo vision where images are taken simultaneously by two cameras. If the scene changed arbitrarily between each view then the projected images are unrelated, making inverse projection ill-posed. There are many applications where the scene can be assumed to be static, in particular object modelling [3], [47]. It is also possible to relax the rigidity constraint if non-

[1]Most SFM techniques use point features. Those based on line features have somewhat different input and output forms [28], [7], [71], [26].

rigid motion can be identified and segmented by other means [33], [5], [58].

SFM also assumes that the correspondence problem is solved. Determining feature correspondences is non-trivial because the exact camera motion between each image in the sequence is unknown. Features can be tracked reliably between images if the frame rate is high relative to camera motion [52], [76]. Nevertheless, feature tracking is an active area of research and correspondence errors do occur, which, if undetected, can overwhelm subsequent SFM analysis and render the solution meaningless. Detecting correspondence errors is an important feature of Projected Error Refinement.

Many SFM techniques assume the intrinsic camera parameters are known (i.e. focal length, aspect ratio, principle point and the angle between the image axes), called *intrinsic camera calibration*. Several SFM techniques based on projective geometry determine these parameters in the course of solving for structure and motion [34], [41], [2]. Camera calibration is important and self-calibration is obviously desirable, however, it is somewhat distinct from the problem of recovering structure and motion. For example, calibration is easiest when camera motion is rotational, whereas recovering structure and motion strictly requires camera translation [18]. In many cases, the intrinsic camera parameters are constant and can be determined beforehand [18], [30] or they change smoothly and can be dynamically adjusted [32], [41]. Camera calibration is not examined in this thesis and it is assumed that the intrinsic camera parameters are found by other means.

1.2 Motivation

SFM is a powerful approach to computing scene structure and camera motion because it simultaneously recovers both the positions of features and the location of the observer, and it makes fairly minimal assumptions and thus is suitable for a wide variety of applications. However, existing SFM algorithms impose additional constraints to reduce the problem to linear complexity, do not scale to large numbers of features or images, or are not robust to the irregularities and imperfections of real image sequences, such as image noise, occlusion, missing

features and correspondence errors. As a result, existing SFM techniques typically do not perform well in real applications [62], [39]. This research was motivated to address the limitations of existing SFM techniques. In particular, five key deficiencies are identified:

10. An accurate optical projection model. This implies at least perspective projection. Parallel projection, weak perspective and para-perspective projection are only accurate for limited camera motions and scene structures.

11. Robust to noise caused by an imperfect camera model or introduced in prior stages of the vision pipeline. This requires examining *all* the available image information.

12. Handle occlusion and missing features. Occlusion is an innate property of real scenes.

13. Handle correspondence errors. Feature detection and tracking cannot be expected to give perfect correspondences. Occasional correspondence errors must be handled, which may manifest as gross errors in the projected positions of some feature points.

14. Recursive. Techniques that can incrementally refine an existing solution by adding new images are most suited to real-time video image processing. Recomputing the solution from scratch for every new image is inefficient.

1.3 Projector-based Image Representation

Selecting the representation of images is important because it largely determines the complexity and capabilities of the solution. More precisely, for the inverse projection problem, there is no compelling reason to use planar images (a similar argument is made by Naeve and Eklundh [36] for using projective geometry). Projected Error Refinement represents

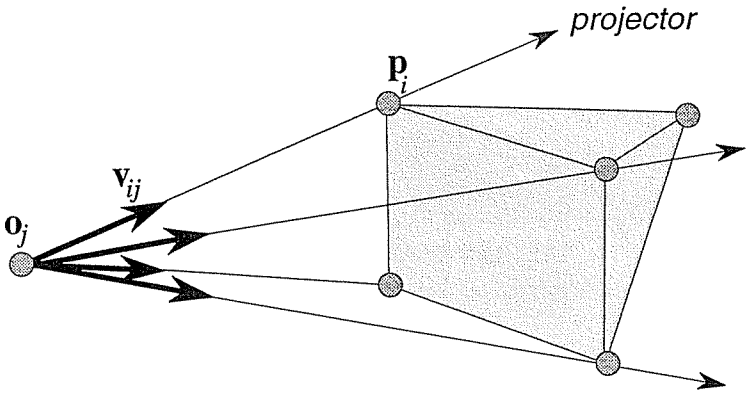


Figure 1.2: A *projector* is the line extending from the optical center $\mathbf{o}_j$ of the camera to the feature point $\mathbf{p}_i$ in the scene. $\mathbf{v}_{ij}$ is the *unit direction vector* of this line and an 'image' is the set of direction vectors for all the visible features. The only available information about the structure of the scene is the *relative* direction of each feature point from the optical center.

images in terms of *projectors*; that is, the lines of projection extending from the optical centers of the camera and passing through the feature points in the scene, as shown in Figure 1.2. The only information about the 3D position of a feature point is its relative direction from the camera - there is no knowledge of its distance. This is precisely modelled by a projector which describes a line through the scene on which the feature point must lie. Unlike planar image coordinates, projectors depend only on the position of the center of projection and not the rotation of the image around it.[2] This is desirable because only the camera's location provides useful information for recovering 3D structure; i.e., rotating the camera provides no new information about scene structure whereas moving the camera to a different location does. Projectors concisely and explicitly represent the relevant information about scene structure, which leads to a more intuitive understanding of the inverse projection problem and can facilitate simple solutions to otherwise difficult problems, such as occlusion. The projector-based image representation also facilitates outlier detection because a projector is the principal axis of a cone describing the confidence region of a feature's position due to projection noise, as

[2]That is to say, the image coordinate frame is not important and can be normalized.

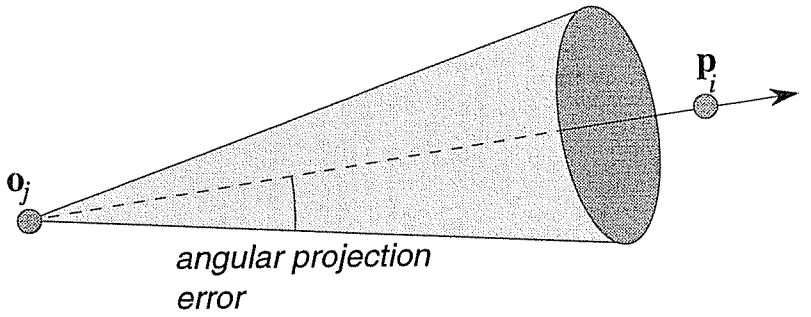


Figure 1.3: A projector is the principal axis of an *cone* describing the expected position of a feature point $\mathbf{p}_i$ due to noise. Note: the error in the displaced position of $\mathbf{p}_i$ due to noise increases the further it is from the center of projection $\mathbf{o}_j$.

---

shown in Figure 1.3. Any projector that lies outside its expected error cone is defined to be an outlier. Projectors can be defined for any geometric projection transformation in any dimension and are easily adapted to different projection models. 2D and 3D perspective projection are examined in this thesis because they are most applicable in computer vision.

The projector-based image representation is equivalent to *normalized projective image coordinates* and *spherical image coordinates* that are used by SFM techniques based on *projective geometry* [76], [50], [17], [2]. As described in Chapter 3, projective image coordinates are essentially a 3D homogeneous representation of 2D planar image coordinates of the form $[x\ y\ f]$, where f is a non-zero scalar multiplier [36], [17]. Spherical image coordinates are similar except that the image is spherical and image coordinates are represented by points on the surface of a *sphere* rather than a plane [31], [23]. Spherical image coordinates are thus isomorphic to unit direction vectors in the projector model. Homogeneous coordinates are frequently used in SFM because they simplify incorporating the intrinsic camera parameters, particularly focal length, into the computation of structure and motion, thereby facilitating self-calibration [2], [41]. Projective geometry also has the useful property that parallel and perspective projection are defined identically [50]. Projective geometry is a well-developed field of mathematics with a strong theory of projection and inverse projection. Although Projected Error Refinement uses a projector-based image representation, it formulates the inverse projection problem

in terms of Euclidean geometry. A topic of future research, described in Chapter 10, is to investigate whether Projected Error Refinement may have a simpler definition when described in terms of projective geometry.

The representational differences between projectors, projective image coordinates, spherical image coordinates and planar images are less significant than how they are used. Most SFM techniques are based on exploiting geometric or algebraic properties that are invariant under projection to multiple images, from which camera rotation and translation is more easily extracted. For example, the *centroid* of several points is constant under parallel projection, and the *cross-ratio* of four collinear points is constant under perspective projection. The *essential matrix* [29], the *fundamental matrix* [17] and the *Factorization Method* [59] all exploit various invariants of perspective and parallel projection to recover the relative extrinsic camera parameters from two or more images. However, these invariant properties are typically defined for a specific number of feature points and images, and therefore these approaches do not permit occlusion or are not easily scalable to additional features and images and thus are sensitive to noise.[3] Projected Error Refinement, on the other hand, optimizes the structure and motion parameters using *all* the visible projectors, thus it supports occlusion, missing features and is arbitrarily scalable.

1.4 Projected Error Refinement

The projectors defined for a feature point by different images are necessarily concurrent; i.e., they all intersect at a single point in the scene. This imposes a weak constraint on the position and rotation of each camera, expressed in terms of the extrinsic camera parameters. A well known result in SFM is that given sufficient number of features and images, there are a finite

[3]The Factorization Method is scalable in the number of features and images, but it assumes parallel projection. Techniques based on the essential and fundamental matrices can be extended to include more feature points but they are only defined for two images, which is inadequate for reliable reconstruction [63], [53], [10]. None of these methods intrinsically handle occlusion or missing features.

number of camera and feature locations that are consistent with the observed images [66], [64], [20], [29], [14], [74]. It is also well understood that these so-called *minimal data solutions* are very sensitive to noise and additional image data must be included for reliability [49], [62], [61], [10]. Redundancy over-determines the inverse projection problem and therefore projectors no longer precisely intersect due to noise. Projected Error Refinement therefore formulates inverse projection as a *optimization problem* for determining the positions of the cameras and feature points such that the projectors defined by the images *optimally* intersect.[4]

The objective error function in Projected Error Refinement is the mean-squared *angular projection error* - the angle between the projectors defined by the observed images and the corresponding projectors in the solution. The angular error of a projector is independent of the distance between the feature point and camera, therefore the further away a feature is from the camera the greater the anticipated error in its position due to noise (see Figure 1.3). As described in Chapter 5, optimization methods that instead minimize the *distance* between projectors are biased against features close to the camera [68], [11], [8], [53]. In Projected Error Refinement the error of individual projectors can be weighted to allow additional information to be included during optimization, such as the strength of a feature match. Occlusion is handled naturally because only visible feature points define projectors whose error terms contribute to the global sum total - missing features are simply ignored. Projected Error Refinement is non-linear and an efficient *parallel iterative refinement* algorithm is used which takes an initial estimate of the structure and motion parameters, as found by a minimal data solution, and then alternately refines the features and images to reduce the angular projection error. Separating the refinement of structure and motion provides a high degree of parallelism - all the feature positions are refined in parallel, as are the camera locations and image rotations. Refinement of the solution can continue to an arbitrary level of precision or can terminate pre-

[4]Optimality is defined in terms of the *angular projection error*; that is, the difference between the observed images and the solution.

maturely, for example, due to limited processing time. Projected Error Refinement is recursive and thus is suitable for real-time video analysis because the projectors from a new image can be added at any time and the solution further refined. Many SFM techniques compute their solution to a fixed precision and must run to completion, or process images in batch mode and therefore must re-compute the solution for every new image.

1.5 Major Contributions

The major contributions of this thesis can be summarized as:

1. A model of SFM based on *projectors* and *angular projection error* that views inverse projection as a geometric problem rather than an algebraic one.

2. A new *minimal data solution* to the inverse projection problem for 2D and 3D perspective projection that gives an intuitive geometric interpretation of the constraints and parameters of the problem.

3. A new optimization-based SFM technique called *Projected Error Refinement* that uses an efficient *parallel iterative refinement* algorithm to optimize scene structure and camera motion by iteratively minimizing the *angular projection error* in the images. This technique models perspective projection and is arbitrarily scalable and robust to noise. Occlusion is supported and outliers are detected and rejected in a well-defined manner.

1.6 Thesis Outline

The organization of this thesis reflects the development of the Projected Error Refinement technique. The input to this method is a labelled list of feature points and their projected positions in each image, expressed as projectors. The output is a 2D or 3D map containing the

recovered positions of the feature points and camera centers and the rotations of the images. No *a priori* knowledge of the positions of the feature points or cameras is assumed and the solution is correct up to a scale factor and a rigid rotation and translation. Results are derived for both 2D and 3D perspective projection.

Chapter 2 reviews related work in SFM. This field has received considerable attention over the years and no attempt is made here to be exhaustive. Rather, the predominant approaches to the inverse projection problem are described, focusing on the assumptions that are made and the practical advantages and disadvantages of each approach.

Chapter 3 describes the projector model of inverse projection on which Projected Error Refinement is based. Different models of optical projection that are used in SFM are compared. The projector model is then used to describe the inverse projection problem for multiple images to identify the constraints and parameters of the problem.

Chapter 4 describes a projector-based *minimal data solution* that examines the minimum number of feature points and images necessary for reconstruction and assumes zero image noise. The effect of noise on this solution is then described and why it is insufficient to examine a fixed minimal number of image measurements; that is, why reliable SFM techniques must be scalable.

Chapter 5 extends the minimal data solution derived in Chapter 4 to examine additional feature points and images. Whereas the former computes the solution that precisely matches the (minimal) images, the so-called *refined solution* instead alternately adjusts the positions of the feature points and camera centers to minimize the angular projection error in the non-intersecting projectors. First, the minimal data solution provides an initial estimate of the positions of a subset of feature points and cameras, which is then augmented with the remaining features and images. This estimate is refined using parallel iterative refinement until further optimization has negligible effect on reducing mean angular projection error. An example is given using synthetic images showing how the refined solution gives a better

reconstruction of the scene than the minimal data solution.

Chapter 6 briefly describes the Kanade-Lucas-Tomasi (KLT) feature tracker for detecting features and tracking them over the image sequence [52], [4]. The KLT tracker obtains good results on a variety of scenes and does not require extensive parameter adjustment. Nevertheless, feature tracking is never perfect and examples are given of the types of errors that can occur.

Chapter 7 examines the problem of *outliers* and describes two complementary approaches to detecting outliers. The first, called *Random Sample Consensus* (RANSAC) [15], grows a minimal data solution by adding only verifiable consistent data points. The second, called *pruning*, computes the solution based on all the data points and then prunes inconsistencies. An explanation is given why RANSAC is unsuitable for outlier detection in the Projected Error Refinement approach and why pruning is used instead. An example is given to show how pruning outliers improves the solution.

Chapter 8 briefly examines the issue of occlusion and missing features, which are handled naturally by Projected Error Refinement. An example shows how missing features affect the accuracy of reconstruction.

Chapter 9 describes experimental results on synthetic and real image sequences. Synthetic images provide ground-truth data and allow quantitative error analysis. A series of experiments on 2D and 3D synthetic scenes show the performance of Projected Error Refinement under different simulated viewing conditions. Several real (3D) image sequences are analyzed to show how the technique performs in a variety of real applications.

Chapter 10 summarizes the main contributions of this thesis and discusses directions for future research.

Appendix A describes *triangulation*, which is used to augment the minimal data solution with additional feature points.

Appendix B describes the 2D *beacon problem* and the 3D *location determination problem*, which are used to augment the minimal data solution with additional images, for 2D and 3D perspective projection respectively.

Appendix C describes the *structure error* and *motion error*, which are the errors in the recovered positions of the feature points and cameras centers, respectively. These terms measure the accuracy of the recovered scene structure and camera motion when ground-truth data is available.

Chapter 2
Related Work

Recovering 3D structure from 2D images is a long-standing problem in computer vision but it was first identified in psychology in the study of human vision and in photogrammetry with determining elevation from aerial photographs. Psychologists observed that the changing appearance of an object undergoing motion is a powerful cue for determining its structure, from which the term *Structure From Motion* originated [16]. An analogous problem arises in photogrammetry when computing elevation from overlapping aerial photographs. The problem in both cases is determining 3D structure, or depth, from 2D projections *when motion is not precisely known*. Thompson [57] showed that two aerial photographs sharing five distinct landmarks is sufficient to determine the relative camera motion by solving a system of five non-linear simultaneous equations, after which the 3D elevation of the landmarks was found by triangulation. Ullman [66] derived an equivalent result and showed that parallel projection requires four points in three images.

SFM continues to be an active area of research and numerous results have been published addressing different aspects of the SFM problem and proposing different solutions. Existing SFM techniques fall into three categories based on their general approach [39]:[5]

1. Image pairs - these techniques examine two images to determine the relative change in the pose of the camera. The positions of the feature points are found by subsequent triangulation.

[5] These categories are not necessarily mutually exclusive.

2. Kalman filtering - these techniques combine partial reconstructions computed from subsets of images (usually two).

3. Optimization - these techniques determine the globally optimal structure and motion parameters by minimizing an error function over a many features and images. *Projected Error Refinement* falls into this category.

For practical purposes, however, what is most important are the assumptions a technique makes and the sensitivity of the solution to common errors. Therefore, after describing SFM techniques based on their algorithmic differences, the requirements of a reliable general-purpose SFM technique are reviewed. The chapter conclude by proposing a two-stage approach satisfying most of these requirements.

2.1 SFM from Two Images

Structure From Motion is a hard problem because both the camera motion and scene structure are unknown. When either is known the problem becomes much simpler; for example, stereo vision assumes calibrated cameras and has a simple linear solution. The most popular approach in SFM has been to determine the change in the pose of the camera between two images by separating the computation of camera motion from that of scene structure. For example, the *essential matrix* [29], [64] depends only on camera motion, not the 3D positions of the features, allowing the relative extrinsic camera parameters of the two images to be recovered independently of scene structure. Similarly, the *Factorization Method* [59] assumes a parallel projection camera model, which allows camera translation to be easily determined and simplifies the problem to one of finding camera rotation.

Recovering structure and motion from two images exploits invariant properties of projection that depend on only a subset of the parameters involved, making the inverse projection problem simpler and, in some cases, linear. Mundy and Zisserman [35] described various invariants of parallel and perspective projection that have been used in SFM.[6] Quan [44],

Shariot and Price [49], Holt and Netravali [20], Longuet-Higgins [29], Tsai and Huang [64], Tomasi [59], Moons *et al.* [34], Shashua and Navab [51] and Shashua [50] described SFM techniques that differ in their camera model, linearity, number of features, and restrictions on camera motion and scene structure. Invariant properties of projected *lines* (as opposed to feature points) have also been explored; for example, Huang and Faugeras [28] gave a non-linear solution for determining camera motion from lines projected to three images, Weng *et al.* [71] gave a linear solution to a similar problem, and Laganière and Mitiche [26] combined lines and point features. Other SFM techniques have incorporated image *velocity* information, e.g., Ullman [67] and Sawhney *et al.* [45].

Many SFM techniques are based on the *essential matrix* derived by Longuet-Higgins [29] and independently by Tsai and Huang [64]. The essential matrix describes a polynomial invariant derived from two uncalibrated images that share eight feature points under perspective projection, from which the associated camera motion parameters can be extracted linearly. The essential matrix has a counterpart for uncalibrated images called the *fundamental matrix* [17]. The essential and fundamental matrices form the basis of most SFM techniques [50], [30], [63], [72], [76], [14], [44], [64], [29], [40]. However, these 8-point algorithms are very sensitive to noise because they only examines a small number of features in two images. Torr and Murray [63], Hartley [19] and Philip [40] described robust methods for computing the essential matrix and fundamental matrix by examining additional feature points. However the essential and fundamental matrices are only defined for two images and other techniques must be used to incorporate additional images; e.g., Kalman filtering (see below).

The *Factorization Method*, developed by Tomasi [59], is also widely used in SFM [46], [5], [42], [9]. The Factorization Method assumes parallel projection, which allows the translation of the camera in each image to be determined directly from the projected centroid of the feature points. This leaves linear rotation equations which are solved using *Singular*

[6] Some invariants, such as the centroid and cross-ratio, have an obvious geometric interpretation. Others, such as the essential matrix, are defined purely algebraically.

value decomposition (SVD). SVD determines the optimal rotation parameters, in terms of affine least-squares, from a complete set of projected feature points. SVD does not enforce an orthonormal rotation matrix, i.e., it only recovers affine motion, therefore the result is normalized by examining additional feature points to obtain metric reconstruction [59]. Unlike most invariant-based techniques, the Factorization Method is scalable in both the number of features and the number of images. However, it is not recursive and it assumes parallel projection, which severely limits camera motion and scene structure (these limitations are described in more detail in Chapter 3). An extension of the Factorization Method to para-perspective projection by Poelman and Kanade [42] relaxed this constraint, but the fundamental limitations of non-perspective projection remain.

SFM techniques that examine a fixed number of images can be further distinguished according to whether they recover *metric*, i.e., Euclidean, structure [72], [64], [28], [29], [71], [74], [66], [41], *affine* structure [59], [34], [42], [46], [60], [51] or *projective* structure [50], [76], [44], [36], [17], [2]. That is, the solution is correct up to a scale factor and Euclidean transformation (i.e., translation and rotation), affine transformation (i.e., translation and non-rigid rotation, or *skew*) or an arbitrary projective transformation. As noted by Oliensis [39], projective reconstruction is equivalent to Euclidean reconstruction up to a non-rigid interpretation of the rotation matrix, therefore the choice depends largely on the intended purpose and whether the intrinsic camera parameters are known. For example, object modelling clearly requires metric structure, whereas for object recognition, i.e., deciding whether two images correspond to the same object, the projective structure is sufficient [50].

For many linear SFM techniques it is possible to derive uniqueness results. For example, Tsai and Huang [64] derived the uniqueness of the 8-point algorithm, and Horn [21], Holt and Netravali [20], Quan [44], Negahdaripour [38], Faugeras and Maybank [14] and Weng *et al.* [71] examined the general uniqueness properties of reconstruction from two and three images. Uniqueness is primarily of interest for mathematical completeness and it is rarely an issue when redundant data is available. However, for some applications uniqueness may be

important; for example, coplanar and collinear points are common in indoor scenes and may result in degenerate solutions. Uniqueness requirements are therefore of most interest during feature selection.

The principle advantage of SFM techniques based on a small number of images is that they often have linear solutions. However, because they examine few features and images the solution is very sensitive to noise in these measurements. This is typically addressed by examining additional feature points and performing least-squares analysis [71], [64], [28], [34], [33], [63], [19], [40], [32]. However, with the exception of the Factorization Method, these techniques are rarely scalable to multiple images which limits the extent to which the solution can be made robust. Some methods exploit invariants of non-perspective projection models and thus are only accurate for specific cases of camera motion and scene structure; e.g., the Factorization Method and its derivatives [59], [46], [5], [42], [9]. In particular, parallel projection, para-perspective projection and weak perspective projection cannot model perspective foreshortening (an important depth cue) and therefore the camera must remain a constant distance from the scene and the scene cannot vary significantly in depth.

2.2 Kalman Filtering

SFM techniques that examine only two images have limited reliability and cannot be scaled to longer image sequences. *Kalman filtering* is often used to combine multiple reconstructions over time to obtain better estimates of the structure and motion parameters [75].[7] For example, Shapiro [48] used a Kalman filter to integrate affine structure computed from image pairs, Azarbayejani and Pentland [2] used an *Extended Kalman Filter* to integrate projective structure and motion recovered from uncalibrated images using the fundamental matrix,[8] Weng *et*

[7] A Kalman filter is a general mathematical technique for modelling a dynamic linear system that provides a least-square estimate of the system's parameters (i.e., the structure and motion) derived from the previously observed measurements (i.e., the images).

al. [72] gave a similar solution for calibrated images using the essential matrix, and McLauchlan and Murray [32] used a *Variable State-Dimension Filter* - a Kalman filter with a modified state vector - to integrate the structure and motion parameters separately, which allowed features to be missing or occluded.

In general, Kalman filtering produces a better estimate of the structure and motion parameters than its component reconstructions. However, the quality of the result still depends on the method used to analyze each pair of images. In particular, Kalman filtering only gives optimal parameter estimates for a *linear system*, and the error distribution of the measurements must be approximately Gaussian with a zero mean (Note: in this context the "measurements" refer to the structure and motion values recovered from each pair of images, not the images themselves). However, SFM methods that examine pairs of images have a non-uniform error distribution due to their sensitivity to noise; i.e., they non-uniformly amplify the original image noise. The ability of Kalman filtering to assimilate these solutions is therefore undermined.

Kalman filtering essentially provides an external mechanism for extending SFM techniques based on image pairs to longer image sequences. In many cases this gives a better estimate of the structure and motion parameters. However, under the conditions in which Kalman filtering is used, the solution cannot be guaranteed to be any more reliable than the underlying technique [39].

2.3 Optimization-based SFM Techniques

Whereas Kalman filtering improves estimated parameter values by using successive measurements, optimization-based SFM techniques determine globally optimal parameter values by minimizing an objective error function with respect to *all* the images and features at once. For

---

[8] The Extended Kalman filter (EKF) approximates non-linear systems by a linear Taylor series expansion.

example, Weng *et al.* [72] obtained an initial estimate of structure and motion from two images using the essential matrix, which was then refined using *Levenberg-Marquardt* (LM) optimization[9] with error function based on noise variance. Szeliski and Kang [55] also used LM to perform non-linear optimization and initially estimated the positions of feature points directly from a single image by assuming the distance from the scene to the camera was known. Coorg and Teller [11] optimized the camera positions relative to known feature positions using least-squares, and Spetsakis [53] used a non-linear "loaded spring" model to minimize the errors in the positions of feature points. The Factorization Method may also be considered an optimization-based technique because SVD is a least-squares method. In fact, least-squares optimization is widely used in SFM techniques based on image pairs, where additional feature points are examined to combat noise [71], [64], [28], [34], [33], [63], [19], [40], [32]. Although examining more feature points gives a somewhat better result, reconstruction from only two images is known to be unreliable in general [53].

Optimization-based SFM techniques primarily differ in their projection model and objective error function. The 'ideal' error function, at least for recovering structure and motion, is the metric distance between the original and recovered camera positions and feature points. However, this is impossible to compute because it requires complete scene information. Any claim of optimality is therefore largely meaningless because it depends entirely on the chosen error function.[10]

Optimization produces the best estimate of structure and motion from the given images with respect to the chosen error function. However, optimization is fundamentally a search algorithm and convergence to a local minimum is possible.[11] The ability to locate the

---

[9] Levenberg-Marquardt is a batch least-squares optimization method [43].

[10] By definition, all optimization methods are 'optimal' with respect to their objective error function, assuming they converge to the global minimum. The relative merit of different error functions is an open question.

[11] This is also true for Kalman filtering, which performs iterative linear least-squares optimization.

global minimum depends on the accuracy of the initial estimate and the shape of the error surface, the latter of which is difficult to characterize because of the large number of parameters involved. Non-linear methods are also slower to converge and typically have more complex error surfaces, thus there is a greater likelihood of converging to local minimum.

Optimization and Kalman filtering can be considered alternatives to obtaining reliable structure and motion from fast, but unreliable, invariant-based methods. Kalman filtering integrates multiple partial reconstructions over time in the hopes of obtaining a better estimate, whereas optimization takes an initial estimate and finds globally optimal parameter values from it. Both methods rely on obtaining a good estimate of the structure and motion parameters by other means; Kalman filtering relies on these estimates continuously, whereas optimization requires a good estimate only once.

2.4 Requirements of a General Purpose SFM Technique

A general purpose SFM technique should reliably and efficiently recover scene structure and camera motion while assuming as little as possible about the camera and scene. Initially, SFM research focussed on the problem of making the inverse projection problem tractable and efficient by exploiting invariants of the projection equations for a small number of images [66], [44], [36], [17], [28], [64], [71], [67], [29], [74], [26], [50], [51]. More recently, the reliability of these methods to noise has been the focus [76], [10], [54], [40], [9], [48], [2], [42], [63], [70], [59]. Nevertheless, SFM has yet to be used extensively in real applications because other practical requirements still remain; in particular, dealing with outliers, occlusion and scalability over multiple images. The requirements of a general purpose SFM technique can be summarized as:

1. Fast - real-time SFM requires efficient algorithms. In many cases obtaining an approximate solution quickly is more important than its precision.

2. Reliable - the solution must be robust to common errors such as noise, outliers and missing features (i.e., occlusion).

3. Accurate - the scene, camera, and camera motion must be accurately modelled.

These requirements provide a practical basis to evaluate SFM techniques. Inverse projection is a hard problem and trade-offs are unavoidable.

Fast methods require linear or efficient non-linear algorithms. For example, SFM techniques based on the essential matrix or the fundamental matrix provide a fast, if unreliable, estimate of structure and motion from only two images [29], [64], [17]. The Factorization Method gives a more reliable solution in some cases by examining multiple images, but only under conditions where the orthographic camera model is valid [59]. Speed and rate of convergence is an important reason why linear optimization methods, such as SVD and Kalman filtering [72], [48], [2], [32] have been preferred over non-linear optimization methods [55], [53].

In real-time applications video images arrive continuously and are temporally coherent. For maximum efficiency, each image must be processed quickly by updating an internal model; that is, processing is *recursive*. The alternative, called *batch processing*, requires storing all the previous images and recomputing the solution for each new image, which is prohibitively expensive in terms of storage and processing time. SFM techniques based on non-linear optimization [55], [72] and Kalman filtering [48], [2], [72], [32] are recursive and therefore suited to real-time SFM. The Factorization Method [59], [5], [46], [9], on the other hand, is a batch method and is less suitable, although recursive variations have been developed [42], [70].

The unreliability of recovering structure and motion from a small number of features or images is well understood [54], [53], [62]. Noise can be addressed by applying least-square

analysis to a pair of images [76], [71], [40], [63], [28], or using Kalman filtering to integrate estimates from successive image pairs [72], [48], [2], [32], or by globally optimizing the parameters over all the features and images [59], [55], [53], [72], [11]. The problem of outliers, on the other hand, is rarely addressed. In particular, *correspondence errors* resulting from mismatching features between images are not uncommon in practice. These errors have a non-uniform distribution; i.e., they appear as *outliers*. Outliers are problematic because they can easily overwhelm least-squares optimization and render the result meaningless. Most existing SFM techniques do not handle outliers and assume they are (manually) removed during pre-processing. Notable exceptions are Szeliski and Kang [55] who discarded feature points that had a large projected error after LM optimization; McReynolds and Lowe [33], who tested for the presence of non-rigid motion by examining the residual error after LM optimization;[12] Boult and Brown [5], who examined the residual error after SVD to segment the scene into its rigid components; and Thompson *et al.* [58], Zhang *et al.* [76] and Shapiro [48], who performed outlier detection and rejection from a pair of images. In general, the issue of outliers is poorly addressed in SFM, if at all. Nevertheless, this problem must be addressed before SFM can be used in automated environments.

The problem of outliers is closely related to that of occlusion because both require that features can be missing in specific images. One of the problems of linear SFM techniques is that they invert linear systems which requires having full matrices. That is to say, all the features must be present in all the images because there is no mechanism for representing "no information." Occlusion is more readily handled by non-linear optimization methods because they examine the error in individual feature points. For example, Szeliski and Kang [55] and McReynolds and Lowe [33] assigned occluded feature points a zero weight. McLauchlan and Murray [32] proposed a novel adaptation of the (linear) Kalman filter by introducing a dynamic state variable that allowed features to be added and removed. Most SFM techniques, however, do not support occlusion and therefore must examine only those feature points that

[12] Non-rigid motion violates the rigidity constraint and is another source of outliers.

are present in all the images [50], [26], [74], [2], [71], [44], [17], [72], [51], [46], [9], [70], [42], [60]. Their solutions are therefore sub-optimal because not all the available image data is used. This approach to occlusion also does not scale to long image sequences because few, if any features are present in all the images. A different approach was taken by the Factorization Method which instead filled in occluded feature points by estimating their 3D positions from a subset of images, and then re-projected these points back to the images where they were occluded, a process called *hallucination* [59]. However, any notion of optimality is lost because no distinction can be made between original and derived data and there is a danger of introducing artificial outliers. Despite being an inherent property of real scenes, the issue of occlusion has received surprisingly little attention in SFM.

A final requirement of a general-purpose SFM technique is that the scene, camera, and camera motion are accurately modelled. In practice, this requires perspective projection. Parallel projection, para-perspective and weak perspective projection are frequently used in SFM because they offer a simpler projection model that enables linear solutions [59], [46], [42], [5], [48], [10], [70], [66]. However, these projection models are only valid approximations to optical projection when the camera is far from the scene and maintains a constant distance, and when the relative depth of the scene is small. SFM techniques based on non-perspective projection are therefore not general-purpose because they limit camera motion and scene structure. The *pin-hole camera* model, i.e., perspective projection, is a more accurate model of optical projection but does not model some non-linear lens distortions. More accurate camera models do exist, such as Tsai's camera model [65], but they are rarely used in SFM.

2.5 A Two-Stage Approach: Projected Error Refinement

This thesis proposes a new optimization-based SFM technique called *Projected Error Refinement*. This is a two-stage approach where the structure and motion parameters are first estimated from a subset of the features and images; e.g., using a SFM method defined for a pair of images. This initial estimate is then refined using non-linear optimization with the objective

error function measuring the angular projection error between the solution and the original images. Occlusion is handled naturally because only the visible feature points contribute error terms - missing features are ignored. Outlier detection and rejection is also well-defined in terms of the projector model; after optimization, if the residual error of a feature point is statistically inconsistent with the noise model then it is considered to be an outlier and is rejected. This is similar to Szeliski and Kang [55], who also optimized structure and motion based on a projected image error measurement and handled occlusion and outliers in a similar manner. However, Szeliski and Kang assumed the distance between the camera and scene was known and constant. This simplified the projection equations and allowed the initial positions of feature points to be estimated directly from one image. Another similar two-stage technique was proposed by Weng *et al.* [72], however they ignored the issue of outliers and occlusion. Spetsakis [53] described a non-linear optimization method that supported occlusion but did not consider the problem of outliers.

Projected Error Refinement is non-linear and an efficient *parallel iterative refinement* algorithm is used that alternately refines the structure and motion parameters. After initially estimating the camera's motion and scene structure, the camera's pose (i.e., its position and rotation) in each image is fixed and the positions of the feature points are optimized in parallel. Next, the feature points are fixed and the cameras' pose in each image is optimized, again in parallel. Parallel iterative refinement converges rapidly towards the global minimum. Refinement allows the solution to be computed to an arbitrary precision or it can be terminated prematurely; the latter being important for real-time applications where the precision of the solution can be made subject to the processing resources available. A similar approach to alternately refining the structure and motion parameters was recently proposed by Poelman and Kanade [42] for approximating perspective projection by iterative para-perspective reconstruction. However, they did not support occlusion or address the issue of outliers.

To summarize, Projected Error Refinement is an efficient non-linear optimization technique that is scalable in the number of features and images, supports missing features (i.e.,

occlusion) and detects outliers in a well-defined manner. A perspective camera model is used and no additional assumptions about camera motion or scene structure are made other than the rigidity constraint. Further, any initial estimate of camera motion and scene structure can be used. In this sense, Projected Error Refinement gives at least as accurate a reconstruction as other SFM techniques.

Chapter 3
Optical Projection Models

Determining the 3D positions of points in a scene is difficult because the projected 2D image only records a point's relative direction from the camera, not its distance. Recovering distance, depth, or equivalent thereof, is known as the *inverse projection problem*. This chapter describes optical projection as performed by the camera and various mathematical models of it used in SFM, including the *projector model* on which Projected Error Refinement is based. Inverse projection is then described in terms of the projector model to identify the parameters of the problem and its constraints.

3.1 Optical Projection

A camera is a direction sensitive sensor. Light enters the camera through a lens which redirects the light rays to particular sensor elements depending on the direction from which they arrived, as shown in Figure 3.1. More precisely, the lens *focuses* the light from different points in the scene onto the image sensor. If a point in the scene lies on the *plane of focus* then its light rays are focussed onto a single sensor element.[13] If not, then the light from the point gets distributed onto a region in the image, i.e., it is burred. The projection of a point onto the image therefore depends on its relative direction from the camera and, to a lesser extent, its distance. The projected image is nevertheless purely two-dimensional and contains no explicit depth information.[14]

[13] For this discussion the granularity or resolution of the sensor is not important and can equally be a CCD array or photographic film.

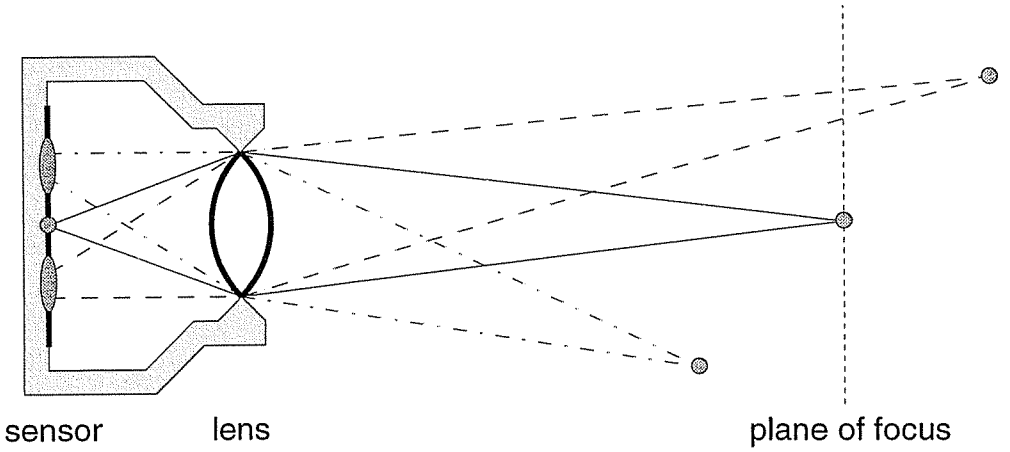


Figure 3.1: A camera lens projects a point in the scene on the *plane of focus* to a point in the image. Points not on the plane of focus are progressively blurred in the image according to their distance from the plane.

3.2 The Pin-Hole Camera and Perspective Projection

The optical properties of a camera lens are very difficult to precisely model. For this reason, the classical model of optical projection used in computer vision is the *pin-hole camera*. The pin-hole camera consists of an enclosed box with a infinitely small hole on one side, called the *center of projection* or *optical center*, through which light enters before striking the image sensor on the opposite side, as shown in Figure 3.2. The projected image is inverted because the image plane is located behind the optical center. The image is typically modelled as being in front of the optical center to obtain an upright image, without loss of generality. If the optical center of the camera is the world origin and the image plane is normal to the positive z-axis, then the transformation mapping a feature point $\mathbf{p} = \begin{bmatrix} x & y & z \end{bmatrix}$ in the scene to a point $\mathbf{p}' = \begin{bmatrix} x' & y' \end{bmatrix}$ in the image is called *perspective projection* and is defined as

$$\mathbf{p}' = \frac{f}{z}\begin{bmatrix} x \\ y \end{bmatrix}, \qquad 3.1$$

[14] Image blur can be used to recover depth from a single image if certain assumptions are made. These techniques are called *depth from focus* [37].

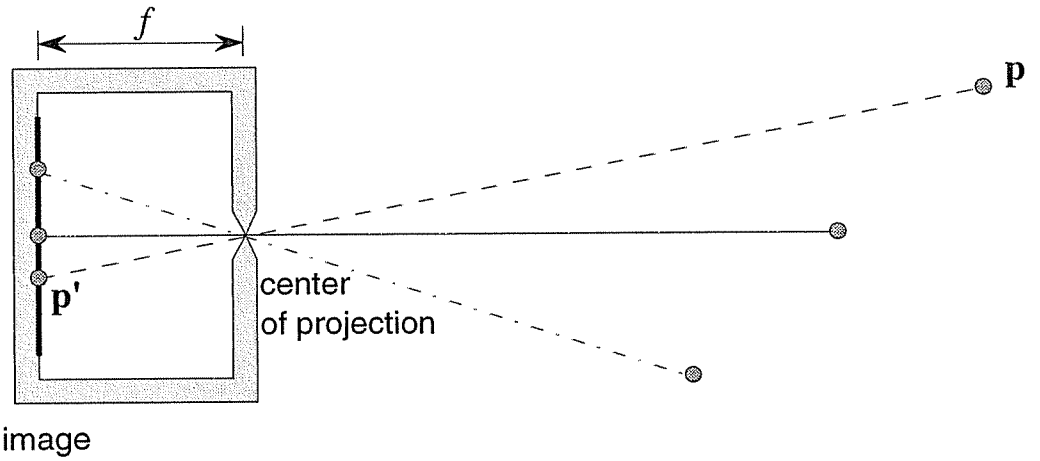


Figure 3.2: The *pin-hole camera* approximates a camera lens by perspective projection. All light from the scene passes through the *center of projection* onto an inverted image. The image is perfectly focussed because only one light ray enters the camera from every direction. Note: the field-of-view of the pin-hole camera cannot exceed 180°.

---

where f is the *focal length* of the camera, i.e., the distance from the optical center to the image plane. Perspective projection does not model lens focus or image blur, but it does model the most important property of optical projection - that objects appear smaller the further they are from the camera.

3.3 Non-Perspective Projection

Although perspective projection only approximates optical projection, it is often still too complex for efficient linear SFM methods. In particular, the projected position $\mathbf{p}'$ of a feature point depends the camera's pose and three different scene parameters, namely the 3D position of $\mathbf{p}$. This results in non-linear solutions and for this reason simpler camera models are frequently used. The simplest, called *parallel projection*, is defined as

$$\mathbf{p}' = \begin{bmatrix} x \\ y \end{bmatrix} \qquad 3.2$$

and is illustrated in Figure 3.3(a). Under parallel projection, the projected position $\mathbf{p}'$ is independent of the distance of $\mathbf{p}$ from the camera. Parallel projection is only a good approximation to optical projection when the camera is located far from the scene, such that the incident light

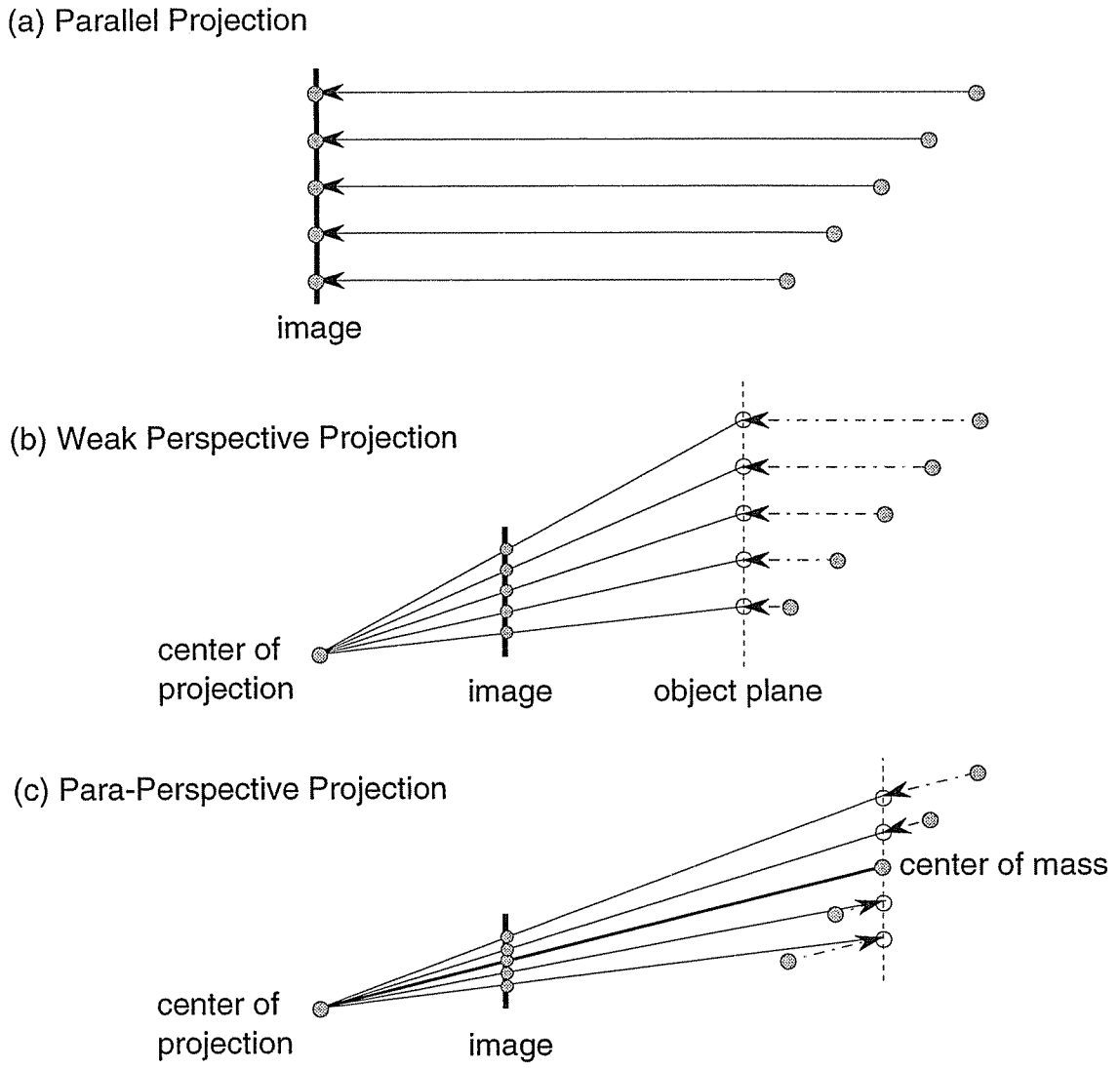


Figure 3.3: (a) *Parallel projection*, (b) *weak perspective projection* and (c) *para-perspective projection* are approximations to perspective projection that simplify the inverse projection problem by reducing the number of parameters, at the expense of constraining camera motion and scene structure.

---

rays are almost parallel. In spite of this, parallel projection is used by several SFM techniques, including the Factorization Method [59], [60], [46], [5], [66], [58], [68], [67]. As a result, these methods require the camera to be located far from the object of interest and to maintain a constant distance so that the projected size of the scene remains unchanged. The depth of the scene along the optical axis must also be small to avoid perspective foreshortening effects in the projected shape of the object.

Weak perspective projection is used to address the constraint of parallel projection on the camera maintaining a constant distance from the scene [10], [70], [48], [55][15]. Weak perspective is defined as

$$\mathbf{p}' = \frac{f}{z_0}\begin{bmatrix} x \\ y \end{bmatrix}, \tag{3.3}$$

where z_0 is the distance from the optical center to the *object plane* - a plane parallel to the image that passes through or nearby the object, as shown in Figure 3.3(b). Weak perspective first projects the scene onto the object plane by parallel projection, followed by a perspective projection onto the image, which simplifies to a scale change. Effectively, weak perspective replaces the depth z of each feature point by an average depth z_0 for each object, thus reducing the number of parameters. However, this requires segmenting features into distinct objects which is difficult without *a priori* depth information, so typically the scene is assumed to contain a single object at a distance z_0 from the camera [10], [70], [48], [55]. As before, the scene cannot vary significantly in depth to prevent foreshortening.

Para-perspective projection addresses the problem of weak perspective in that motion parallel to the image plane does not affect the projected appearance of an object. This has the effect of causing increasing image distortion the further off the optical axis an object is located [42]. Para-perspective projection is defined as

$$\mathbf{p}' = \frac{f}{z_0}\left(\begin{bmatrix} x \\ y \end{bmatrix} + \frac{z_0 - z}{z_0}\begin{bmatrix} x_0 \\ y_0 \end{bmatrix}\right), \tag{3.4}$$

where $\mathbf{c} = \begin{bmatrix} x_0 & y_0 & z_0 \end{bmatrix}$ is the object's *center off mass*, as shown in Figure 3.3(c). Para-perspec-

[15] Szeliski and Kang [55] claim to perform perspective projection but it is more accurate to describe their camera model as 'scaled' perspective. In their model, the scene is first projected onto the object plane, using perspective projection rather than parallel projection, followed by a scale change.

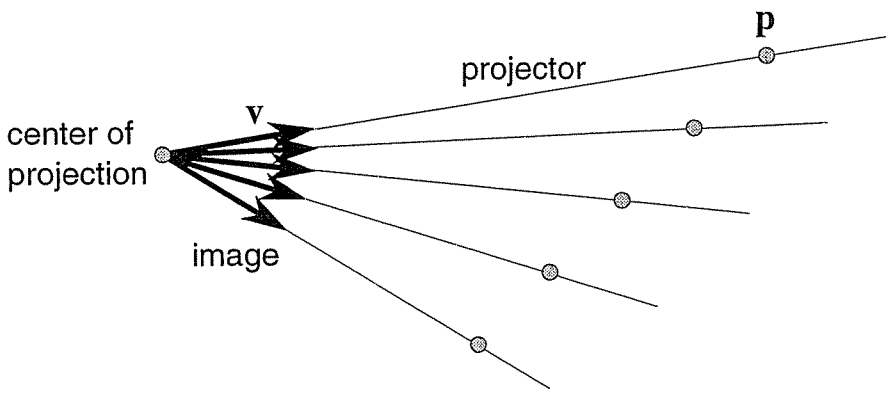


Figure 3.4: In the *projector model* a feature point **p** is projected to a vector **v** in the image, which defines the line through the scene on which **p** must lie. There is no other information on the whereabouts of **p**.

---

tive first projects the scene onto the object plane by parallel projection in the direction of **c**, i.e., parallel to the vector from the optical center to the center of mass. This somewhat compensates for the change in the appearance of an object as it moves off the optical axis. As with weak perspective, segmentation is avoided by assuming the scene contains a single object [42]. As with all the other non-perspective projection models, the scene cannot vary significantly in depth.

SFM is a hard problem and compromises must be made. Non-perspective projection models are used because they facilitate fast linear solutions that are mathematically elegant. However, with the exponentially increasing processing power of computers, the fundamental limitations of techniques based on non-perspective projection is becoming more important than speed, especially with the development of efficient non-linear optimization techniques [55], [72]. Except for limited applications, accurate scene reconstruction requires perspective projection.

3.4 The Projector Model

The projector model is a generalization of the pin-hole camera. A point **p** in the scene is projected to the image by a straight line, called a *projector*, as shown in Figure 3.4. The only information contained in the image about the position of **p** is its direction from the optical cen-

ter, which is described by the unit direction vector **v**. The image of a scene containing multiple features is the set of vectors describing the relative directions of the visible feature points. These direction vectors depend only on the location of the optical center and not the rotation of the camera around it, and therefore any suitable internal coordinate frame can used to describe the vectors.

If the optical center of the camera is the world origin, the direction vector **v** of the projector for a feature point $\mathbf{p} = \begin{bmatrix} x & y & z \end{bmatrix}$ in the scene is

$$\mathbf{v} = \frac{\mathbf{p}}{|\mathbf{p}|}. \tag{3.5}$$

In practice, images are acquired from a camera modelled by perspective projection. The conversion from planar image coordinates $\mathbf{p}' = \begin{bmatrix} x' & y' \end{bmatrix}$ to the corresponding direction vector is

$$\mathbf{v} = \frac{\begin{bmatrix} (x' - x_P) & (y' - y_P) & f \end{bmatrix}}{\left| \begin{bmatrix} (x' - x_P) & (y' - y_P) & f \end{bmatrix} \right|} \tag{3.6}$$

where f is the focal length and $\begin{bmatrix} x_P & y_P \end{bmatrix}$ is the *principle point* of the image; i.e., the point of intersection of the optical axis and image plane.[16]

The projector-based image model is equivalent to *spherical projection* or *central projection*, where the scene is projected onto a unit sphere surrounding the optical center, rather than a planar image as in the case of parallel or perspective projection. Although planar images are used extensively in computer vision, spherical projection or its equivalent has long been used in SFM. For example, Koenderink and Van Doorn [23] used spherical projection to derive invariant properties of the *optical flow field* obtained from a series of closely spaced

[16] Eqn. 3.6 assumes the aspect ratio is 1 and the image's axes are perpendicular. Regardless, the conversion from Cartesian image coordinates to projectors is well-defined in terms of the intrinsic camera parameters.

images to recover the type, magnitude and direction of the local surfaces on a rigid object. Spherical projection was used by Yen and Huang [74] to recover camera translation and rotation from two or three images. More recently, Maybank [31] used spherical image coordinates to extract the *essential matrix* relating the projected positions of feature points in two images, from which the rotation and translation transformation between the two image coordinate frames are recovered.

Projectors and spherical image coordinates are similar to *homogeneous coordinates* used extensively in SFM techniques based on projective geometry [76], [50], [17], [2]. The homogeneous coordinates of a projected feature point are $\mathbf{p}'' = \begin{bmatrix} x'' & y'' & f \end{bmatrix}$, where $f \neq 0$ and $\mathbf{p}' = \frac{1}{f}\begin{bmatrix} x'' \\ y'' \end{bmatrix}$ is the corresponding planar image coordinate. Maybank [31] gave a formulation of the inverse projection problem in terms of *epipolar geometry* and homogeneous coordinates (i.e., in projective space) that is equivalent to his aforementioned essential matrix formulation in Euclidean space, and noted that the projective formulation permits reconstruction up to a *collineation* in the case where the intrinsic camera parameters are unknown and yields a simpler analysis of the uniqueness of the result.[17] Faugeras [13] used homogeneous image coordinates extensively, where they are called *normalized image coordinates*, for intrinsic camera calibration and to recover camera rotation and translation using the essential matrix derived from eight feature points projected to two images. A non-linear solution requiring only five feature points was also given. Homogeneous coordinates differ from projectors and spherical image coordinates in that they are a 3D representation of 2D planar coordinates, hence images represented using homogeneous coordinates strictly cannot have a field of view exceeding 180°, although this is rarely a limitation in practice.

Projectors are a general model of projection that can be defined for any geometric pro-

---

[17] A *collineation* is an arbitrary linear transformation in projective space.

jection transformation in any dimension, although only 2D and 3D perspective projection are examined in this thesis.[18]

3.5 The Inverse Projection Problem

Inverse projection describes the problem of recovering the position of a feature point from its projected position in an image. In the projector model, the projection of a feature point **p** is represented by a unit vector **v**, the direction vector of a line originating at the optical center on which **p** must lie. The position of **p** along this line cannot be determined from the image alone. For a single image of a scene there exists an infinite number of consistent features positions. That is to say, inverse projection from a single image is fundamentally ill-posed.

If the scene is projected to two or more images with different optical centers, and the same features are projected each time (i.e., the rigidity constraint), then the inverse projection problem changes dramatically. For example, if the positions of the optical centers and orientations of the images are known, then the feature points can be directly recovered by finding the intersection of any two projectors, called *stereoscopic triangulation*, as shown in Figure 3.5. The unknown parameters of the inverse projection problem are therefore the positions of the optical centers of the cameras and the orientations of the projected images around them, i.e., the extrinsic camera parameters. The only constraints on these parameters are that all the projectors defined for each feature point are concurrent, i.e., all the projectors must intersect at a single point. As described in Chapter 4, the concurrency of multiple lines can be described mathematically in terms of the camera parameters. The more features and images there are the more constrained these parameters become. In some cases the constraints may be insufficient to uniquely determine the solution. For example, 2D inverse perspective projection from two images is under-constrained regardless of the number of feature points because two images

[18] To be precise, 2D and 3D *central projection* are examined because images can have a field of view exceeding 180°. For practical purposes, however, central projection is equivalent to perspective projection.

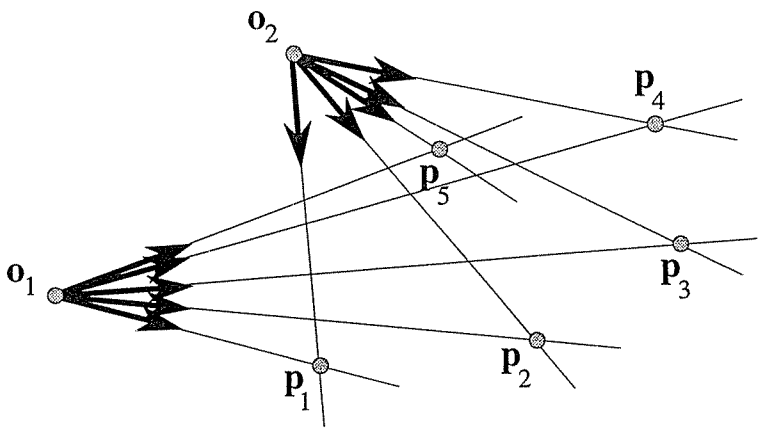


Figure 3.5: Inverse projection from multiple images involves determining the positions of the optical centers $\mathbf{o}_j$ for each camera and the orientation of the image vectors around them. If the extrinsic camera parameters are known then the positions of the feature points $\mathbf{p}_i$ can be determined by triangulation.

---

can be placed in virtually any configuration and their respective projectors will intersect. However, for 3D inverse perspective projection, two images of five points has at most three solutions in the general case, as proved by Thompson [57] and Ullman [66], and shown in Chapter 4 using the projector model.

SFM has traditionally formulated the inverse projection problem as determining the rotation and translation transformations between two images by exploiting algebraic invariants of the associated projection equations; for example, expressed by the essential matrix. The projector model, on the other hand, gives a more intuitive geometric interpretation of the constraints and parameters of the inverse projection problem based on the fact that the projectors defined by each image must intersect. This is a natural way of describing SFM and it has practical benefits in terms of scalability and dealing with outliers and missing features.

Chapter 4
Reconstruction from Minimal Image Data

Given multiple images of a rigid scene, the constraints on the locations of the cameras and orientations of the images are that the projectors defined for each feature point in the scene must intersect. The concurrence of these lines can be described in terms of the extrinsic camera parameters. Each image adds a set of associated camera parameters to the inverse projection problem, and each feature point adds constraints on these parameters. Given the minimal number of images and features necessary for reconstruction, the resulting system of simultaneous equations can be solved exactly to recover the camera parameters, and hence scene structure.

This chapter derives a *minimal data solution* to the inverse projection problem for 2D and 3D perspective projection. In the projector model, only the *relative* directions of the feature points in each image are important, therefore the projected images are first normalized to a standard image coordinate frame. The parametric equation of a projector is then described, which defines the line through the scene on which the feature point must lie, whose parameters are the position of the camera and the rotation of the normalized image. Each feature point defines a projector for every image in which it is visible. The necessary concurrence of these lines defines a system of equations which are solved using Newton's method to recover the extrinsic camera parameters. An example is given of recovering structure and motion from a minimal set of synthetic images. The chapter concludes by describing the effect of projection noise on the minimal data solution and why additional features and images must be examined to obtain a reliable reconstruction.

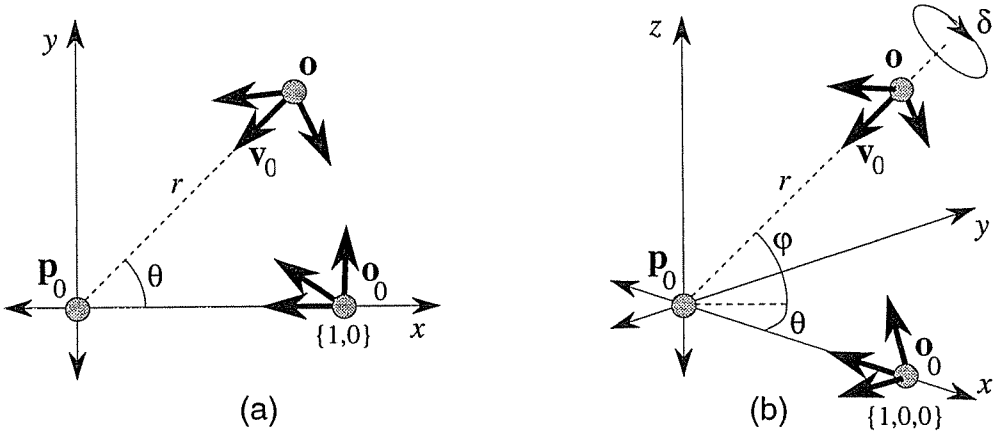


Figure 4.1: The world coordinate system for (a) 2D and (b) 3D perspective projection defines feature point $\mathbf{p}_0$ as the origin and the optical center $\mathbf{o}_0$ as unit distance along the positive x-axis. The remaining optical centers $\mathbf{o}$ are described by *polar coordinates*. Note: the image vector $\mathbf{v}_0$ always points to the origin. In $\Re^2$ the rotation of the image is determined entirely by the position of $\mathbf{o}$. In $\Re^3$ the rotation of the image about the axis of $\mathbf{v}_0$ is given by δ.

4.1 Extrinsic Camera Parameters

The unknown parameters of the inverse projection problem are the extrinsic parameters of each camera, i.e., the location of the optical center and the rotation of the image around it. The original scene's coordinate system is unknown and therefore a world coordinate system is imposed where an arbitrary feature point $\mathbf{p}_0$ is defined as the origin and the optical center $\mathbf{o}_0$ of an arbitrary image as being unit distance along the positive x-axis, as shown in Figure 4.1.

The optical centers of the cameras are described in polar coordinates in $\Re^2$ as

$$\mathbf{o} = r\begin{bmatrix}\cos\theta\\ \sin\theta\end{bmatrix}, \tag{4.1}$$

and in $\Re^3$ as

$$\mathbf{o} = r\begin{bmatrix}\cos\theta\cos\varphi\\ \sin\theta\cos\varphi\\ \sin\varphi\end{bmatrix}. \tag{4.2}$$

In $\Re^2$, the rotation of the image is determined entirely by the position of the optical center because the projector $\mathbf{v}_0$ must always points to $\mathbf{p}_0$ at the origin. In $\Re^3$, the only free parameter is the rotation of the image about the axis of $\mathbf{v}_0$, which is parameterized by δ. Therefore, for the general case of J images, the extrinsic camera parameters for 2D perspective projection are $\mathbf{o}_j = \{r_j, \theta_j\}$, where $\mathbf{o}_0 = \{1, 0\}$, for $j = 0 \ldots (J-1)$. For 3D perspective projection the parameters are $\mathbf{o}_j = \{r_j, \theta_j, \varphi_j\}$ and δ_j, where $\mathbf{o}_0 = \{1, 0, 0\}$ and $\delta_0 = 0$, for $j = 0 \ldots (J-1)$.[18]

Imposing a world coordinate system is necessary but as a consequence the original scene structure and camera motion can only be recovered up to a scale factor and a rigid translation and rotation. This is true of all SFM techniques that recover metric structure.

4.2 Image Normalization

As described in Chapter 3, only the positions of the optical centers contribute useful information for inverse projection, not the rotations of the images. Therefore, the vectors in each image are first normalized. In particular, all vectors are made unit length and then rotated to align $\mathbf{v}_0$, the unit vector for feature point $\mathbf{p}_0$, along the negative x-axis. For 2D perspective projection this rotation is given by

$$\mathbf{v}' = \mathbf{v} \cdot \begin{bmatrix} -u_0 & v_0 \\ -v_0 & -u_0 \end{bmatrix}, \qquad 4.3$$

where $\mathbf{v}'$ is the normalized vector, $\mathbf{v}$ is the unit vector and $\mathbf{v}_0 = \begin{bmatrix} u_0 & v_0 \end{bmatrix}$. For 3D perspective projection the rotation is given by

[18] These parameters only apply to the minimal data solution where zero noise is assumed and projectors precisely intersect. The refined solution, described in Chapter 5, employs the full 3 or 6 degrees of freedom associated with the camera location and rotation, for 2D and 3D perspective projection respectively.

$$\mathbf{v}' = \mathbf{v} \cdot G(-\mathbf{v}_0, -\mathbf{v}_1, -\mathbf{v}_0 \cdot -\mathbf{v}_1), \tag{4.4}$$

where G is a Gram-Schmidt orthonormal basis constructed from $\mathbf{v}_0$ and $\mathbf{v}_1$, after which $\mathbf{v}'_1$ is parallel to the x-y plane.

4.3 Parametric Equation of Projectors

A feature point **p** projected to an image with optical center **o** defines a projector, with direction vector **v**, passing through the scene on which **p** is known to lie (see Figure 3.4). The parametric equation of this line is

$$\mathbf{p} = \mathbf{o} + \Omega\mathbf{v}. \tag{4.5}$$

The polar coordinates of **o** are given by Eq. 4.1 and Eq. 4.2. As the (unknown) position of **o** changes, the vectors in the image must rotate to maintain $\mathbf{v}_0$ pointing to $\mathbf{p}_0$ at the origin (see Figure 4.1). Therefore, the direction vector **v** of a projector depends on the normalized image vector $\mathbf{v}'$ suitably rotated according to the position of the optical center and, in the 3D case, the parameter δ describing the rotation of the image. Substituting Eq. 4.1 and the matrix for a 2D rotation of θ into Eq. 4.5 gives the parametric equation of a projector in $\Re^2$:

$$\mathbf{p} = r\begin{bmatrix}\cos\theta \\ \sin\theta\end{bmatrix} + \Omega\mathbf{v}' \cdot \begin{bmatrix}\cos\theta & \sin\theta \\ -\sin\theta & \cos\theta\end{bmatrix}. \tag{4.6}$$

Likewise, substituting Eq. 4.2 and the matrix for a 3D rotation defined by $\{\theta,\varphi,\delta\}$ into Eq. 4.5 gives the parametric equation of a projector in $\Re^3$:

$$\mathbf{p} = r\begin{bmatrix}\cos\theta\cos\varphi \\ \sin\theta\cos\varphi \\ \sin\varphi\end{bmatrix} + \Omega\mathbf{v}' \cdot \begin{bmatrix}1 & 0 & 0 \\ 0 & \cos\delta & \sin\delta \\ 0 & -\sin\delta & \cos\delta\end{bmatrix} \cdot \begin{bmatrix}\cos\varphi & 0 & \sin\varphi \\ 0 & 1 & 0 \\ -\sin\varphi & 0 & \cos\varphi\end{bmatrix} \cdot \begin{bmatrix}\cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1\end{bmatrix}. \tag{4.7}$$

4.4 Concurrency Constraint

A feature point defines a projector for each image in which it is visible. By definition, these

lines are concurrent. The concurrency of multiple lines constrains the parameters of the lines involved; in this case, the extrinsic camera parameters. The concurrency of lines in $\Re^2$ and $\Re^3$ are defined differently and therefore the two cases are examined separately.

4.4.1 Concurrent Lines in $\Re^2$

The concurrence of two lines in $\Re^2$ provides no useful constraint except that the direction vectors of the two lines cannot be parallel. Given only two projected images of a scene, the positions of the two cameras is under-constrained regardless of the number of features. 2D inverse perspective projection from two images is therefore ill-posed and at least three images are required.

A feature point projected to three images defines three concurrent lines:

$$\begin{aligned} \mathbf{p} &= \mathbf{o}_1 + \Omega_1 \mathbf{v}_1 \\ \mathbf{p} &= \mathbf{o}_2 + \Omega_2 \mathbf{v}_2 \\ \mathbf{p} &= \mathbf{o}_3 + \Omega_3 \mathbf{v}_3 \end{aligned} \tag{4.8}$$

giving

$$\begin{bmatrix} x_1 \\ y_1 \end{bmatrix} + \Omega_1 \begin{bmatrix} u_1 \\ v_1 \end{bmatrix} = \begin{bmatrix} x_2 \\ y_2 \end{bmatrix} + \Omega_2 \begin{bmatrix} u_2 \\ v_2 \end{bmatrix} = \begin{bmatrix} x_3 \\ y_3 \end{bmatrix} + \Omega_3 \begin{bmatrix} u_3 \\ v_3 \end{bmatrix}, \tag{4.9}$$

where $\mathbf{o} = \begin{bmatrix} x & y \end{bmatrix}$ is the optical center and $\mathbf{v} = \begin{bmatrix} u & v \end{bmatrix}$ is the normalized unit direction vector obtained from Eq. 4.6. Eliminating Ω_j and simplifying gives

$$\begin{vmatrix} \begin{vmatrix} x_1 & u_1 \\ y_1 & v_1 \end{vmatrix} & \begin{vmatrix} x_2 & u_2 \\ y_2 & v_2 \end{vmatrix} & \begin{vmatrix} x_3 & u_3 \\ y_3 & v_3 \end{vmatrix} \\ u_1 & u_2 & u_3 \\ v_1 & v_2 & v_3 \end{vmatrix} = 0. \tag{4.10}$$

Eq. 4.10 describes the constraints on the parameters of three concurrent lines in $\Re^2$.

4.4.2 Concurrent Lines in $\Re^3$

Unlike $\Re^2$, the concurrence of two lines in $\Re^3$ is special and imposes useful constraints on the parameters of the lines. Two lines are concurrent if

$$\begin{aligned} \mathbf{p} &= \mathbf{o}_1 + \Omega_1 \mathbf{v}_1 \\ \mathbf{p} &= \mathbf{o}_2 + \Omega_2 \mathbf{v}_2 \end{aligned} \tag{4.11}$$

giving

$$\begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix} + \Omega_1 \begin{bmatrix} u_1 \\ v_1 \\ w_1 \end{bmatrix} = \begin{bmatrix} x_2 \\ y_2 \\ z_2 \end{bmatrix} + \Omega_2 \begin{bmatrix} u_2 \\ v_2 \\ w_2 \end{bmatrix}, \tag{4.12}$$

where $\mathbf{o} = \begin{bmatrix} x & y & z \end{bmatrix}$ is the optical center and $\mathbf{v} = \begin{bmatrix} u & v & w \end{bmatrix}$ is the normalized unit direction vector obtained from Eq. 4.7. Eliminating Ω_j and simplifying gives

$$\begin{vmatrix} (x_1 - x_2) & u_1 & u_2 \\ (y_1 - y_2) & v_1 & v_2 \\ (z_1 - w_1) & w_1 & w_2 \end{vmatrix} = 0. \tag{4.13}$$

Eq. 4.13 describes the constraints on the parameters of two concurrent lines in $\Re^3$.

4.5 Minimal Data Solution

The minimal data solution examines the minimum number of feature points and images necessary to solve the inverse projection problem. Each image adds an independent set of parameters describing the pose of the camera. Each feature point in the scene imposes a non-linear concurrency constraint on these parameters, given by Eq. 4.10 and Eq. 4.13. The inverse projection problem is well-conditioned if the number of equations equals the number of parameters. If the system of equations are independent they can be solved numerically to determine the extrinsic camera parameters. If the equations are not independent then the problem

remains under-constrained; the uniqueness of SFM reconstruction and its degenerate conditions are described in [66], [64], [14], [38], [20]. In most non-degenerate cases there is a very small number of solutions, possibly one. The minimal data solutions for inverse projection in $\Re^2$ and $\Re^3$ are somewhat different and therefore they are examined separately.

4.5.1 Minimal Data Solutions for 2D Inverse Perspective Projection

Each image in $\Re^2$ has two extrinsic camera parameters $\mathbf{o}_j = \{r_j, \theta_j\}$, where $\mathbf{o}_0 = \{1, 0\}$. For J images there are therefore $2(J-1)$ parameters. For I feature points, each point defines one constraint for every three images (Eq. 4.10), i.e., $J-2$ equations per feature point, with the exception of $\mathbf{p}_0$ whose projectors are implicitly concurrent in the definition of the world coordinate system. The resulting system of equations is well-conditioned if

$$\begin{aligned} (J-2)\,(I-1) &\geq 2\,(J-1) \\ IJ - 3J - 2I + 4 &\geq 0 \end{aligned} \qquad 4.14$$

which has two minimal solutions, {J=3, I=5} and {J=4, I=4}. That is, there are *two* minimal data solutions to the 2D inverse perspective projection problem - one minimal in the number of images and the other minimal in the number of features; they are: three images of five points and four images of four points.

4.5.2 Minimal Data Solution for 3D Inverse Perspective Projection

Each image in $\Re^3$ has four extrinsic camera parameters: the location of the optical center $\mathbf{o}_j = \{r_j, \theta_j, \varphi_j\}$ and the rotation of the image δ_j, where $\mathbf{o}_0 = \{1, 0, 0\}$ and $\delta_j = 0$. For J images there are therefore $4(J-1)$ parameters. For I feature points, each point defines one constraint for a pair of images (Eq. 4.13), with the exception of $\mathbf{p}_0$. The resulting system of equations is well-conditioned if

$$\begin{aligned} (I-1) &\geq 4\,(J-1) \\ I - 4J + 3 &\geq 0 \end{aligned} \qquad 4.15$$

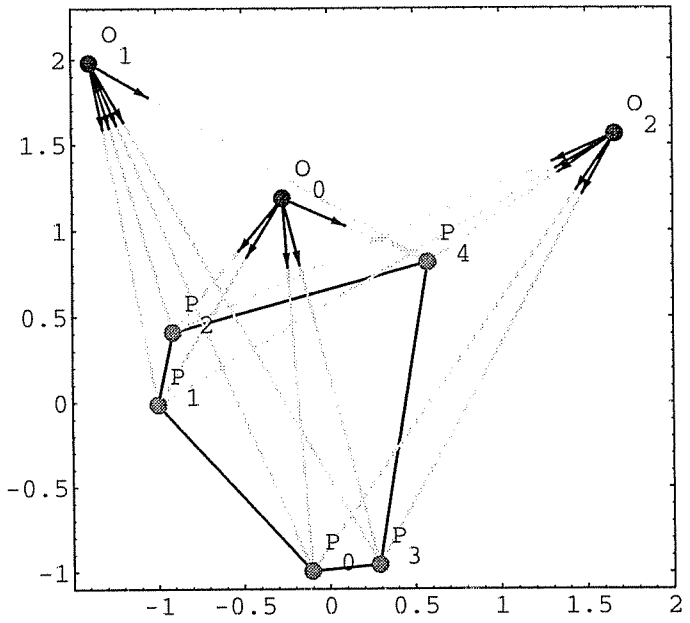


Figure 4.2: A synthetic 2D scene containing five feature points $\mathbf{p}_i$ projected to three images with optical centers $\mathbf{o}_j$ by ideal 2D perspective projection.

---

which has one minimal data solution $\{J=2, I=5\}$. The minimal data solution to the 3D inverse perspective projection problem is therefore two images of five points. This result is consistent with Thompson [57] and Ullman [66].

4.6 Example

Figure 4.2 shows a synthetic scene in $\Re^2$ containing five feature points $\mathbf{p}_0$, $\mathbf{p}_1$, $\mathbf{p}_2$, $\mathbf{p}_3$, $\mathbf{p}_4$ projected to three images with optical centers $\mathbf{o}_0$, $\mathbf{o}_1$, $\mathbf{o}_2$ by ideal perspective projection (i.e., zero image noise). First, the images are normalized and a world coordinate system defined with $\mathbf{p}_0$ as the origin and $\mathbf{o}_0$ at $\{1,0\}$. The four unknown parameters are the polar coordinates of the remaining optical centers $\mathbf{o}_1 = \{r_1, \theta_1\}$ and $\mathbf{o}_2 = \{r_2, \theta_2\}$. Each feature point defines one equation describing the concurrence of its three projectors, with the exception of $\mathbf{p}_0$. This gives a system of four non-linear equations in four unknowns, which are solved numerically using Newton's method to give the polar coordinates of the optical centers:[19]

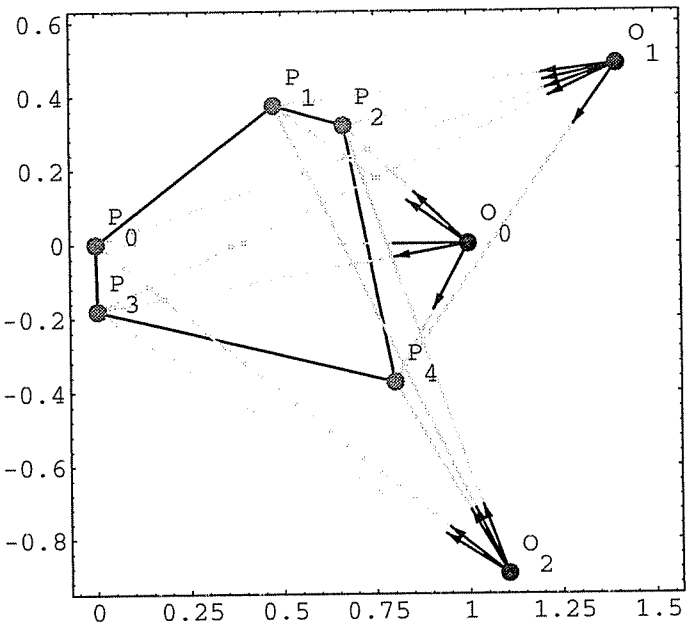


Figure 4.3: The reconstructed scene from the minimal images projected in Figure 4.2. The relative positions of the feature points and images is identical to the original scene, up to a scale factor, translation and rotation.

$$\mathbf{o}_0 = \{1, 0°\}$$
$$\mathbf{o}_1 = \{1.4826, 19.1237°\}$$
$$\mathbf{o}_2 = \{1.4212, -39.0534°\}$$

The positions of the feature points are then found by triangulation from a pair of images:

$$\mathbf{p}_0 = \{0, 0\}$$
$$\mathbf{p}_1 = \{0.4779, 0.3747\}$$
$$\mathbf{p}_2 = \{0.6676, 0.3204\}$$
$$\mathbf{p}_3 = \{0.0035, -0.1806\}$$
$$\mathbf{p}_4 = \{0.8013, -0.3738\}$$

The reconstructed scene is shown in Figure 4.3, which is identical to the original scene shown in Figure 4.2 up to a scale change, rotation and translation. That is, the metric structure of the

---

[19] This calculation took 0.075 seconds of CPU time on a 133MHz Pentium™ PC running Mathematica™ 2.2 using the FindRoot operator (i.e., Newton's Method).

original scene and camera motion is recovered.

To summarize, given three images of a scene containing five feature points, and with no prior knowledge of the whereabouts of either the features or camera centers, the positions of the feature points and the optical centers of the cameras can be recovered based on the constraint that the projectors defined by each image intersect. A 2D example was shown here for clarity - an equivalent example can be constructed for 3D inverse perspective projection using two images of five points.

4.7 Effect of Projection Noise

In practice, it is impossible to precisely measure the direction of feature points in the scene due to imperfections of the camera model, digitization, etc. To illustrate the effect of noise on the minimal data solution, the images projected in Figure 4.2 are corrupted by zero-mean Gaussian noise with an angular error of $\sigma = 0.15°$ in each projector, corresponding to approximately $\sigma = 3$ pixels for a camera with a 35mm lens and a 640 pixel wide image. The scene reconstructed from the noisy images is shown in Figure 4.4. The result shows that the error in the solution resulting from a nominal amount of image noise is significant. This behavior is typical of SFM techniques that examine a small number of feature points and images [54], [62]. Inverse projection is inherently unstable and small errors in the projected positions of feature points are greatly magnified in the solution. In order to obtain a reliable estimate of scene structure and camera motion, redundant data in the form of additional feature points and images must be examined.

4.8 Summary

The minimal data solutions to 2D and 3D inverse perspective projection described in this chapter are not intended to be viable alternatives to existing SFM techniques. Indeed, there exist better linear solutions that require only a few more feature points, such as those based around the essential matrix [29], [64]. Rather, the minimal data solution gives an intuitive geo-

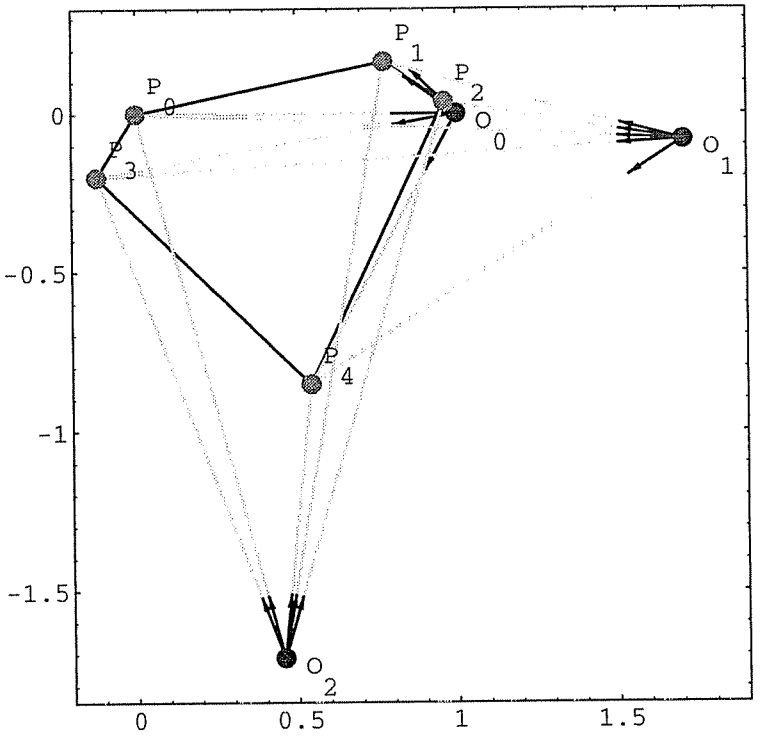


Figure 4.4: The reconstructed scene from the images projected in Figure 4.2 after each projector is corrupted by 0.15° noise (approximately 3 pixels). The recovered positions of the feature points is skewed and there is a significant error in the position of $\mathbf{o}_0$ compared with the original scene.

---

metric interpretation of the parameters and constraints of the inverse projection problem based on the concurrency constraint, and how these determine the minimal number of feature points and images necessary for reconstruction. The minimal data solution also introduces the projector-based model of the inverse projection problem that is used by Projected Error Refinement to deal with noise, scalability, occlusion and outlier detection, as described in the next chapter.

Chapter 5
Projected Error Refinement

Inverse projection from few features or images is very sensitive to noise and most SFM techniques examine additional image data and apply least-squares analysis [61], [63], [59], [70], [9], [42], [40], [76], [55], [46], [11] or Kalman filtering [34], [72], [48], [2] to obtain more reliable reconstruction. As described in Chapter 2, many SFM methods are only defined for two or three images and the reliability of these methods is limited because additional images cannot be included. Kalman filtering attempts to overcome this limitation by integrating multiple estimates over time, however Kalman filtering cannot guarantee a more reliable solution than its component reconstructions due to their non-uniform error distribution. Optimization-based SFM techniques, on the other hand, are scalable in both the number of features and the number of images, and examine *all* the available projected information. The principle disadvantage of optimization-based methods are that many are non-linear.

This chapter describes *Projected Error Refinement*, an optimization-based SFM technique based on the projector model. In the minimal data solution, described in Chapter 4, the projectors precisely intersect because perfect perspective projection is assumed. In reality, projectors do not precisely intersect because of image noise. The optimal positions of the images and feature points are therefore defined as those which minimize the mean-squared *angular projection error* of each projector. The resulting objective error function is non-linear and an efficient *parallel iterative refinement* algorithm is used that separates the refinement of structure and motion. In particular, the structure parameters (i.e., the positions of the feature points) and motion parameters (i.e., the positions of the cameras' optical centers and the rota-

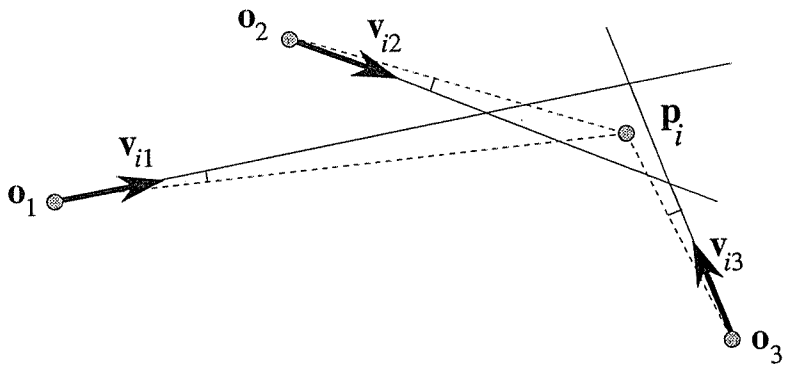


Figure 5.1: The optimal position of a feature point $\mathbf{p}_i$ minimizes the angle between the observed image projector $\mathbf{v}_{ij}$ and the projector $(\mathbf{p}_i - \mathbf{o}_j)$ in the solution. Note: the further a feature is from the camera the greater the error in its distance from the projector.

tions of the images) are alternately adjusted while the other is kept constant. Each iteration monotonically decreases the mean projection error. Projected Error Refinement requires an initial estimate of the structure and motion parameters which is obtained using the minimal data solution described in Chapter 4. An example is given at the end of this chapter of recovering the structure and motion parameters from synthetic 2D images containing noise, showing how examining additional feature points and images and minimizing the projection error gives a more reliable reconstruction of the scene.

5.1 Concurrency of Projectors

The projected position of feature point in an image is imprecise due to an imperfect camera model, quantization, feature detector resolution, and other sources of 'noise'. The optimal position of a feature point with respect to its non-concurrent projectors is defined to be the point which minimizes the observed error in the images; in particular, the mean-squared *angular projection error* between the feature's observed direction vectors and its estimated direction vectors from the recovered camera locations and the feature's position, as shown in Figure 5.1. Other optimization-based SFM techniques that minimize the observed (planar) image error include Poelman and Kanade [42], Weng *et al.* [72] and Szeliski and Kang [55]. Others, such as Coorg and Teller [11] and Spetsakis [53], define the optimal position of a fea-

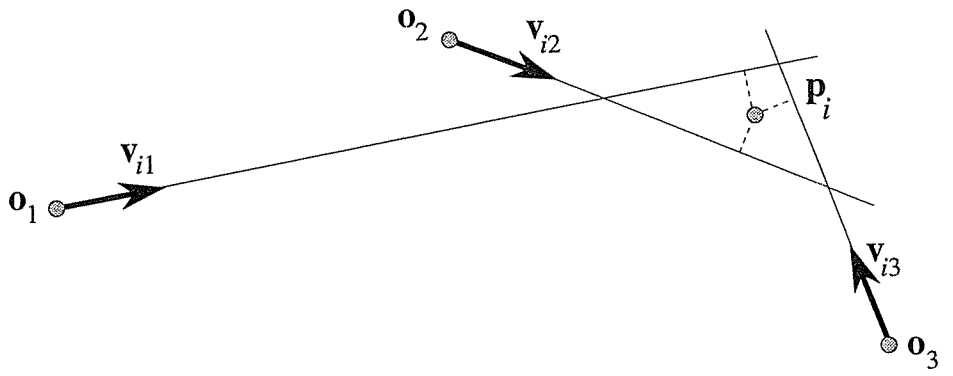


Figure 5.2: An alternative definition of the optimal feature position is the point which minimizes the distance between the projectors. However, features far from the camera are expected to have a greater error in their position, therefore this definition favors feature points far away.

ture to be the point that is minimum distance from all its projectors, as shown in Figure 5.2. However, this is a less accurate model because projection noise is primarily caused by limitations of the camera sensor should not be modelled differently for different features. In other words, features located far from the camera *should* have a greater error in their recovered position. Minimizing the distance between non-concurrent projectors, rather than the projected error, effectively increases the measured precision of features far away, which is unwarranted.

5.2 Angular Projection Error

The angular projection error of a projector is the angle ε_{ij} between the observed projector $\mathbf{v}_{ij}$ in the image and the recovered projector $(\mathbf{p}_i - \mathbf{o}_j)$ in the solution, shown in Figure 5.3 and given by

$$\tan\varepsilon_{ij} = \frac{\left|\text{prog}_{\mathbf{v}_{ij}^{\perp}}(\mathbf{p}_i - \mathbf{o}_j)\right|}{\left|\text{prog}_{\mathbf{v}_{ij}}(\mathbf{p}_i - \mathbf{o}_j)\right|}. \qquad 5.1$$

For $-\frac{\pi}{2} \le \varepsilon_{ij} \le \frac{\pi}{2}$, ε_{ij} is minimized when $\tan\varepsilon_{ij}$ is minimized,[20] therefore the *squared* projection error is

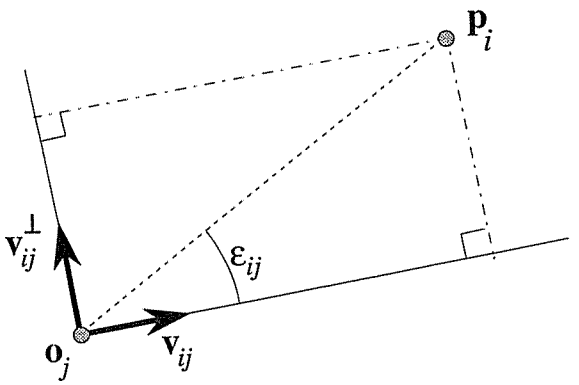


Figure 5.3: The angular projection error ε_{ij} is given by the ratio of the projection of the vector $(\mathbf{p}_i - \mathbf{o}_j)$ onto the image's projector $\mathbf{v}_{ij}$, and the projection of $(\mathbf{p}_i - \mathbf{o}_j)$ onto the orthogonal vector $\mathbf{v}_{ij}^{\perp}$.

$$\tan^2\varepsilon_{ij} = \frac{\left|\text{prog}_{\mathbf{v}_{ij}^{\perp}}(\mathbf{p}_i - \mathbf{o}_j)\right|^2}{\left|\text{prog}_{\mathbf{v}_{ij}}(\mathbf{p}_i - \mathbf{o}_j)\right|^2} = \frac{\left|\dfrac{\mathbf{v}_{ij}^{\perp} \cdot (\mathbf{p}_i - \mathbf{o}_j)}{\mathbf{v}_{ij}^{\perp} \cdot \mathbf{v}_{ij}^{\perp}} \mathbf{v}_{ij}^{\perp}\right|^2}{\left|\dfrac{\mathbf{v}_{ij} \cdot (\mathbf{p}_i - \mathbf{o}_j)}{\mathbf{v}_{ij} \cdot \mathbf{v}_{ij}} \mathbf{v}_{ij}\right|^2}. \qquad 5.2$$

$\mathbf{v}_{ij}$ and $\mathbf{v}_{ij}^{\perp}$ are unit vectors so Eq. 5.2 simplifies to

$$\tan^2\varepsilon_{ij} = \frac{(\mathbf{v}_{ij}^{\perp} \cdot (\mathbf{p}_i - \mathbf{o}_j))^2}{(\mathbf{v}_{ij} \cdot (\mathbf{p}_i - \mathbf{o}_j))^2}. \qquad 5.3$$

In $\Re^2$ the orthogonal vector of $\mathbf{v}_{ij} = \begin{bmatrix} u_{ij} & v_{ij} \end{bmatrix}$ is $\mathbf{v}_{ij}^{\perp} = \begin{bmatrix} -v_{ij} & u_{ij} \end{bmatrix}$, giving

$$\tan^2\varepsilon_{ij} = \frac{(\begin{bmatrix} -v_{ij} & u_{ij} \end{bmatrix} \cdot (\mathbf{p}_i - \mathbf{o}_j))^2}{(\begin{bmatrix} u_{ij} & v_{ij} \end{bmatrix} \cdot (\mathbf{p}_i - \mathbf{o}_j))^2}. \qquad 5.4$$

---

[20] The Taylor Series expansion of $\tan\varepsilon$ is $\varepsilon + \frac{\varepsilon^3}{3} + \frac{\varepsilon^5}{5} + \ldots$ For small angles $\varepsilon < \frac{\pi}{180}$ (i.e., $\varepsilon < 1°$) the difference between ε and $\tan\varepsilon$ is $O\left(\frac{\pi^3}{180^3}\right)$, or approximately 0.01%.

Determining the orthogonal vector $\mathbf{v}_{ij}^{\perp}$ in $\Re^3$ is more difficult. However, the length of $\mathbf{v}_{ij}^{\perp}$ is given by the cross product

$$\begin{aligned} \text{prog}_{\mathbf{v}_{ij}^{\perp}}(\mathbf{p}_i - \mathbf{o}_j) &= \mathbf{v}_{ij} \times (\mathbf{p}_i - \mathbf{o}_j) \\ \left|\text{prog}_{\mathbf{v}_{ij}^{\perp}}(\mathbf{p}_i - \mathbf{o}_j)\right|^2 &= (\mathbf{v}_{ij} \times (\mathbf{p}_i - \mathbf{o}_j)) \cdot (\mathbf{v}_{ij} \times (\mathbf{p}_i - \mathbf{o}_j)) \end{aligned} \qquad 5.5$$

giving

$$\tan^2 \varepsilon_{ij} = \frac{(\left[u_{ij}\ v_{ij}\ w_{ij}\right] \times (\mathbf{p}_i - \mathbf{o}_j)) \cdot (\left[u_{ij}\ v_{ij}\ w_{ij}\right] \times (\mathbf{p}_i - \mathbf{o}_j))}{(\left[u_{ij}\ v_{ij}\ w_{ij}\right] \cdot (\mathbf{p}_i - \mathbf{o}_j))^2} \qquad 5.6$$

where $\mathbf{v}_{ij} = \left[u_{ij}\ v_{ij}\ w_{ij}\right]$.

Eq. 5.4 and Eq. 5.6 give the angular projection error $\tan^2 \varepsilon_{ij}$ between $\mathbf{v}_{ij}$, the observed direction vector in the image, and $(\mathbf{p}_i - \mathbf{o}_j)$, the recovered projector in the solution, for 2D and 3D perspective projection respectively. In both cases, this error is defined in terms of the parameters of $\mathbf{o}_j$ and $\mathbf{p}_i$.

Because the *tangent* of ε_{ij} is minimized, angular errors greater than 90° are indistinguishable from those less than 90°. As a result, if the estimated position of a feature point $\mathbf{p}_i$ is such that the angle between the projector $\mathbf{v}_{ij}$ in the image and $(\mathbf{p}_i - \mathbf{o}_j)$ in the solution exceeds 90°, then optimization will converge to a solution where the feature point is located *behind* the camera instead of in front of it. This is a case where Projected Error Refinement fails to converge to the correct solution. Projected Error Refinement models projectors as lines rather than rays and therefore it does not enforce the constraint that feature points must be located in front of the camera.

5.3 Extrinsic Camera Parameters

In the minimal data solution, the unknown parameters are the positions of the cameras' optical centers and, for 3D perspective projection, an additional parameter describing the rotation of the image. However, if the images contain noise then the projectors no longer intersect. Therefore, the vector $\mathbf{v}_0$, i.e., the projector for feature point $\mathbf{p}_0$, may not precisely pass through the origin, as it does in the minimal data solution (see Figure 4.1). That is to say, the rotation of the image becomes uncoupled from the location of the optical center. Each image in 2D perspective projection therefore has three parameters: the position of the camera's optical center $\mathbf{o}_j = \{r_j, \theta_j\}$ and a parameter μ_j describing the rotation of the image. Similarly, each image in 3D perspective projection now has six extrinsic parameters: the position of the camera's optical center $\mathbf{o}_j = \{r_j, \theta_j, \varphi_j\}$ and three parameters $\{\mu_j, \lambda_j, \delta_j\}$ describing the rotation of the image.

5.4 Structure Parameters

In the minimal data solution, the positions of the feature points are computed directly from the recovered images by triangulation. This is not possible when the projectors do not precisely intersect, where the position of each feature is now the point which minimizes the angular projection error of its projectors. This position is determined by the placement of the projectors and hence can be described using the same parameters, i.e., the extrinsic camera parameters. However, in Projected Error Refinement the positions of the feature points are treated as *independent* parameters. Although this may seem to introduce additional parameters that are unnecessary, it allows for the structure and motion parameters to be refined *separately* and *in parallel*, as described in the next section. This is significantly more efficient than optimizing all the extrinsic camera parameters simultaneously. In Projected Error Refinement, the parameters of the inverse projection problem are therefore the positions of the cameras' optical centers and the rotations of the images, *and* the positions of the feature points in the scene.

5.5 Parallel Iterative Refinement

Projected Error Refinement performs non-linear optimization and determines the parameter values by minimizing an objective error function; in this case, the mean-squared angular projection error of the projectors. For a set of I feature points projected to J images, the sum-squared angular projection error is given by

$$\sum_{i=0}^{I-1}\sum_{j=0}^{J-1}\omega_{ij}\tan^2\varepsilon_{ij}, \qquad 5.7$$

where $\tan^2\varepsilon_{ij}$ is the projected error of the feature point $\mathbf{p}_i$ in the image with optical center $\mathbf{o}_j$, and ω_{ij} is the *weight* or confidence of this projector.[21] The error in each projector depends only on the position of $\mathbf{o}_j$, the rotation of the image, and the estimated position of the feature point $\mathbf{p}_i$ (see Eq. 5.4 and Eq. 5.6). The error contributed by each feature point can therefore be computed independently of the other features. Similarly, the error contributed by each image can be computed independently of the other images. In other words, Eq. 5.7 may be rewritten as

$$\sum_{i=0}^{I-1}\left(\sum_{j=0}^{J-1}(\omega_{ij}\tan^2\varepsilon_{ij})\right), \qquad 5.8$$

representing the sum-squared error contributed by each feature point, or

$$\sum_{j=0}^{J-1}\left(\sum_{i=0}^{I-1}(\omega_{ij}\tan^2\varepsilon_{ij})\right), \qquad 5.9$$

representing the sum-squared error contributed by each image.

Parallel iterative refinement exploits this equivalence to reduce the number of parame-

---

[21] All weights are either zero or 1, depending on whether the feature is occluded or visible in the image. Chapter 10 describes a proposed use of variable weights for analyzing long image sequences.

ters considered during optimization. During the first iteration, the parameters of all the feature points are optimized *in parallel* whilst leaving the extrinsic camera parameters constant. That is, the cameras are fixed in space and each feature point is adjusted to find its optimal position. In the next iteration, the parameters of all the cameras are optimized *in parallel* whist leaving the feature positions constant. That is, the feature points are fixed and each camera is optimized. Because Eq. 5.7, Eq. 5.8 and Eq. 5.9 are equivalent, the global projected error monotonically decreases.

Parallel iterative refinement is significantly faster than attempting to simultaneously optimize all the parameters. For example, the general case of I features and J images under 3D perspective projection involves $6J$ extrinsic camera parameters and a very large complex non-linear objective error function. Separating the refinement of structure and motion reduces the number of parameters that are considered at any one time and involves much simpler error functions. In particular, optimizing the feature positions involves only three parameters per feature, and optimizing the camera positions involves only six parameters per image. Most importantly, all the features and images are optimized in parallel.

Parallel iterative refinement scales well because additional features and images are refined in parallel and do not significantly increase the complexity of the error functions. Iterative refinement also allows the solution to be refined to an arbitrary precision or to be terminated at any time to obtain the best current estimate of the structure and motion parameters. This is important for real-time SFM applications where the time available to process each image is limited. Non-iterative methods cannot provide meaningful intermediate results and must execute to completion and have a fixed precision [59], [46], [48], [50]. Parallel iterative refinement also allows new images to be added at any time, by adding their associated projectors and resuming refinement with the additional data. Non-recursive methods unsuitable for real-time applications because they must recompute the solution from scratch for every new image [59], [46], [70], [50], [33], [9], [17], [55].

The parallel iterative refinement algorithm described here was developed independently but is similar to an iterative technique recently described by Poelman and Kanade [42]. However, in their technique it was used to approximate perspective projection by iterative para-perspective projection and they do not handle outliers or missing features. The case of 2D parallel iterative refinement is also similar to a 2D algorithm proposed by Taylor *et al.* [56].

5.6 Initial Estimate

Projected Error Refinement requires an initial estimate of the structure and motion parameters, which is provided by the minimal data solution described in Chapter 4. First, a subset of the feature points and camera centers are reconstructed from a minimal set of images. The recovered images are then used to estimate the positions of the remaining feature points by triangulation, as described in Appendix A. The positions of the remaining cameras' optical centers are then estimated from the recovered feature points, called the *beacon problem* in 2D [24] or the *location determination problem* in 3D [28], [15], as described in Appendix B. Briefly, if a set of known 'beacons' (i.e., the feature points) are observed from an unknown location, then the location of the observer can be determined from the relative observed direction of the beacons; i.e., the positions of the cameras' optical centers can be determined from the relative direction of the recovered feature points.

The minimal data solution was chosen to provide the initial estimate to show that SFM can be solved entirely based on the projector model. In fact, Projected Error Refinement can use an estimate of the structure and motion parameters obtained from any existing SFM technique, such as the Factorization Method or one based on the essential matrix. Although the initial estimate will often be poor because if it is computed from a small set of features and images, Projected Error Refinement usually converges to the global minimum and convergence to local minima has not be a problem in practice.

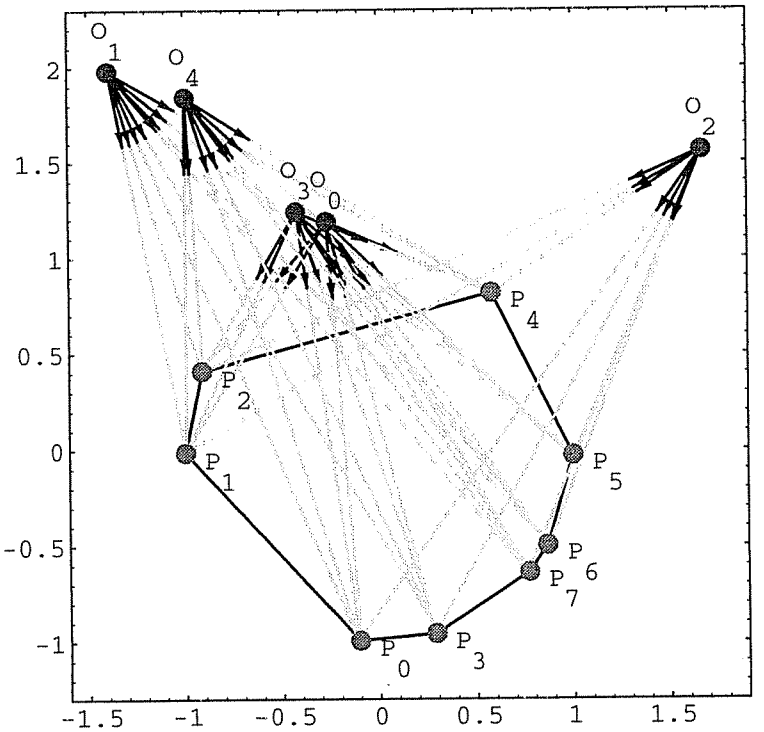


Figure 5.4: A synthetic scene containing eight feature points $\mathbf{p}_i$ projected to five images with optical centers $\mathbf{o}_j$ by 2D perspective projection. The projectors (arrows) are corrupted by Gaussian noise of $\sigma = 1.5°$ (approximately 3 pixels).

5.7 Example

Figure 5.4 shows a synthetic scene in $\Re^2$ containing eight feature points $\mathbf{p}_i, i = 0, \ldots, 7$ projected to five images with optical centers $\mathbf{o}_j, j = 0, \ldots, 4$. Gaussian noise is applied to all the images, where $\sigma = 0.15°$, or approximately $\sigma = 3$ pixels. A random subset of five features and three images is extracted, in this case $\{\mathbf{o}_0, \mathbf{o}_1, \mathbf{o}_4\}$ and $\{\mathbf{p}_1, \mathbf{p}_3, \mathbf{p}_5, \mathbf{p}_6, \mathbf{p}_7\}$. The positions of these feature points and optical centers are estimated using the minimal data solution. Next, the positions of the remaining feature points are determined by triangulation from a pair of the recovered images (see Appendix A) and the locations of the remaining two images are determined from the initial five recovered feature points (see Appendix B). The resulting initial estimate is shown in Figure 5.5 and has an mean angular projection error of 4° and a maximum error of 18°, in the projector from $\mathbf{o}_0$ to $\mathbf{p}_4$. As shown, there is a noticeable error in the

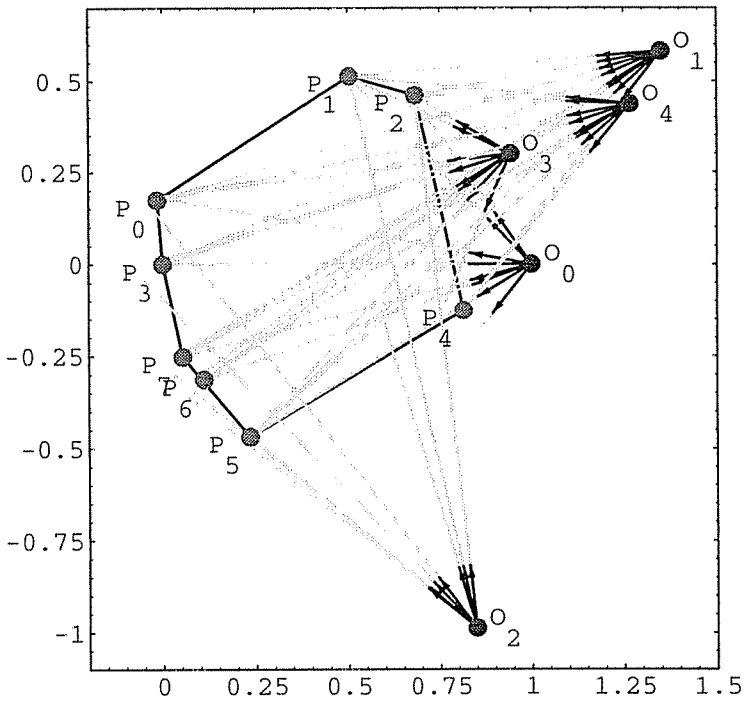


Figure 5.5: The initial estimate of the feature points and optical centers recovered from a minimal subset of the images projected in Figure 5.4. This solution is poor due to image noise.

recovered positions of the feature points, compared to the original scene shown in Figure 5.4, and a significant error in the recovered location of $\mathbf{o}_3$.

The initial estimate of the cameras' optical centers, image orientations and feature points are refined using Projected Error Refinement to minimize the angular projection error in all the images. Refinement was terminated when the difference in the mean projection error between two successive solutions was less than 2%; in this example, 17 iterations. The final refined solution is shown in Figure 5.6 and has an mean angular projection error 0.1° and a maximum error of 0.24°, in the projector from $\mathbf{o}_1$ to $\mathbf{p}_5$.

The recovered scene structure and camera motion is only accurate up to a scale factor and a translation and rotation because the original scene's coordinate system is unknown. Quantitatively comparing the solution to the original scene is therefore difficult because it requires determining the optimal scale, rotation and translation to map the solution's coordi-

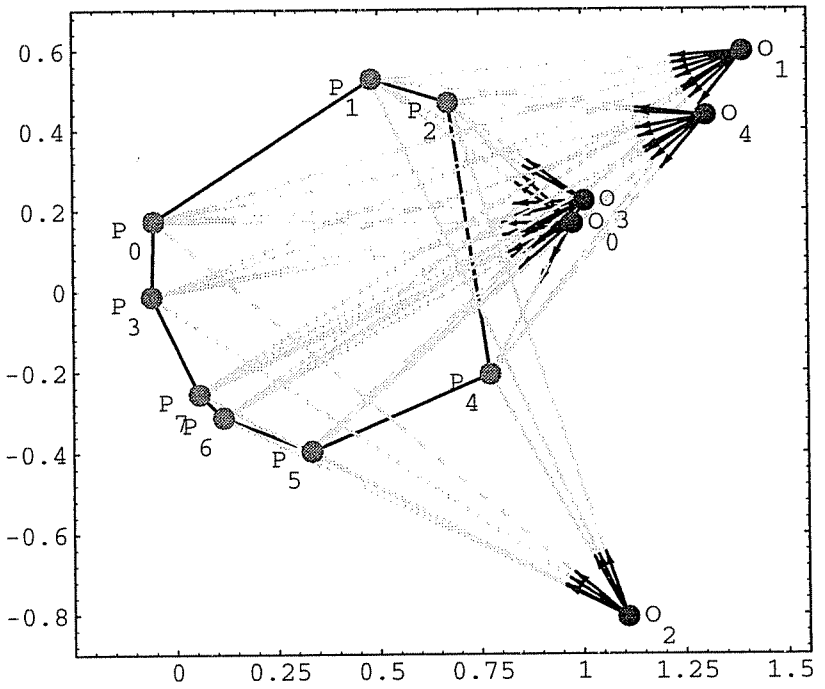


Figure 5.6: The refined solution after 17 iterations. This solution gives a better reconstruction of the original scene, shown in Figure 5.4, than the initial estimate, shown in Figure 5.5.

nate system to the original scene. This is called *data fitting* or *shape registration* and is a non-trivial optimization problem [1], [22]. In order to simplify measuring the error in the solution, the feature points in the original synthetic scene are selected around a unit circle for 2D perspective projection, or a unit sphere for 3D perspective projection. The feature points in the solution should therefore also be located on a unit circle or sphere, making the transformation mapping the solution to the original scene easier to determine, which is described in detail in Appendix C. After the solution has been transformed to the original scene's coordinate system, the mean distance between the original and recovered feature points gives the *structure error* of the solution, and the mean distance between the original and recovered positions of the camera's optical centers gives the *motion error*, as shown in Figure 5.7. In this example, the structure error of the refined solution is 0.03 and the motion error is 0.08. Both distances are measured relative the circle with radius 1.[22] Figure 5.8 shows a trace of the refinement from the initial estimate to the final refined solution, showing how iteratively minimizing the angular projection error results in the solution converging towards the ideal feature and cam-

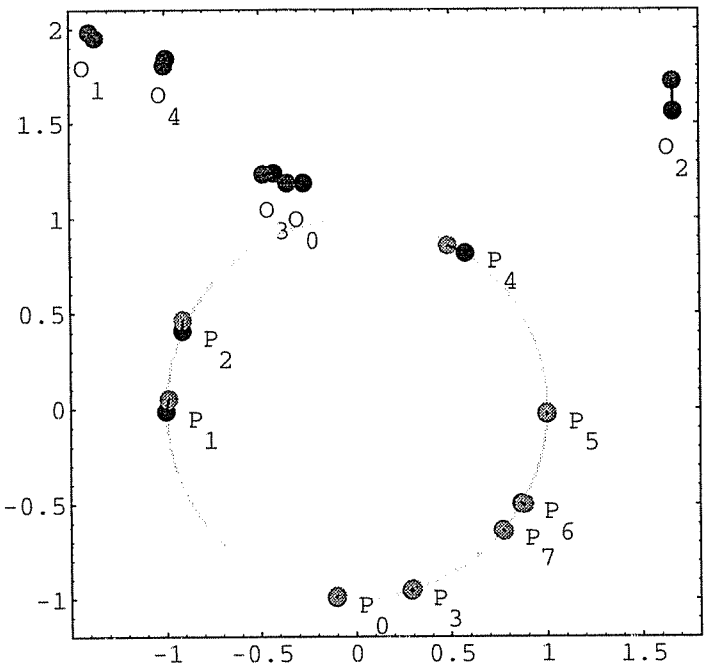


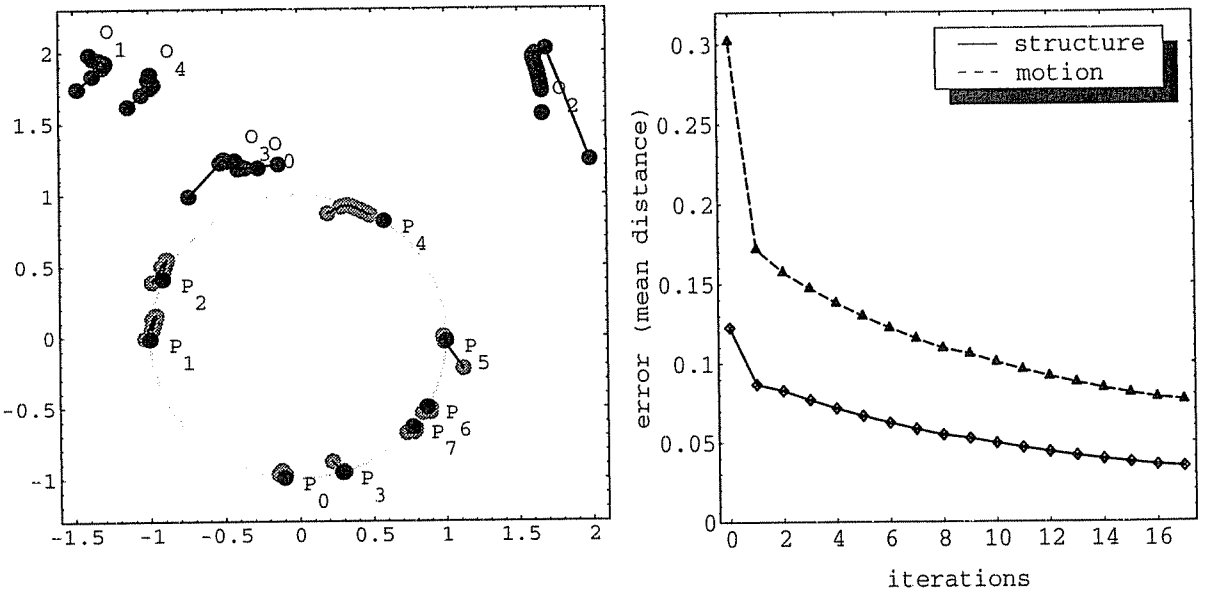
Figure 5.7: The refined solution is transformed to the original scene's coordinate system to measure the reconstruction error (Note: the original feature points lie on a unit circle centered at the origin). The mean distance between the original features **p** (black) and their recovered positions (gray) gives the *structure error*, and the mean distance between the original camera centers **o** (black) and their recovered positions (gray) gives the *motion error.*

era positions.

5.8 Summary

Projected Error Refinement extends the minimal data solution described in Chapter 4 to examine additional feature points and images. Projectors do not precisely intersect due to image noise, therefore the optimal positions of the feature points and images are those which minimize the mean-squared angular projection error of the projectors. Projected Error Refinement consists of three steps:

[22] The motion error is typically greater than the structure error because the positions of the cameras are not considered when computing the transformation between the solution the original scene, not because camera motion is less reliably recovered than scene structure.



(a) (b)

Figure 5.8: (a) Trace of refinement (gray) from the initial estimate to the final solution, compared with the original feature and camera positions (black). (b) Plot of the structure and motion error vs. refinement iterations.

---

1. Estimate the positions of a subset of feature points and images using a suitable method, such as the minimal data solution.

2. Extend the initial estimate to include the remaining feature points and images.

3. Iteratively refine the structure and motion parameters by alternately optimizing the positions of the feature points and camera poses to minimize the angular projection error of each projector.

Projected Error Refinement reliably recovers the metric structure of the original scene in the presence of image noise. It is scalable to an arbitrary number of features and images and does not make additional assumptions about scene structure and camera motion other than the rigidity constraint. As with all optimization methods, the ability to converge to the global min-

imum depends on the accuracy of the initial estimate. In practice, convergence as not been a problem; it is not known whether this is related to optimizing the structure and motion parameters separately.

Optimization-based SFM techniques give the most accurate scene reconstruction because they examine all the feature points and images and thus are the least sensitive to noise [39]. Projected Error Refinement is similar to some existing SFM techniques. For example, Projected Error Refinement uses an objective error function based on the projected image error, similar to Szeliski and Kang [55]; it uses an efficient parallel iterative refinement algorithm, similar to Poelman and Kanade [42]; it is recursive, like Weinshall and Tomasi [70]; and is scalable, like Tomasi [59]. Projected Error Refinement is unique in that it combines all these properties into a single SFM technique using an intuitive geometric model of inverse projection based on projectors. Further, Projected Error Refinement supports occlusion and deals with outliers in a well defined manner - two practical issues that are widely ignored by other SFM methods [76], [63].

Chapter 6
Feature Detection and Tracking

Projected Error Refinement, like most SFM techniques, recovers scene structure and camera motion from *point features*. An important assumption of SFM is that the correspondence problem is solved; that is, image feature points have already been detected and matched to feature points in the other images. Feature detection involves identifying 'interesting' image features, in the hopes that these correspond to important structural features of the scene. Feature detection is a difficult problem, however, because image features can be caused by a wide variety of physical phenomena. Features detected in one image must be matched with the features in the other images to determine their correspondences. Image sequences are typically generated from a moving camera and if the camera motion is smooth then features can be *tracked* between images based on the locality of their projected positions in each image. However, SFM does not assume exact camera motion is known and correspondence errors do occur. Correspondence errors can cause large errors in the identified positions of some feature points, i.e., *outliers*. Outliers can also be introduced when feature detection fails to identify rigid scene features.

This chapter briefly describes the *Kanade-Lucas-Tomasi* (KLT) feature tracker that is used to detect and track feature points in real image sequences. A full description of the method is given in [59], [52], [4]. Feature detection and tracking is one of the principal sources of outliers in SFM, so it is important to understand how outliers are introduced. The detection of outliers is discussed in the next chapter.

6.1 The Kanade-Lucas-Tomasi Feature Tracker

Many feature tracking algorithms define feature detection and feature tracking as separate operations; that is, features are selected based on some local image interest operator that may ignore how features are subsequently tracked. For example, Shapiro [48] defined a corner detector but determined feature correspondences using image correlation. The KLT tracker, on the other hand, specifically defines good features as those that can be tracked well; in other words, feature detection and tracking are closely integrated [59].

6.1.1 Feature Tracking

The KLT tracker matches small image patches between two images. For the purpose of SFM, the center of each patch is considered the feature 'point'. If the camera's frame rate is high then the transformation of a small image patch between two frames can be approximated by a simple 2D translation. A feature patch W is tracked between two images I_n and I_{n+1} by determining the 2D displacement vector $\mathbf{d}$ that minimizes the least-squares difference ε, or *dissimilarity*, of the image intensity gradient over the two regions, defined as

$$\varepsilon = \iint_W [I_n(\mathbf{x}) - I_{n+1}(\mathbf{x}+\mathbf{d})]^2 w(\mathbf{x}) \quad d\mathbf{x}, \tag{6.1}$$

where $I_n(\mathbf{x})$ is the intensity of the image at point $\mathbf{x}$ and $w(\mathbf{x})$ is an optional weighting function. In other words, a feature patch is tracked by finding the best nearby patch in the second image that looks almost identical.

Features degrade as they are tracked over several images. For example, a feature may become partially occluded or rotate to face away from the camera. Feature patches are therefore periodically compared to their appearance in the first image and any that have grown too dissimilar are discarded. Because a large camera motion may have transpired since the feature patch first appeared, the dissimilarity between the current and original feature patch is instead measured by finding the *affine* transformation $\mathbf{A}$ and displacement vector $\mathbf{d}$ that minimizes the

function

$$\xi = \iint_W [I_0(\mathbf{x}) - I_n(\mathbf{A}\mathbf{x} + \mathbf{d})]^2 \quad d\mathbf{x}, \tag{6.2}$$

where ξ is the dissimilarity of W between image I_0 and I_n, and $\mathbf{A}$ is a 2×2 matrix describing the deformation of the feature patch W due to camera motion.

6.1.2 Feature Detection

The KLT tracker selects feature patches based on how well they can be tracked. In particular, Eq. 6.1 is well-conditioned if the two eigenvalues λ_1 and λ_2 of the spatial intensity gradient of the feature patch W are large and approximately the same magnitude [52]. Patches with large eigenvalues correspond to corners, salt-and-pepper textures and other patterns that can be reliably tracked. Features are selected by scanning a patch window across the entire image. Any patch where $\min(\lambda_1, \lambda_2) > \lambda$ is selected as a good feature for tracking, where λ defines the minimum feature strength, and so indirectly determines the number of features selected.

6.2 Correspondence Errors

Correspondence errors occur when two features are matched that are not projections of the same point in the scene. This can occur if the scene contains multiple features that are similar in appearance and locally adjacent. For example, Figure 6.1 shows two images in which three features in the first image have been incorrectly matched with features that are nearby in the second image but are different points in the scene. Because camera motion is unknown, feature correspondences must be made without prior knowledge of the expected image motion and therefore it is difficult to check correspondences for consistency. Shapiro [48] estimated the projected positions of feature points from their previous trajectory which avoided some correspondence errors, but this required more than two images and did not enforce global consistency.

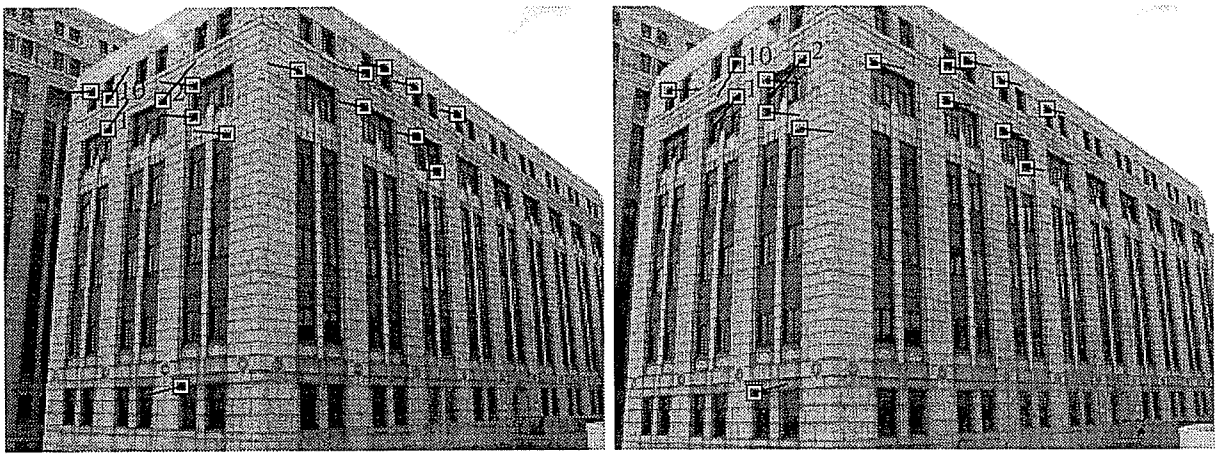


Figure 6.1: Example of *correspondence errors*. Features #1, #2 and #10 are incorrectly matched to features that are similar in appearance and adjacent in the images, but are different points in the scene.

Correspondence errors cause large apparent errors in projected positions of some feature points, which can easily be an order of magnitude greater than the error due to noise. Whereas noise can be modelled by a zero mean Gaussian distribution, correspondence errors have a large non-uniform error. The frequency with which correspondence errors occur and their magnitude depends on the ability of the feature detector to discriminate features, the motion of the camera, and the appearance and density of features in the scene, over which the observer has limited control.

6.3 False Features and Non-Rigid Motion

Outliers can be also introduced when detected features do not correspond to rigid points in the scene. This can occur when an image feature is generated by a depth discontinuity in the scene, as shown in Figure 6.2. These are called *false features* because they do not represent rigid feature points in the scene. False features are equivalent to non-rigid motion because their apparent positions migrate along the occluding contour as viewpoint changes. False features and non-rigid motion are difficult to detect without first recovering full 3D structure because their projected image errors between any two images are small and they follow smooth image trajectories that are similar to those of rigid feature points.

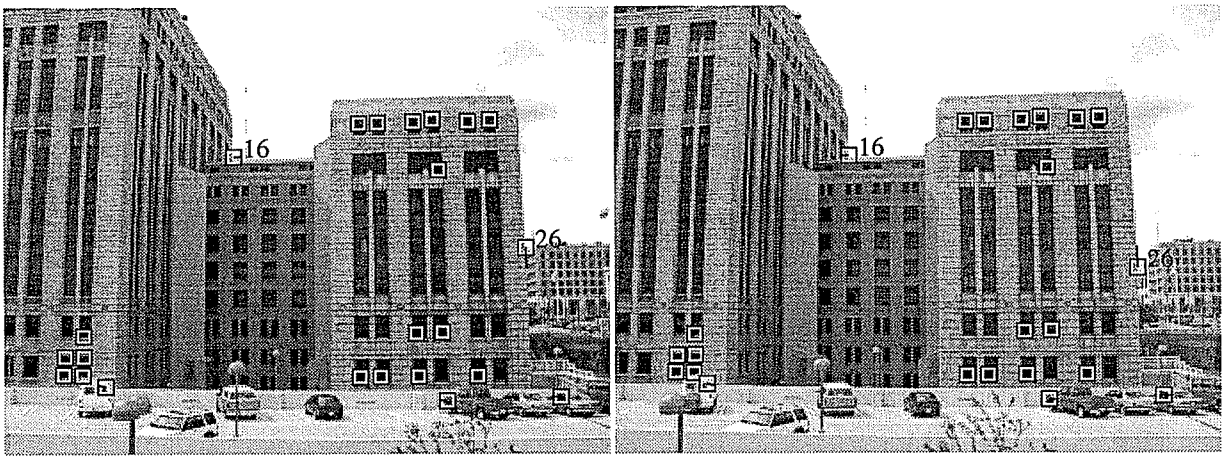


Figure 6.2: Example of *false features*. Features #16 and #26 are caused by a depth discontinuity and do not represent rigid points in the scene. Instead, their apparent 3D position changes with viewpoint. False features are difficult to detect because their projected image motion is smooth and similar to that of rigid feature points.

False features and non-rigid motion are caused by the fundamental limited ability of feature detectors to identify rigid points in the scene. Although the rigidity constraint precludes non-rigid motion, false features cannot be wholly avoided without constraining scene structure.

6.4 Summary

The KLT tracker performs well in a variety of real scenes and does not require extensive parameter adjustment. However, as with all feature trackers, it is not perfect. Correspondence errors and false features occur under many different conditions and are impossible to predict or avoid without *a priori* knowing the scene structure and camera motion. Both types of outliers effectively violate the rigidity constraint on which SFM is based and if they go undetected they can overwhelm subsequent SFM analysis. The detection of outliers caused by correspondence errors, false features and non-rigid motion is therefore an important component of a robust SFM technique, and is the topic of the Chapter 7.

Chapter 7
Outlier Detection

Recovering scene structure and camera motion from image sequences is subject to many sources of error. Some errors are introduced during quantization or are caused by imperfections of the camera model and can be reasonably modelled as Gaussian noise. Noise can be addressed by examining additional feature points and images and performing least-squares analysis or equivalent; e.g., Kalman filtering [48], [2], [72], [32], Singular valued decomposition [40], [70], [42], [9], [46] or non-linear optimization [55], [53]. Others sources of error, such as correspondence errors and non-rigid motion, cause large non-uniform errors that cannot be modelled as Gaussian noise, i.e., they introduce *outliers*. Outliers cannot be wholly avoided in real image sequences. Nevertheless, the issue of outliers has been widely ignored in SFM and the results of feature detection and tracking are typically manually checked for outliers prior to SFM analysis.

This chapter describes two complementary approaches to automatic outlier detection. The first approach, called *Random Sample Consensus* (RANSAC), incrementally grows a solution by adding only consistent data points. The second approach, called *pruning*, incrementally trims a solution by removing inconsistent data points from it. Both have been used by some SFM techniques [76], [15], [63]. As explained in this chapter, RANSAC is unsuitable for outlier detection in the projector model and therefore pruning is used instead. An example is given using synthetic 2D images containing outliers that shows how detecting and pruning the outliers gives a more reliable reconstruction of scene structure and camera motion.

7.1 Random Sample Consensus

Random Sample Consensus is a general technique proposed by Fischler and Bolles [15] for parameter estimation from noisy data that contains outliers. RANSAC was used by Fischler and Bolles for the problem of determining camera location from landmarks; i.e., the location determination problem (see Appendix B). More recently is was used by Torr and Murray [63] as a robust method for computing the fundamental matrix from a pair of images that contains some correspondence errors.

In the RANSAC approach, a model $\mathbf{M}$ is fitted to a set of noisy data points $\mathbf{P}$, of which an unknown subset are outliers. In the context of SFM, the model $\mathbf{M}$ describes the parameters of camera motion and scene structure, and the data points are the projected positions of the features points in the images. First, a subset $\mathbf{P}_{min}$ of the data points are randomly selected to obtain an initial estimate $\mathbf{M}_0$ of the model (i.e., a minimal data solution). The remaining data points are then examined to identify which are consistent with $\mathbf{M}_0$. In particular, if the error ε of a point is consistent with the noise model, e.g., $\varepsilon \leq 3\sigma$, then the point is added to the *consensus set* of $\mathbf{P}_{min}$. If a point is found to be inconsistent with $\mathbf{M}_0$, e.g., $\varepsilon > 3\sigma$, then it is considered an outlier and discarded. Figure 7.1 shows a example of RANSAC applied to a simple linear data fitting problem. Two data points $\{2, 2.3\}$ and $\{3, 3.5\}$ provide the initial estimate. Of the remaining data points, the error in $\{5, 15.1\}$ and $\{15, 5.0\}$ exceed the noise threshold and thus are considered outliers. The solution is recomputed using the consensus set to obtain a more reliable fit to the data points.

RANSAC obtains the initial model $\mathbf{M}_0$ from a random subset of data points, and it quite possible that one or more of these points are outliers. If the number of remaining points that are inconsistent exceeds the expected outlier frequency then it is most likely caused by a poor initial model, and a new model is computed from a different subset of points.

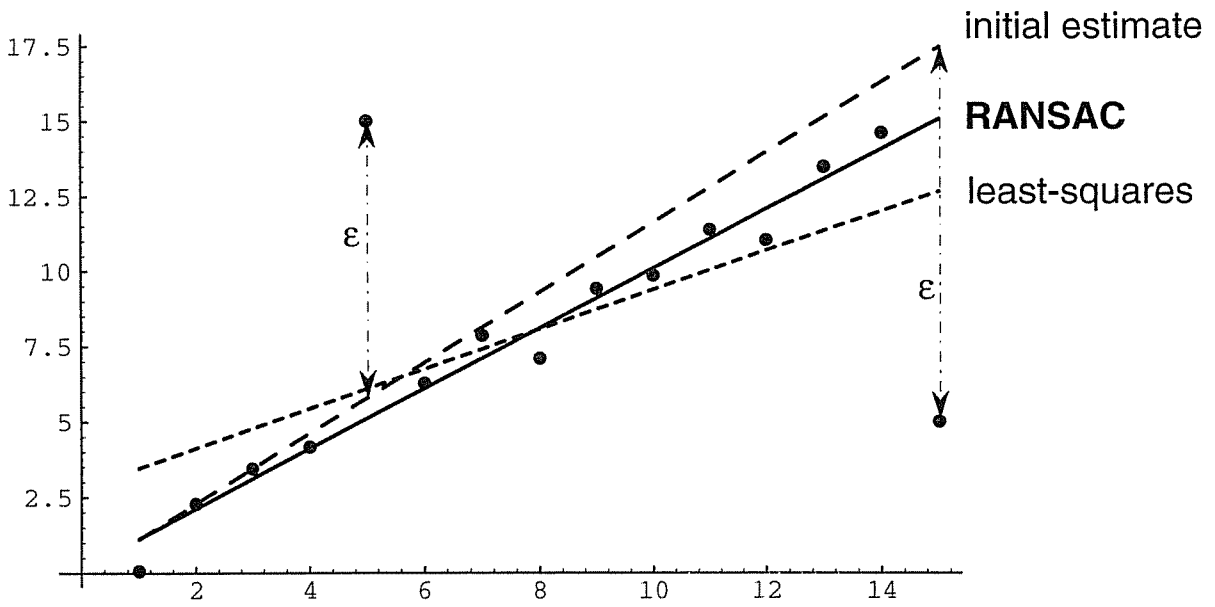


Figure 7.1: Example of RANSAC applied to linear data fitting. RANSAC obtains an initial estimate from two data points and detects outliers based on their residual error ε. Outliers are ignored when computing the final solution.

RANSAC has been shown to work well for outlier frequencies as high as 25% and it is suitable for applications where the magnitude of outliers greatly exceeds the noise level [15]. However, RANSAC has the following implicit requirements:

1. an initial estimate of the solution can be obtained from a subset of points,

2. individual points can be added to the initial model to measure their error and thereby determine whether they are outliers.

For example, RANSAC was used by Torr and Murray [63] to compute the fundamental matrix from two images. First, a minimal subset of feature point correspondence pairs were selected to estimate the camera motion parameters. The remaining pairs of features were then examined to determine whether they were consistent with the estimated camera motion. A data point in this context is a pair of corresponding features. RANSAC can be used to identify outliers in this case because the error introduced by a single pair of features can be measured to determine if the pair are consistent with the previously estimated camera motion.

Projected Error Refinement, however, does not examine pairs of features but rather individual projectors. In particular, the error in an individual projector cannot be measured without first estimating the position of the associated feature point, which requires adding other projectors in order to perform triangulation. However, by doing so, it is not possible to determine which of these projectors are outliers. In other words, RANSAC cannot be used for outlier detection in the Projected Error Refinement approach because all the projectors for an image or all the projectors for a feature point must be added at once, making it impossible to identify which are outliers.

7.2 Pruning Outliers

As described above, a robust SFM solution cannot be obtained by adding only consistent projectors to a minimal data solution because projectors must be added in groups. However, it is possible to *remove* or *prune* inconsistent projectors from an existing solution. Whereas RANSAC starts from a small but good solution and adds only consistent data points, pruning starts from a large but poor solution and removes inconsistent data. The rationale behind pruning is that least-squares minimization attempts to distribute the projected error evenly over all the projectors. As a result, in most cases a least-squares solution fits the valid projectors better than the outliers. In particular, the residual error of outliers will be somewhat greater, as illustrated in Figure 7.2. The extent to which outliers can be detected based on their residual error depends on their frequency, magnitude and distribution, and it is entirely possible for outliers to have a smaller residual error than valid data points in some cases. In general, however, the error of outliers is greater than that of non-outliers, in which case they can be detected based on their residual error in the initial solution.

Most SFM techniques that perform outlier detection use some form pruning; that is, they compute a solution based on many features and/or images and then remove inconsistencies. For example, Szeliski and Kang [55] performed LM optimization and discarded feature points whose projected residual error exceeded 3σ. The residual error after optimization can

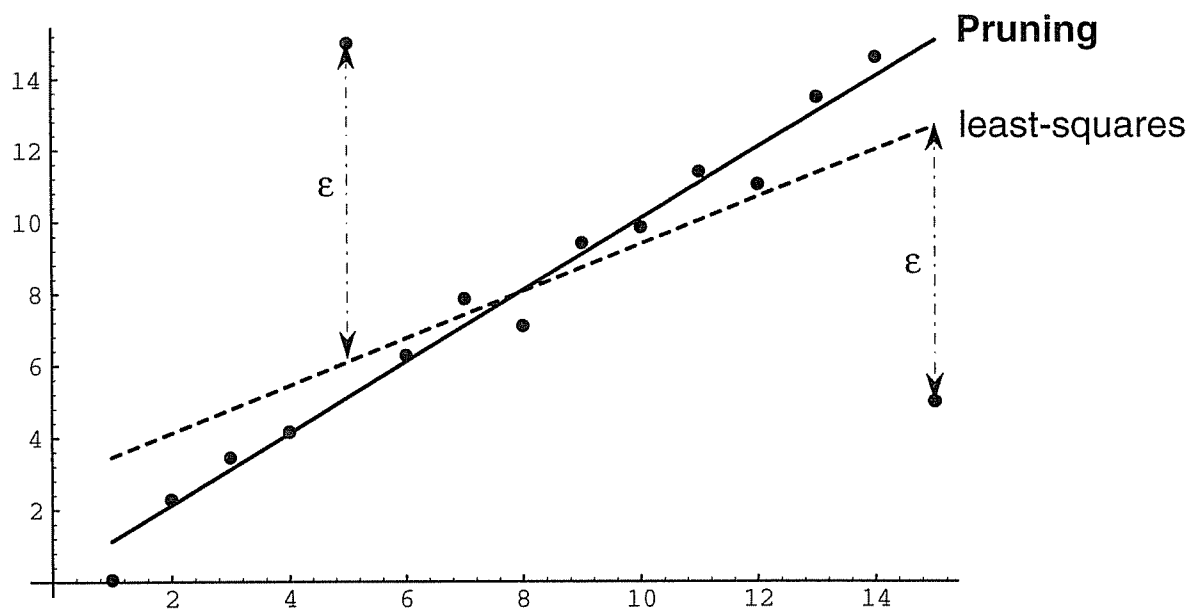


Figure 7.2: Example of pruning applied to linear data fitting. Outliers are detected based on the magnitude of their residual error ε in the initial least-squares solution. Pruning the outliers and re-computing the solution gives a better solution.

---

also be used to indicate the presence or absence of non-rigid motion; e.g., Boult and Brown [5] examined the residual after SVD to segment a non-rigid scene into rigid components, and McReynolds and Lowe [33] determined whether a scene is rigid based the residual error after LM optimization. Other SFM techniques use weighted least-squares, or *M-estimators*, to adjust the weight of each projector during optimization according to the residual error, with the effect that outliers have less impact on corrupting the solution [63]. Other techniques replace least-squares minimization with more robust optimization methods, such as *least median of squares* (LMedS) [76], [58], [63]. LMedS minimizes the *median* of the squared errors; that is, it finds the solution whose median error is smallest. LMedS is very robust to outliers but requires a non-linear search of the solution space and cannot be reduced to least-squares, unlike M-estimators.

The outlier detection performed by Projected Error Refinement is similar to Szeliski and Kang [55]. First, the solution is computed from all the feature points and images using parallel iterative refinement. After refinement has converged, the angular projection error ε of

each projector is measured to determine whether it is consistent with the noise model, i.e., $\varepsilon \leq 3\sigma$. Two pruning options are possible. The first, called *conservative pruning*, removes only the outlier with the greatest residual error, after which the solution is re-refined. The justification for this approach is that the projector with the greatest residual error is the most likely to be an outlier. The second approach, called *liberal pruning*, removes all projectors whose error exceeds 3σ. However, because the solution has been corrupted by outliers, this approach may unintentionally prune some valid projectors. As described in Chapter 8, the occlusion of a few valid projectors does not significantly affect the solution and liberal pruning is performed because it is faster, albeit less accurate.

To summarize, after parallel iterative refinement has converged, all the projectors whose residual error exceeds 3σ are pruned. The solution is then re-refined using the remaining projectors. This process is repeated until all the projectors are consistent with the noise model.

7.3 Example

Figure 7.3 shows a synthetic scene in $\Re^2$ projected to three noisy images, where four of the projectors $\{\mathbf{v}_{0,3}, \mathbf{v}_{1,5}, \mathbf{v}_{2,6}, \mathbf{v}_{4,5}\}$ are outliers and have an angular projection error of 3°. The first refined solution, with outliers included, is shown in Figure 7.4 and has structure and motion errors of 0.10 and 0.14, respectively. Of the four outliers, $\{\mathbf{v}_{0,3}, \mathbf{v}_{1,5}, \mathbf{v}_{4,5}\}$ have an angular projection error exceeding 3σ in the solution. These three projectors are therefore pruned and the solution is re-refined using the remaining projectors. At the end of the second refinement stage all the projectors have an error less than 3σ, including the fourth outlier $\mathbf{v}_{2,6}$. The final solution is shown in Figure 7.5 and has structure and motion errors of 0.05 and 0.08, respectively. In this example, the removal of three of the four outliers resulted in approximately a two-fold improvement in the recovered structure and motion parameters, although

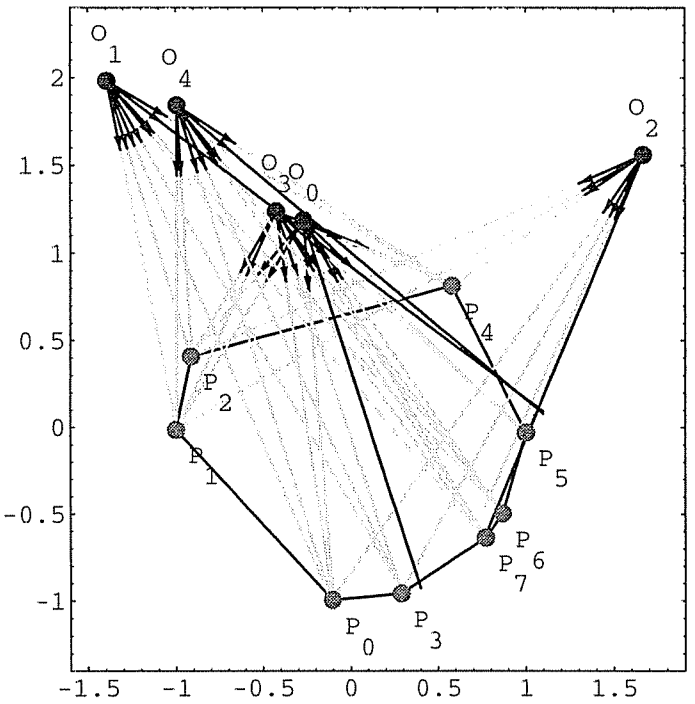
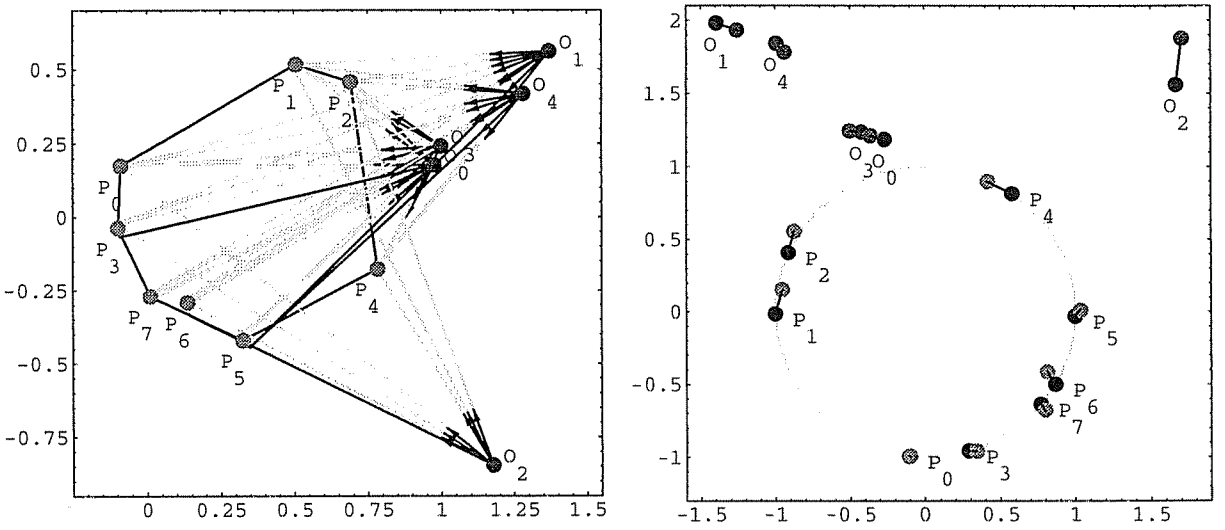
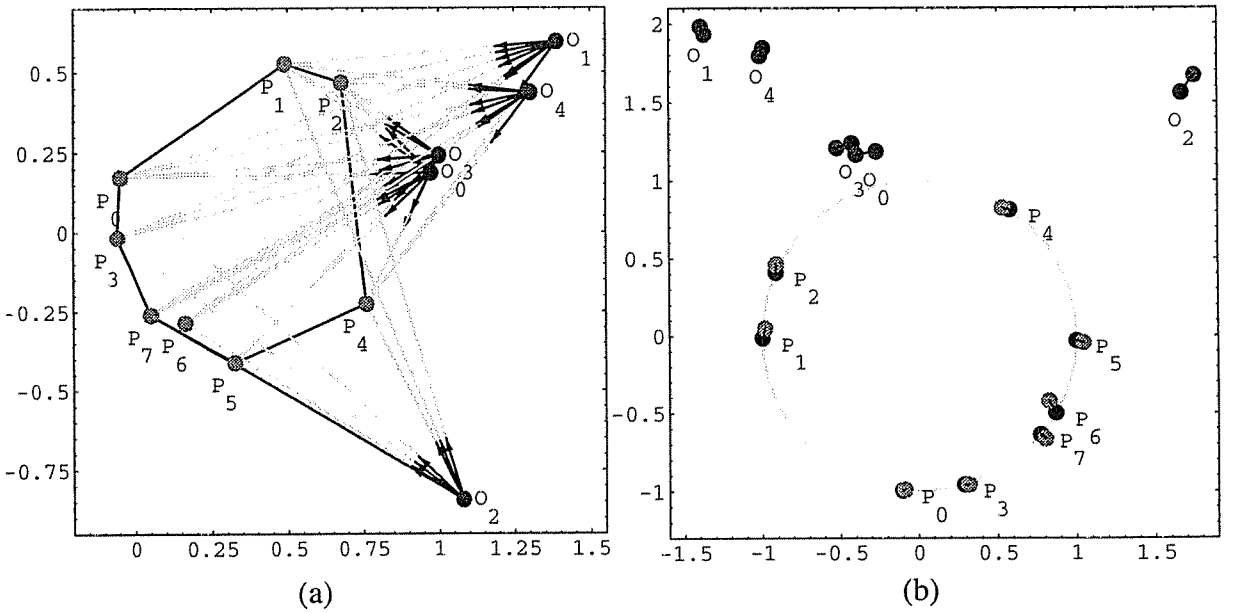


Figure 7.3: The synthetic scene from Figure 5.4 where four of the projectors are now outliers (shown in black) and have an angular projection error of 3°.



(a) (b)

Figure 7.4: (a) The first refined solution to the images projected in Figure 7.3 without outlier detection. (b) This solution is transformed to the original scene's coordinate system to measure the reconstruction error (the original features and camera positions are shown in black).



(a) (b)

Figure 7.5: (a) The final refined solution after pruning outliers. (b) This solution transformed to the original scene's coordinate system. Note: the outlier from $\mathbf{o}_2$ to $\mathbf{p}_6$ could not be identified based on its residual angular projection error.

the fourth outlier could not be identified based on its residual error.

7.4 Summary

Correspondence errors, non-rigid motion and false features cannot be modelled as Gaussian noise because they have large non-uniform error distributions. All these sources of error are therefore considered as *outliers*; in other words, they are modelled as anything inconsistent with the noise model. In both RANSAC and pruning, detecting outliers relies on examining the residual error of the projected feature points. This error should therefore be meaningful. For example, Projected Error Refinement explicitly minimizes the observable angular projection error in the images. The Factorization Method [59] and essential matrix [29], [64], [63], on the other hand, perform least-squares analysis of systems of linear equations, where the geometric or visual interpretation of the residual error is unclear. If outliers are to be detected based on their residual error then it is important that this error is well-defined in terms of image noise. Otherwise, the risk of failing to detect outliers or detecting false positives is

increased.

Non-linear optimization-based SFM techniques, such as Projected Error Refinement, are particularly suited to outlier detection because individual projected feature points can be removed [55], unlike linear methods which require all the feature points and images to be present [48], [59], [64], [29], [72], [42]. Although outliers can be removed from linear systems after the solution is obtained, the solution must then be recomputed from scratch from the new image data [76]. In Projected Error Refinement, projectors can be removed (or added) at any time and refinement simply continues.

Outlier detection is an open problem and both RANSAC and pruning can fail to detect all the outliers. Nonetheless, outliers are unavoidable in real image sequences and outlier detection is an important component of any robust SFM technique for general-purpose applications.

Chapter 8
Occlusion

Occlusion occurs when a feature is not visible in an image because the surface on which it lies faces away from the observer, because a surface closer to the observer obstructs the view, or because the appearance of the feature has changed and it is no longer detected as such (i.e., a *dropout*). Occlusion is an intrinsic property of real scenes, in fact, the appearance and disappearance of surfaces is one of the strongest depth cues in human vision [16]. The only circumstances where occlusion does not occur is in the special case of a convex object rotating through a limited angle such that all its forward facing surfaces remain visible. Nevertheless, occlusion is handled poorly, if at all, by most SFM techniques. This is one of the reasons why SFM has yet to be used extensively in real applications, where occlusion is ubiquitous.

SFM research has largely focussed on determining efficient, i.e., *linear*, solutions to the inverse projection problem. However, as described in Chapter 3, efficiency is only one of many requirements of a general-purpose SFM technique. It must also accurately model scene structure and camera motion and be robust to common errors, such as noise, outliers and missing features; i.e., occlusion. Linear methods are typically defined for a small number of feature points and images, or they examine multiple features and images and perform linear least-squares optimization. Linear methods are fast because solving linear systems of equations is efficient; for example, SVD or Kalman filtering. However, the coefficients of these linear systems are derived from the projected positions of feature points, which does not allow for the presence of "no information" as occurs when a feature point is missing or occluded. Linear SFM techniques therefore require *all* the features to be present in *all* the images. In effect, the

ability to handle occlusion is sacrificed for efficiency.

The most common solution to the occlusion problem is to examine a subset of features and images where all the features are visible in all the images [17], [72], [49], [50], [51], [2], [9]. This is implicit in SFM techniques based on image pairs [64], [29], [38], [63], [76], [71]. However, examining only a subset of the available projected information means the solution is sub-optimal. For example, the reliability of SFM techniques based on image pairs is limited because additional images cannot be included in the solution even if they are available. Further, in long image sequences few feature points are visible in all the images. In order to recover complete scene structure, subsets of features must therefore be examined, which introduces the additional non-trivial problem of how to combine partially overlapping solutions. Occlusion also poses a problem for Kalman filtering because the state vector representing the structure parameters is fixed. A novel solution to this problem was proposed by McLauchlan and Murray [32] who replaced the state vector by a *variable state-dimension filter* which allowed the structure parameters to be added and removed dynamically.

A different approach to the occlusion problem taken by some linear SFM techniques is to examine all the features and images and 'fill in' the missing coefficients [59], [42], [70]. This is accomplished by recovering the 3D positions of occluded feature points from a (complete) subset of images, and then re-projecting these points back into the images where they are occluded, a process called *hallucination*. However, hallucination precludes optimality because least-squares optimization cannot distinguish between the original and derived data. Hallucination also increases the risks of introducing artificial outliers.

The handling of occlusion by linear SFM techniques is *ad hoc* at best. Non-linear optimization-based SFM techniques are better suited to dealing with occlusion because individual projected feature points can be added or removed arbitrarily. For example, in Szeliski and Kang [55] occluded feature points were given zero weight and so did not contribute to parameter optimization. Similarly, Spetsakis [53] minimized the distance between non-current pro-

jectors that allowed individual projectors to be absent from any image.

8.1 Occlusion in Projected Error Refinement

If a feature point is not projected to an image then the feature should have no influence on the computed position of that image's optical center or the rotation of the image around it. Similarly, an image should have no influence on the computed positions of feature points that are not visible in the image. This obvious property is built-in to Projected Error Refinement. In particular, when refining the extrinsic camera parameters with respect to the (fixed) feature points, only the visible projectors in the image define error terms that are considered during optimization - missing features have no influence. Likewise, when refining the feature points with respect to the (fixed) camera poses, only the projectors defined by the images in which each feature is visible are considered when optimizing the position of that feature point. This is equivalent to Szeliski and Kang [55] where the weight associated with occluded feature points is set to zero.

Occlusion is handled naturally by Projected Error Refinement. Any feature may be present or absent in any image - only the visible projectors are examined when refining the structure and motion parameters. This property also enables features to be added or removed at any time. This facilitates outlier detection and allows new images and their associated projectors to be added without having to recompute the solution, making the approach very suitable for processing long image streams. The ability to dynamically add and remove projectors is a key feature of Projected Error Refinement.

8.2 Example

Figure 8.1 shows a synthetic scene in $\Re^2$ projected to three noisy images, where 8 of the 40 projectors are occluded, i.e., 20% occlusion. The refined solution is shown in Figure 8.2 and has mean structure and motion errors of 0.2 and 0.6, respectively. This solution represent the optimal placement of the feature points and camera centers that minimizes the angular projec-

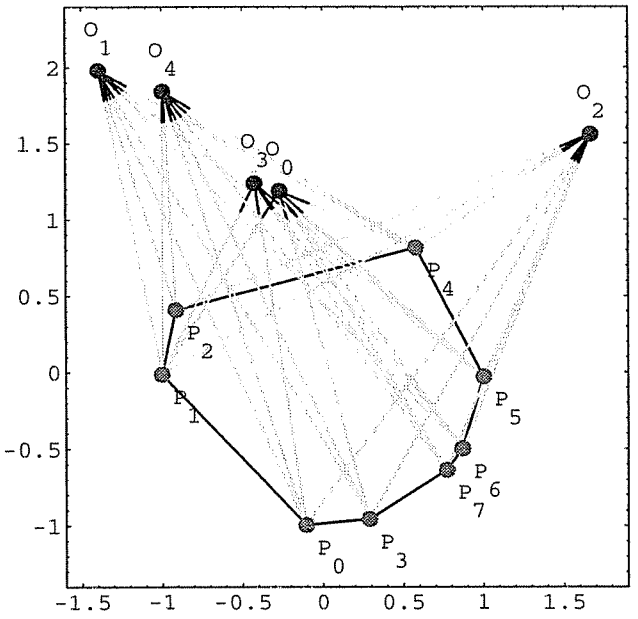


Figure 8.1: The synthetic scene from Figure 5.4 except that 20% of the projectors are occluded.

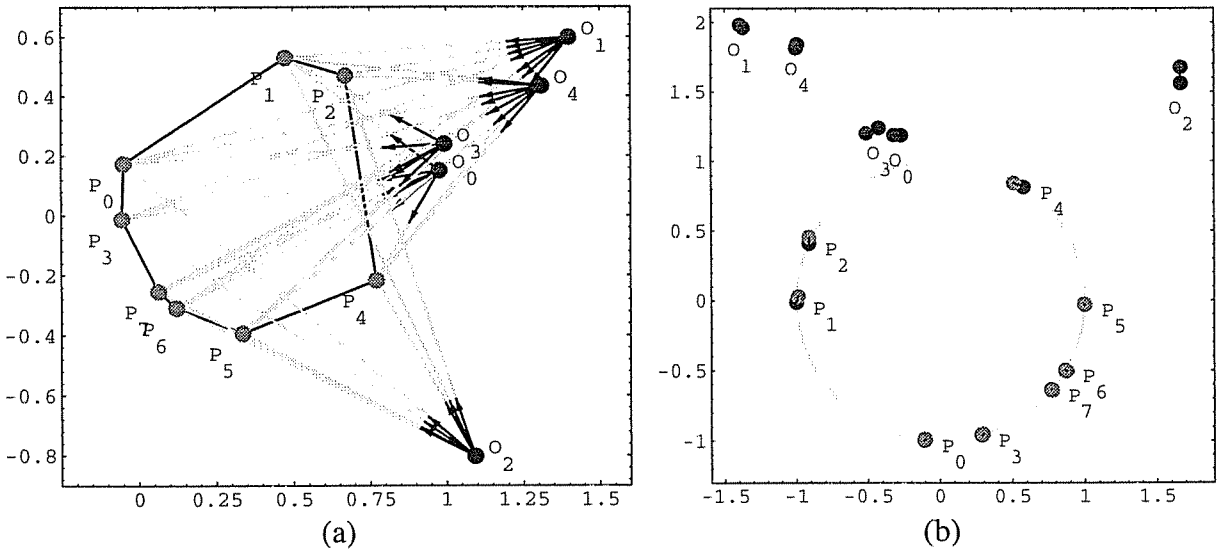


Figure 8.2: (a) The refined solution to the occluded images projected in Figure 8.1. (b) The refined solution is transformed to the original scene's coordinate system to measure the reconstruction error.

tion error in the *visible* projectors. Occlusion affects the accuracy of reconstruction because it reduces the amount of data over-determining the noise errors. However, as shown, scene structure and camera motion is still be reliably recovered even with a large amount of occlusion.

Chapter 9
Experimental Results

This chapter describes experimental results on synthetic and real image sequences. Synthetic images allow the camera motion, scene structure and projection conditions to be controlled and adjusted to observe their effect on reconstruction. Quantitative error analysis is also possible because ground-truth data is available. Several real image sequences are examined to show the performance of Projected Error Refinement on real image data under actual viewing conditions. However, ground-truth data for real image sequences is difficult to obtain, making a quantitative analysis of the recovered structure and motion parameters difficult.[22] With the exception of one of the real image sequences (Figure 9.16), the real image sequences are uncalibrated and therefore only a qualitative examination of the solutions is possible.[23]

9.1 Synthetic Image Sequences

A series of experiments was conducted on synthetic images to analyze the performance of Projected Error Refinement under simulated viewing conditions. The effect of the following parameters are investigated:

---

[22] The SFM field presently lacks a standardized suite of calibrated image data for the quantitative analysis and comparison of different SFM techniques.

[23] The uncalibrated image sequences were obtained from the CMU Vision and Autonomous Systems Center's image database.

1. Refinement iterations - the number of iterations that the structure and motion parameters are refined is varied from 0, i.e., no refinement, to 20 iterations.

2. Image noise - the direction vectors of the projectors are corrupted by angular Gaussian noise, varying from $\sigma = 0°$, i.e., perfect perspective projection, to $\sigma = 0.2°$, corresponding to approximately 4 pixels noise for a camera with a 35mm lens and a 640 pixel wide image.

3. Features and images - the number of features and images is varied from 3 to 13 images for 2D perspective projection, and from 2 to 12 images for 3D perspective projection. From 5 to 15 features are projected in both cases.

4. Occlusion rate - the percentage of projected feature points that are occluded in the images is varied from 0%, i.e., no occlusion, to 50% occlusion.

5. Outlier frequency - the percentage of projected feature points that are outliers is varied from approximately 2% to 10%; i.e., from 1 to 6 projectors for the default of 10 features and 6 images.

6. Outlier magnitude - the magnitude of outliers is varied from 5 times the image noise to 10 times; i.e., 0.5° (10 pixels) to 1° (20 pixels).

Except from the number of features and images, each experiment adjusts one parameter in isolation. The numbers of features and images are varied simultaneously to determine the best working set size - the total number of projectors directly determines the efficiency of non-linear optimization. The default parameter values are: 10 refinement iterations, $\sigma = 0.1°$ image noise (i.e., 2 pixels), 10 features projected to 6 images, no occlusion, and no outliers.

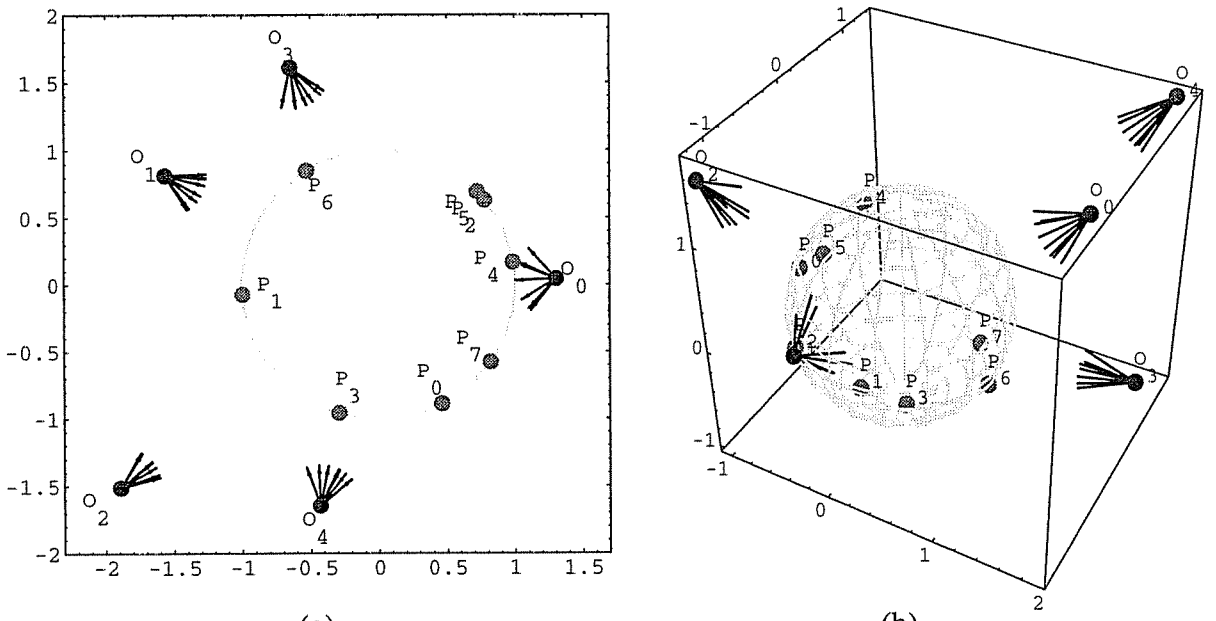
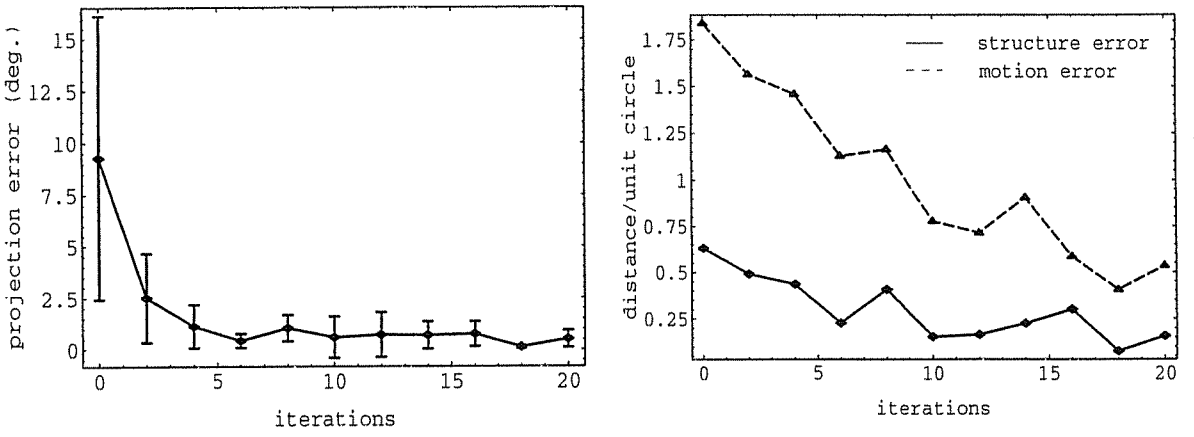


Figure 9.1: A synthetic scene is constructed by placing feature points $\mathbf{p}_i$ around (a) a unit circle for 2D perspective projection or (b) a unit sphere for 3D perspective projection. The projected images are the directions of the feature points from the optical centers $\mathbf{o}_j$.

Both 2D and 3D perspective projection are examined. 3D perspective projection is of most interest because it closely models camera projection. 2D perspective projection is also useful for applications involving predominantly planar motion. Although a full 3D model could be used in these situations, a 2D model may be sufficient and is more efficient.

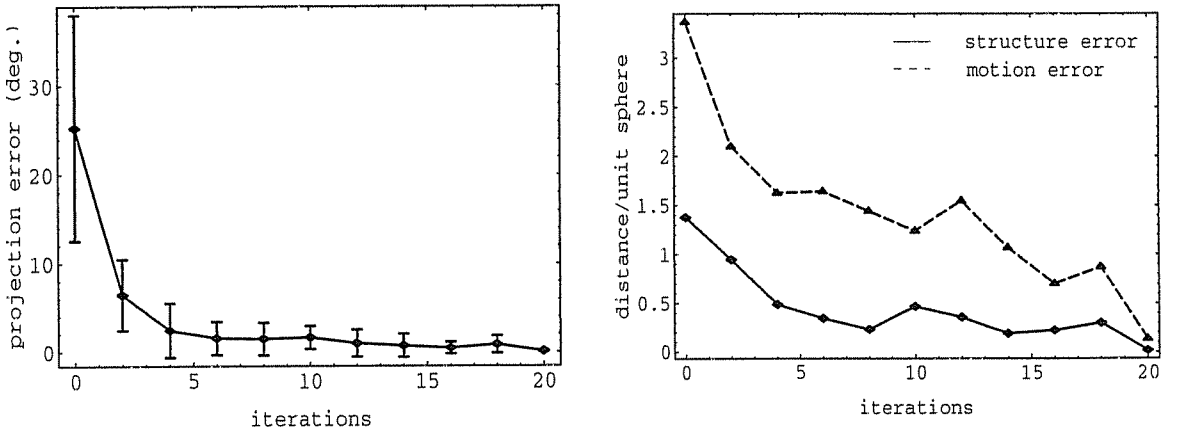
9.1.1 Synthetic Images

Synthetic images were generated by randomly placing feature points and camera centers and recording the relative directions of the features from the optical centers, as shown in Figure 9.1. For 2D perspective projection, feature points are placed on a unit circle with the optical centers distributed around it. In 3D perspective projection, the feature points are placed on a unit sphere. Placing features on a circle or sphere in this way facilitates measuring the structure and motion error of the resulting solution, as described in Appendix C, and does not affect the generality of the results. Each experiment is repeated 10 times with a different scene and the results are averaged. The 95% confidence interval is computed for the angular projec-



(a) (b)

Figure 9.2: (a) The angular projection error and (b) the structure and motion error as a function of the number of *refinement iterations* for *2D perspective projection.*



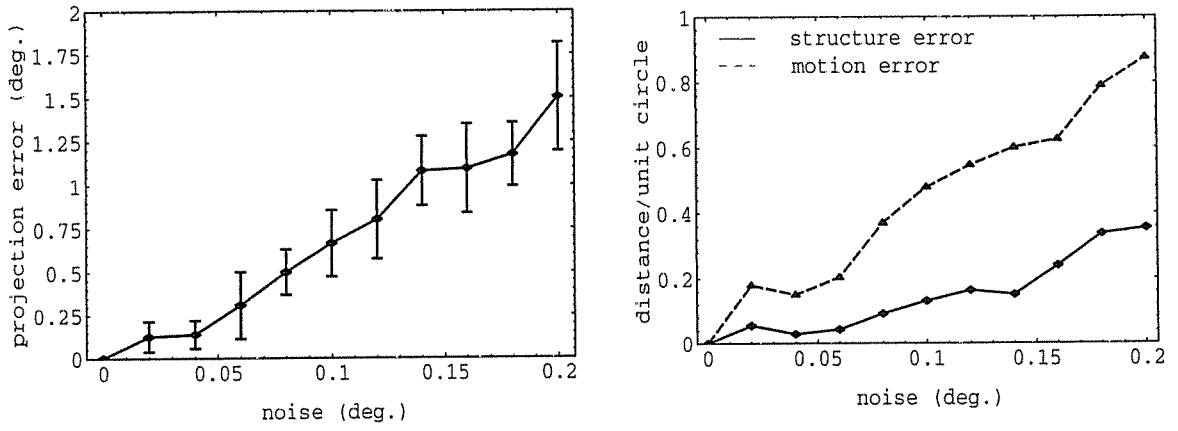
(a) (b)

Figure 9.3: (a) The angular projection error and (b) the structure and motion error as a function of the number of *refinement iterations* for *3D perspective projection.*

tion error.

9.1.2 Experiment 1: Refinement Iterations

This experiment examines how the number of refinement iterations affects the projected image error and the accuracy of reconstruction. The results for 2D and 3D perspective projection are shown in Figure 9.2 and Figure 9.3, respectively. Zero iterations corresponds to the initial estimate obtained from the minimal data solution. As shown, the initial estimate gives a



(a) (b)

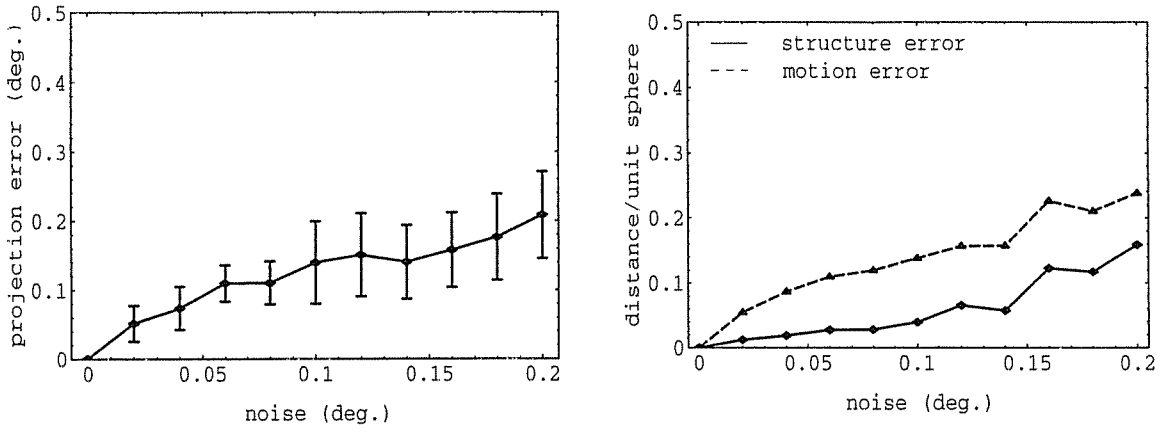
Figure 9.4: (a) The angular projection error and (b) the structure and motion error as a function of the *image noise* for *2D perspective projection.*

poor reconstruction of scene structure and camera motion, and has a large variance in the angular projection error because none of the features or images have yet been refined. However, as the number of refinement iterations increases, the projection error decreases rapidly, with a corresponding decrease in the structure and motion error. After approximately 10 iterations there is negligible further reduction in the projected image error, although the recovered structure and motion error continues to improve somewhat.[24]

9.1.3 Experiment 2: Image Noise

This experiment examines how the amount of image noise affects the projected image error and the accuracy of reconstruction. The results for 2D and 3D perspective projection are shown in Figure 9.4 and Figure 9.5, respectively. With zero noise, i.e., ideal perspective projection, the scene structure and camera motion are recovered perfectly. As noise increases, the projected image error increases comparatively; in particular, the magnitude of the angular projection error after refinement is approximately the same as the angular noise in the original images. As shown, the structure and motion error is related to the projected image error, indi-

[24] The motion error is greater than the structure error because of how the solution is transformed to the original scene's coordinate system, as described in Appendix C, not because camera motion is less reliably recovered than scene structure.



(a) (b)

Figure 9.5: (a) The angular projection error and (b) the structure and motion error as a function of the *image noise* for *3D perspective projection.*

cating that minimizing the angular projection error has the desired effect improving the accuracy of reconstruction. Comparing the experimental results for 2D and 3D indicates that 2D perspective projection appears to be more sensitive to noise than 3D perspective projection. This may be due to the additional dimension in 3D imposing a stronger constraint on the location of a feature point, relative to its projectors, than the planar 2D case.

9.1.4 Experiment 3: Number of Features and Images

This experiment examines how the number of features and images affects the accuracy of reconstruction. The results for 2D and 3D perspective projection are shown in Figure 9.6 and Figure 9.7, respectively.[25] One of the goals of this experiment is to identify a good working set size. This involves a trade-off between the accuracy of reconstruction and the efficiency of refinement - more features and images gives a better reconstruction but refinement takes longer. As shown, for 2D and 3D perspective projection, both the number of features and the number of images have a similar effect of improving the recovered scene structure and camera

[25] The mean angular projection error in the solution is not shown for this experiment because it is independent of the number of features and images. For example, the projected image error for the minimum number of features and images, i.e., the minimal data solution, is always zero.

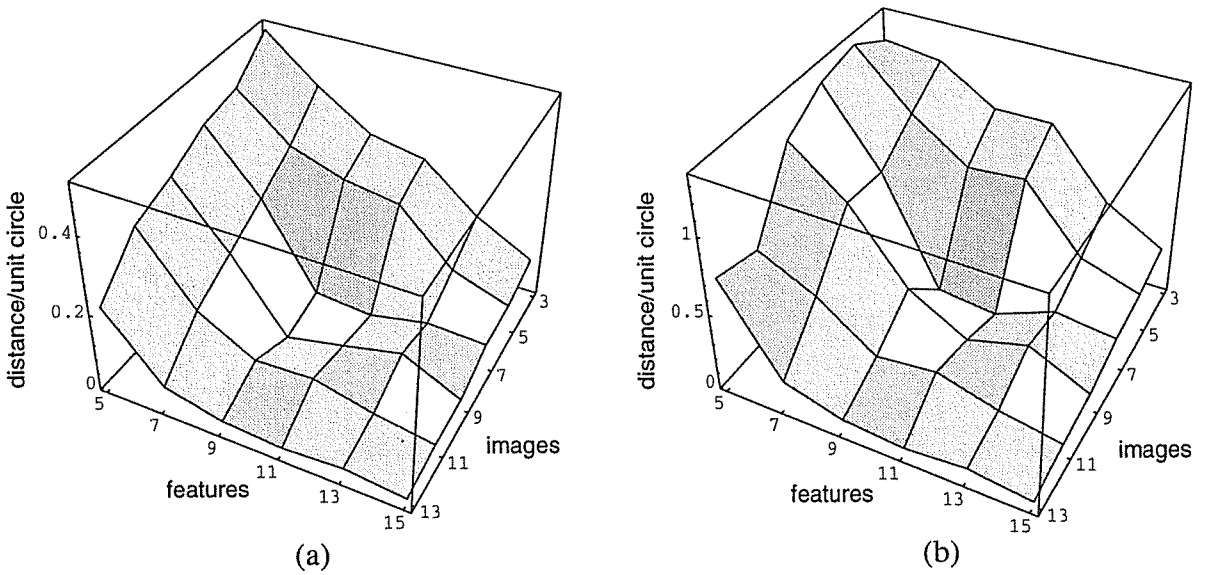


Figure 9.6: (a) The structure error and (b) the motion error as a function of the number of *features and images* for *2D perspective projection.*

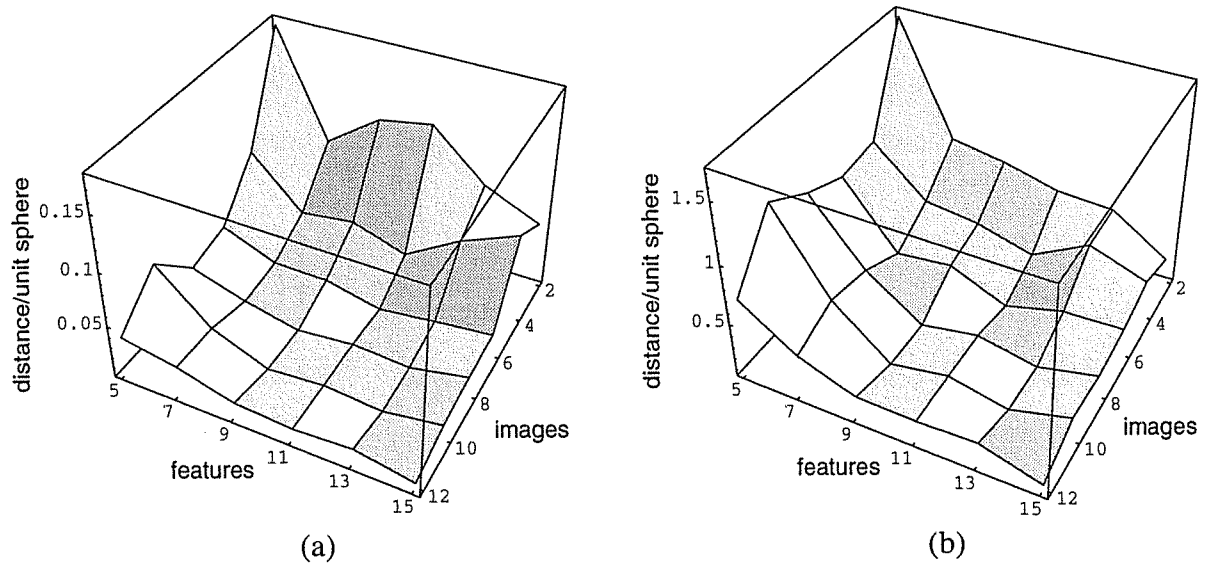
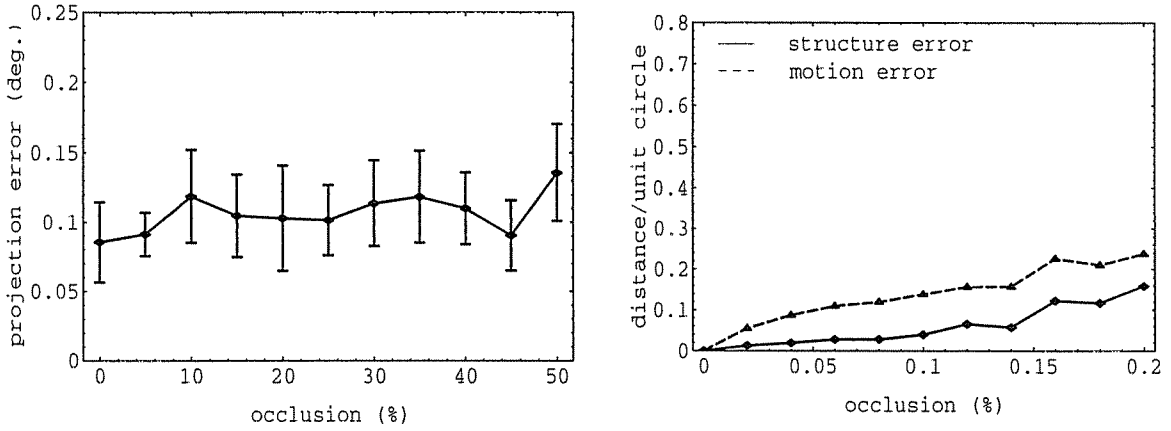


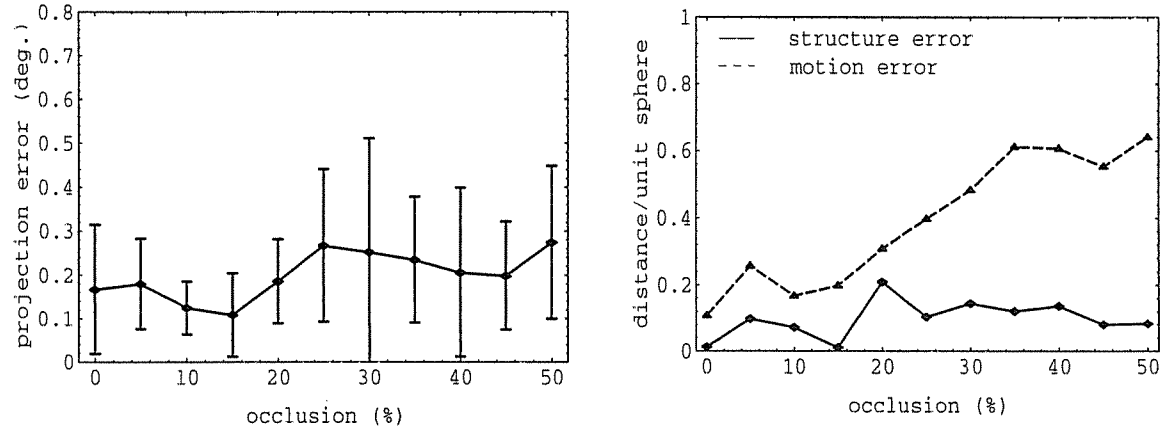
Figure 9.7: (a) The structure error and (b) the motion error as a function of the number of *features and images* for *3D perspective projection.*

motion. There is only a small reduction in the structure and motion error after approximately twice the minimal number of features and images, indicating that a good working set size is about 10 features and 5 images. These results also indicate that both additional features *and* additional images are necessary for reliable reconstruction; that is, examining only more fea-



(a) (b)

Figure 9.8: (a) The angular projection error and (b) the structure and motion error as a function of *occlusion* for *2D perspective projection.*



(a) (b)

Figure 9.9: (a) The angular projection error and (b) the structure and motion error as a function of *occlusion* for *3D perspective projection.*

tures or only more images has a limited effect on improving the solution.

9.1.5 Experiment 4: Occlusion

This experiment examines how the amount of occlusion in the images affects the projected image error and the accuracy of reconstruction. The results for 2D and 3D perspective projection are shown in Figure 9.8 and Figure 9.9, respectively. Occlusion reduces the number of visible projectors that constrain the structure and motion parameters. Therefore, increasing the

amount of occlusion increases the sensitivity of the solution to noise in the visible projectors. As shown, the projected image error and structure error increase slightly with increasing occlusion. In these experiments, the occluded projectors are selected randomly. Because the remaining projectors are evenly distributed they still largely negate the effects of noise, which is reflected by only a small increase in structure error. The larger increase in the motion error is unexpected and may be a side-effect of the registration method used for measuring reconstruction error, rather than camera motion being more sensitive to occlusion than scene structure. In any event, this experiment shows that Projected Error Refinement can reliably recover scene structure in the presence of significant occlusion.

9.1.6 Experiment 5: Outliers

These experiments examine how the presence of outliers in the images affects the projected image error and the accuracy of reconstruction. The first experiment compares the results of pruning outliers with the results when outlier detection is not performed. A synthetic scene containing 10 feature points was projected to 6 images, with 3 of the projectors (i.e., 5%) being outliers. Each experiment was repeated 10 times and the results were averaged. The results for 2D and 3D perspective projection are given in Table 9.1 and Table 9.2 respectively.

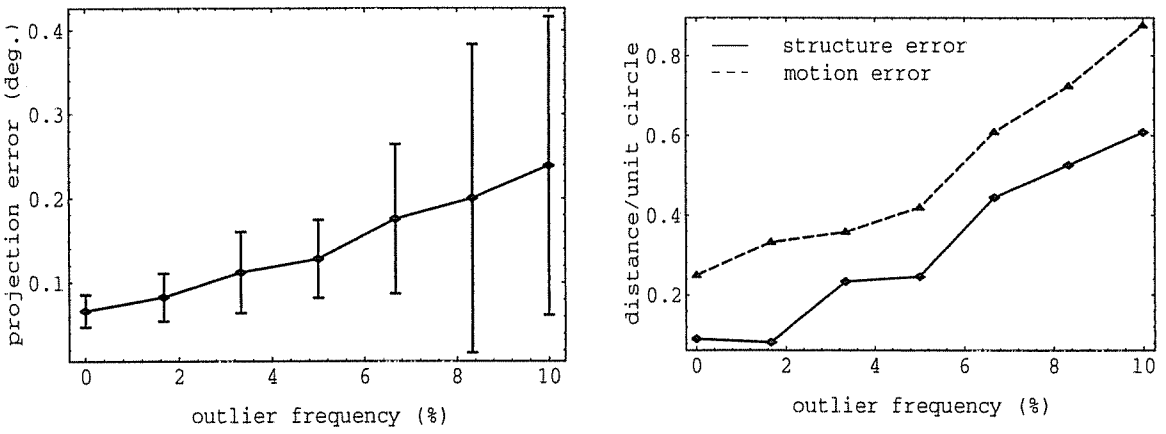
Table 9.1: Pruning outliers for 2D perspective projection.

	Projected Error	Structure Error	Motion Error
with pruning	0.07°	0.05	0.21
without pruning	0.48°	0.14	0.61

Table 9.2: Pruning outliers for 3D perspective projection.

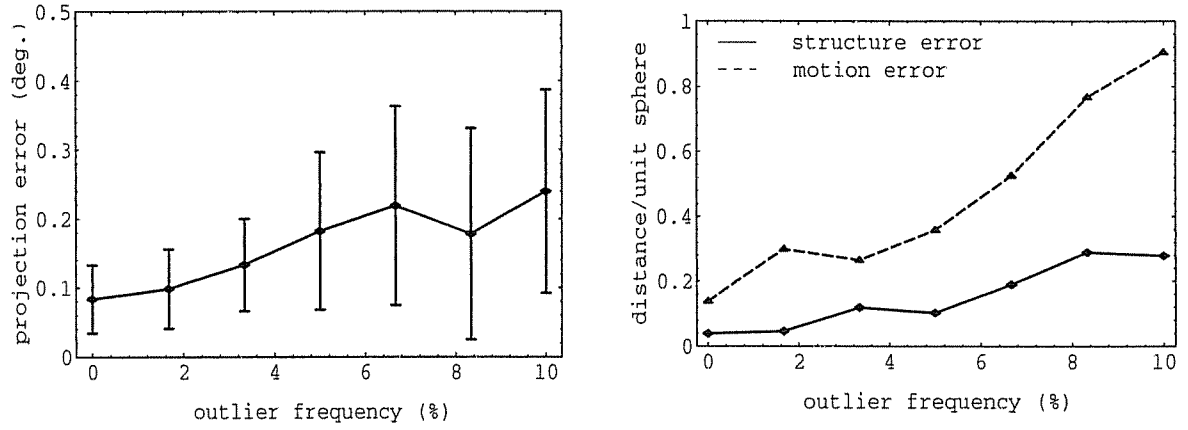
	Projected Error	Structure Error	Motion Error
with pruning	0.08°	0.03	0.18
without pruning	0.53°	0.34	0.68

These results shown that for both 2D and 3D perspective projection, eliminating outliers



(a) (b)

Figure 9.10: (a) The angular projection error and (b) the structure and motion error as a function of *outlier frequency* for *2D perspective projection.*

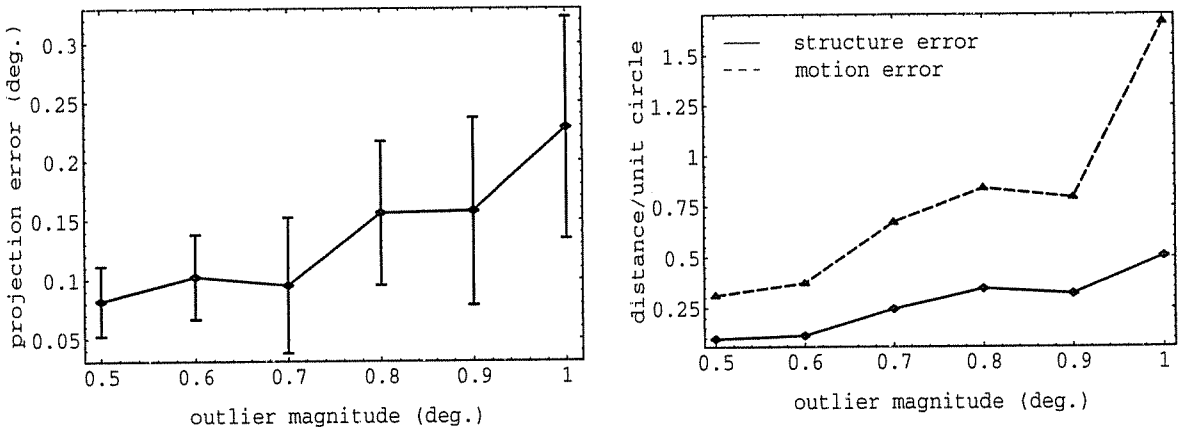


(a) (b)

Figure 9.11: (a) The angular projection error and (b) the structure and motion error as a function of *outlier frequency* for *3D perspective projection.*

improves the recovered structure and motion, up to an order of magnitude.

The second outlier experiment examines how the *number* of outliers affects the projected image error and the accuracy of reconstruction. The results for 2D and 3D perspective projection are shown in Figure 9.10 and Figure 9.11, respectively. Increasing the number of outliers reduces the accuracy of the initial refined solution (prior to outlier detection). As a result, it is more difficult to detect outliers based on their residual error and it is more likely for valid projectors to be pruned unintentionally. As shown, the mean and variance of the pro-



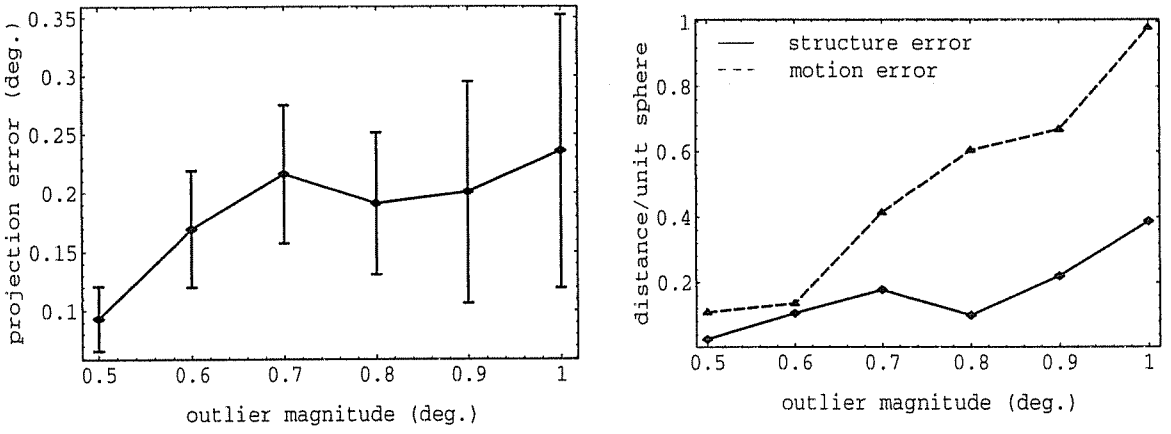
(a) (b)

Figure 9.12: (a) The angular projection error and (b) the structure and motion error as a function of *outlier magnitude* for *2D perspective projection.*

jected image error increases with increasing outlier frequency, with a corresponding increase in the reconstruction error. The larger variance indicates that some outliers were not detected and that these have corrupted the least-squares solution. This experiment shows that while pruning outliers based on their residual error is effective for a small number of outliers, it becomes less effective as the number of outliers increases.

The third outlier experiment examines how the *magnitude* of outliers affects the projected image error and the accuracy of reconstruction. The results for 2D and 3D perspective projection are shown in Figure 9.12 and Figure 9.13, respectively. These results are similar to those for outlier frequency. As the magnitude of outliers increases, the decreasing accuracy of the initial refined solution makes it more difficult to detect the outliers. Any outliers that are not detected corrupt the least-squares solution and decrease the accuracy of recovered structure and motion. This experiment shows that while pruning is effective for outliers whose magnitude is small relative to the image noise, it is less effective for outliers with a large angular magnitude.

To summarize, outlier detection can improve the reconstructed structure and motion, up to approximately an order of magnitude. Pruning outliers based on their residual error in



(a) (b)

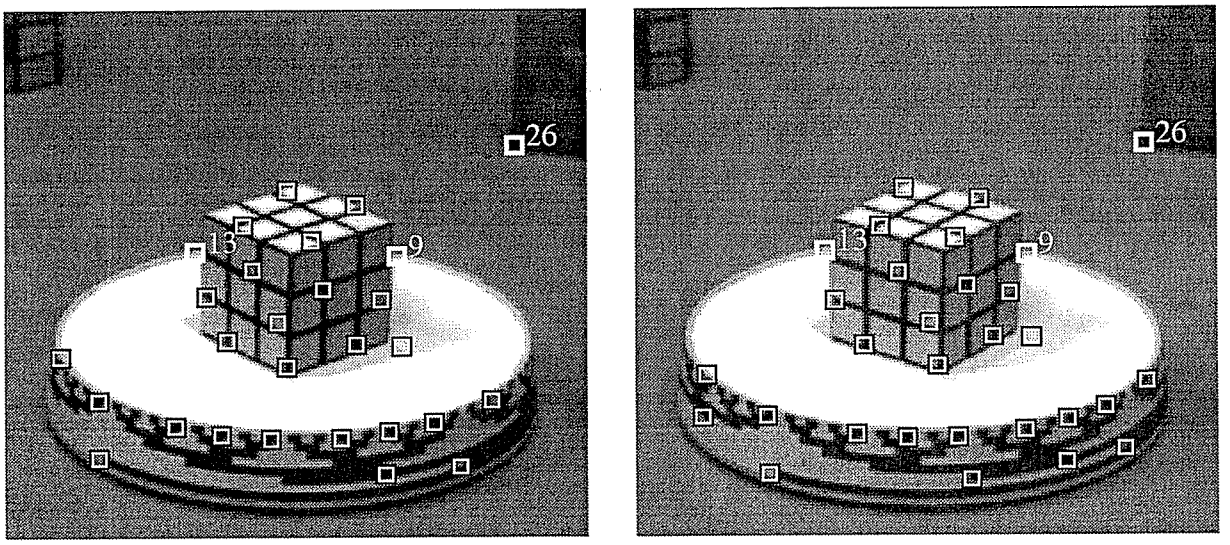
Figure 9.13: (a) The angular projection error and (b) the structure and motion error as a function of *outlier magnitude* for *3D perspective projection.*

---

the refined solution is effective for a few outliers that have a relatively small magnitude. However, increasing the frequency or magnitude of outliers makes them more difficult to detect, and those that remain undetected increasingly corrupt the solution.

9.2 Real Image Sequences

Five real image sequences were examined to show the performance of Projected Error Refinement on real image data. Features were tracked across the image sequences using the Kanada-Lucas-Tomasi feature tracker (see Chapter 6). A minimal subset of features and images from each sequence were selected by hand to obtain an initial estimate of the structure and motion parameters (see Chapter 4). The remaining features and images were then added to this solution (see Appendices A and B). This initial solution was then refined until the difference in the mean angular projection error between two successive solutions was less than 1% (see Chapter 5). Except for the teabox image sequence (Figure 9.16), the original positions of the feature points and camera centers was not known and therefore only a qualitative analysis of the recovered structure and motion is possible.



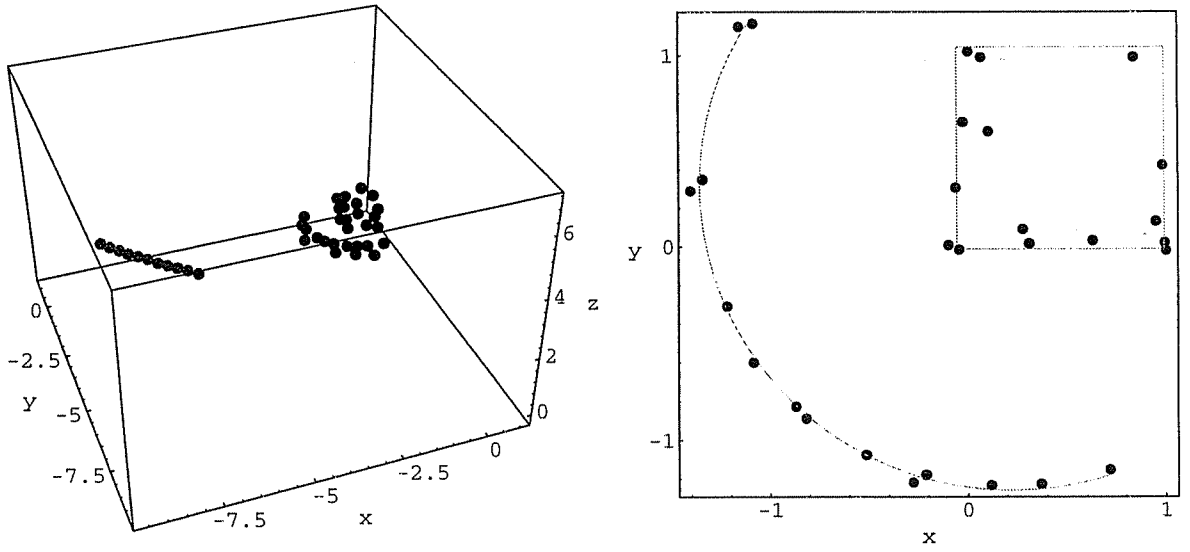
(a) (b)

Figure 9.14: (a) Frame 1 and (b) Frame 11 (of 11) of the Rubic's Cube image sequence.

9.2.1 Image Sequence 1: Rubic's Cube

This sequence contains 11 images of a Rubic's Cube rotating on a turntable. The camera was at a fixed position throughout the sequence. The first and last frames are shown in Figure 9.14. 30 feature points were tracked across the sequence with minimal occlusion, i.e., most of the features were visible in all the images. Feature point #26, shown in the upper-right corner of Figure 9.14, does not rotate with the turntable and therefore exhibits non-rigid relative motion. There are also two feature points, #9 and #13, that are caused by depth discontinuities. The reconstructed 3D positions of the feature points and camera centers are shown in Figure 9.15. After the first refinement stage, feature point #26 is detected as an outlier and its projectors are removed from the images and the solution is re-refined. The residual errors of features points #9 and #13 are less than 3σ and therefore they not identified as outliers.

As shown in Figure 9.15, the positions of the feature points and camera centers are recovered reasonably well. The reconstruction is imperfect largely as a result of the small change in viewpoint over the sequence. The projectors defined for each feature are therefore close to parallel, making the estimated positions of the feature points more sensitive to noise.



(a) (b)

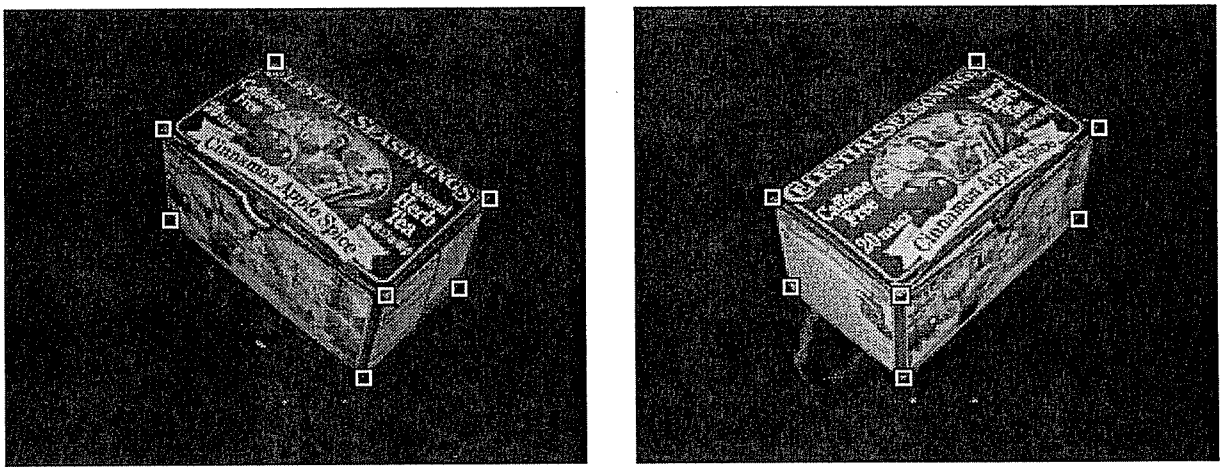
Figure 9.15: (a) The recovered positions of the feature points (black) and camera centers (gray) from the Rubic's Cube image sequence. (b) A top-down view of the recovered feature points (Note: the outlines of cube and turntable are not part of the reconstruction).

This experiment shows that Projected Error Refinement is able to recover a good estimate of scene structure and camera motion under favorable viewing conditions, although in this example the accuracy of the result it is limited by the small viewing angle.

9.2.2 Image Sequence 2: Teabox

The teabox image sequence illustrates how Projected Error Refinement is simplified when either scene structure or camera motion is known *a priori*. This sequence contains eight images of a teabox mounted on a pan-tilt head and rotated through 360° in 45° increments. Two of the images, at 45° and 135°, are shown in Figure 9.16. The eight corners of the box were manually labelled because the images were too widely spaced to permit automated feature tracking. The original structure of the teabox and the positions of the cameras are shown in Figure 9.17.

In the first experiment, the calibrated positions of the cameras were used to directly recover scene structure. The positions of the feature points were first estimated by triangula-



(a) (b)

Figure 9.16: Two frames of the calibrated teabox image sequence, at (a) 45° and (b) 315°.

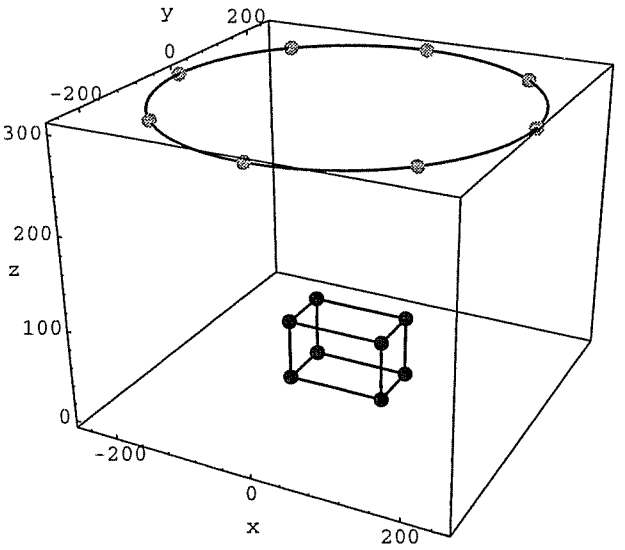


Figure 9.17: The calibrated positions of the cameras (gray) and feature points (black) for the teabox image sequence.

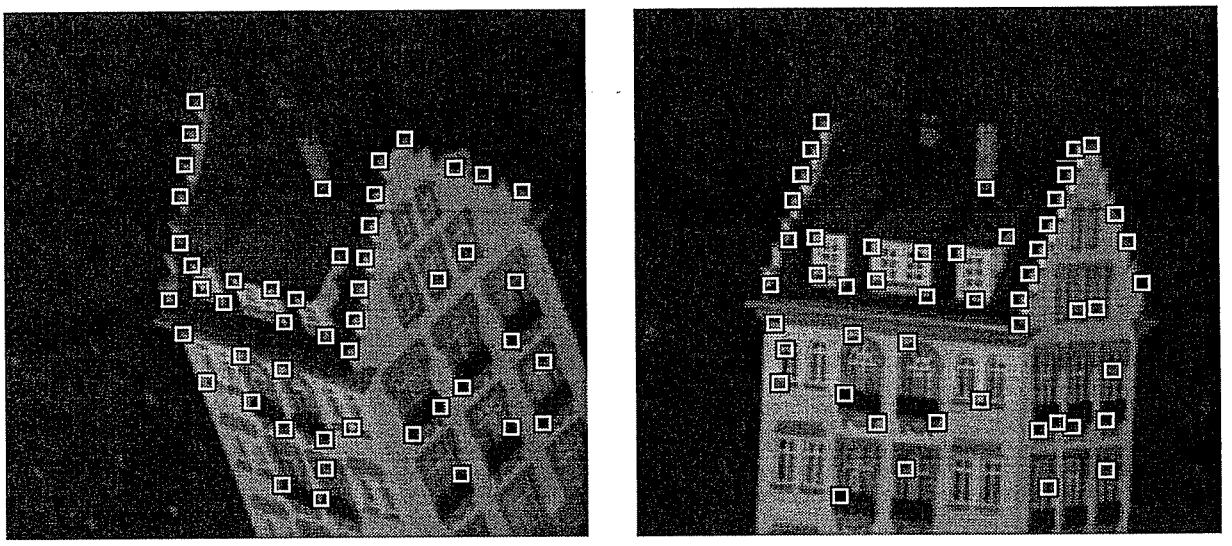
tion from a pair of images, and then refined using all the images. In other words, when the extrinsic camera parameters are known, the positions of the feature points can be determined in a single iteration. In this example, the average structure error in the positions of the recovered feature points is 3mm, where the dimensions of the teabox are 135mm × 80mm × 65mm, or 1.7% of the interior diagonal dimension of the box.

In the second experiment, the calibrated positions of the features were used to directly recover camera motion. First, the positions of the camera centers were estimated from a subset of the feature points. The extrinsic camera parameters were then refined, in parallel, with respect to all the feature points. In other words, when scene structure is known, the positions of the camera can be determined in a single iteration. In this example, the average motion error in the recovered camera positions is 28mm, or 7% of the distance from the camera to the rotational center of the pan-tilt head (395mm). The larger average motion error is most likely because four of the images, at 0°, 90°, 180° and 270°, have only six of the corners of the box visible, making their solutions more sensitive to noise in the images. This result could be improved by examining more feature points.

These experiments show how Projected Error Refinement can be adapted to the problem of direct scene reconstruction from calibrated images, or direct camera calibration from a known calibration target. In both cases, the relevant scene structure or camera motion can be solved in parallel in a single iteration.

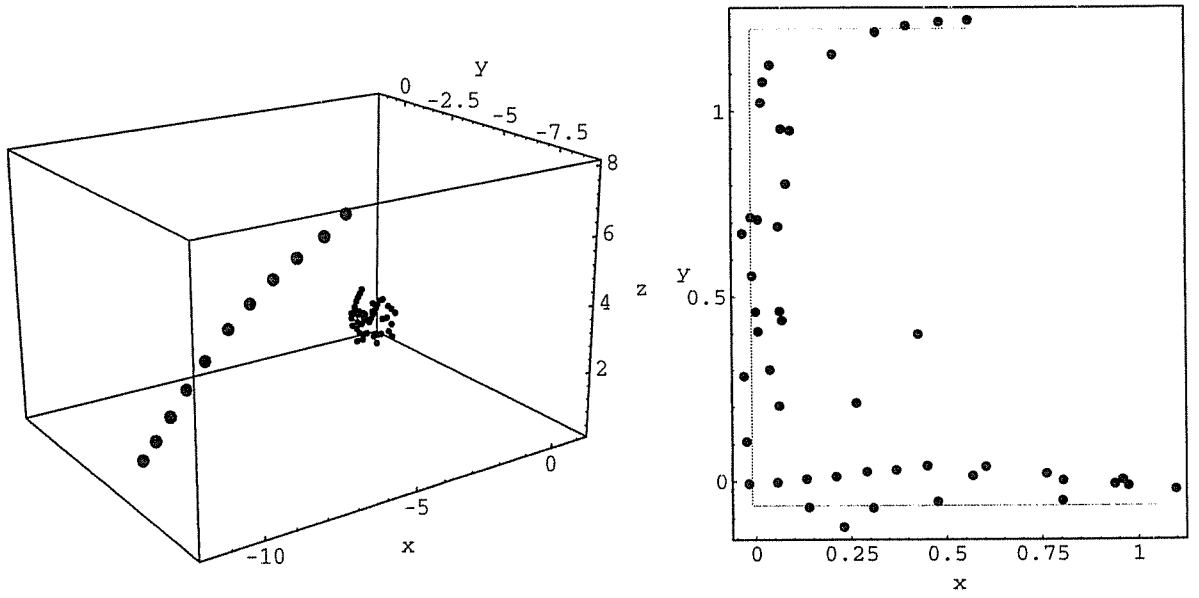
9.2.3 Image Sequence 3: Hotel

This a sequence of 11 images of a camera moving around a model of a hotel building. The first and last images are shown in Figure 9.18. Feature tracking was reliable because camera motion was smooth and 52 feature points were tracked over the sequence. There was minimal occlusion and no correspondence errors or false features. Despite the large change in viewpoint, the relative distance of the camera from the model remained constant. The projected size of the model does not change and the images have minimal perspective foreshortening effects, therefore this image sequence would be suitable for SFM techniques based on a parallel projection camera model [59], [46]. The reconstructed 3D positions of the feature points and camera centers obtained by Parallel Iterative Refinement (using perspective projection) are shown in Figure 9.19. Due to the larger change in viewpoint than in Figure 9.14, the observed projectors more strongly constrained the features' positions. This example shows, at



(a) (b)

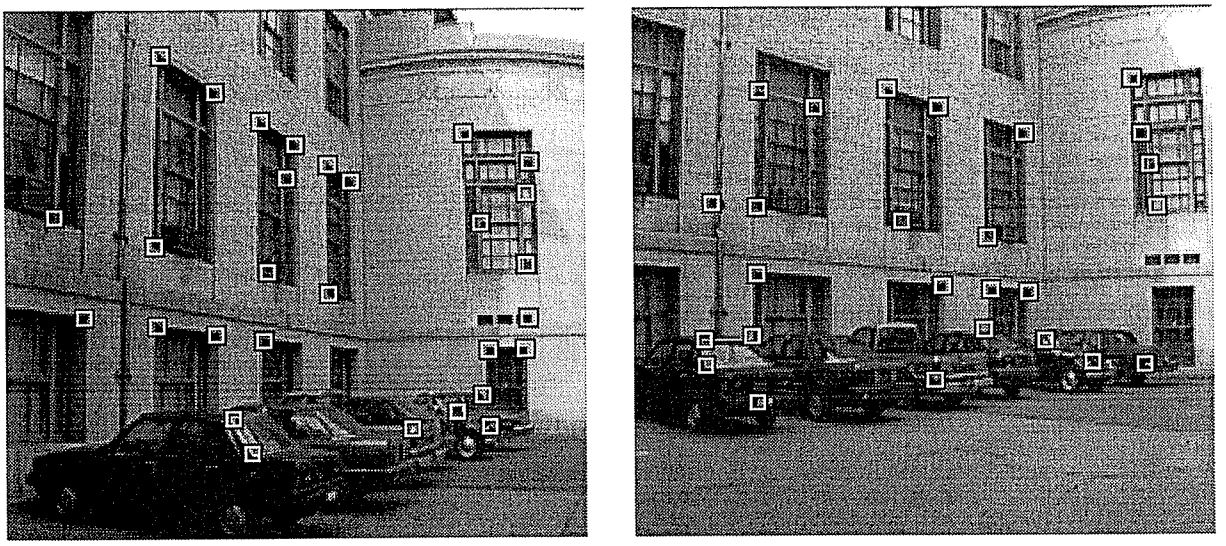
Figure 9.18: (a) Frame 1 and (b) Frame 11 (of 11) of the hotel image sequence.



(a) (b)

Figure 9.19: (a) The recovered positions of the feature points (black) and camera centers (gray) from the hotel image sequence. (b) A top-down view of the recovered feature points (Note: many of the original feature points are not vertically aligned with the sides of the hotel building).

least qualitatively, that Projected Error Refinement gives a better reconstruction of scene structure and camera motion when there is a larger change in viewpoint.



(a) (b)

Figure 9.20: (a) Frame 1 and (b) Frame 12 (of 12) of the building image sequence.

9.2.4 Image Sequence 4: Building

This is an outdoor sequence of 12 images of a building taken by a hand-held video camera. The first and last images are shown in Figure 9.20. Camera motion was erratic because the video camera was hand-held, however the features were sufficiently distributed and distinct so that no correspondence errors occurred during feature tracking. A total of 91 features were tracked over the sequence, however most were present only in a few frames; that is to say, there was substantial occlusion. Camera motion was away from the building and therefore the relative size of the scene changes over the sequence. Parallel projection is ill-suited to this image sequence because the change in the projected size of the scene cannot be modelled. It is also unclear whether weak perspective or para-perspective would work in this example either due to the noticeable perspective foreshortening of the left wall of the building. The reconstructed 3D positions of the feature points and camera centers by Parallel Iterative Refinement are shown in Figure 9.21. A few of the image features were caused by depth discontinuities, however these did not affect the solution because they were present only over a few frames. This experiment shows an application where perspective projection is required for accurate reconstruction because the image sequence contains perspective distortion, namely the size of

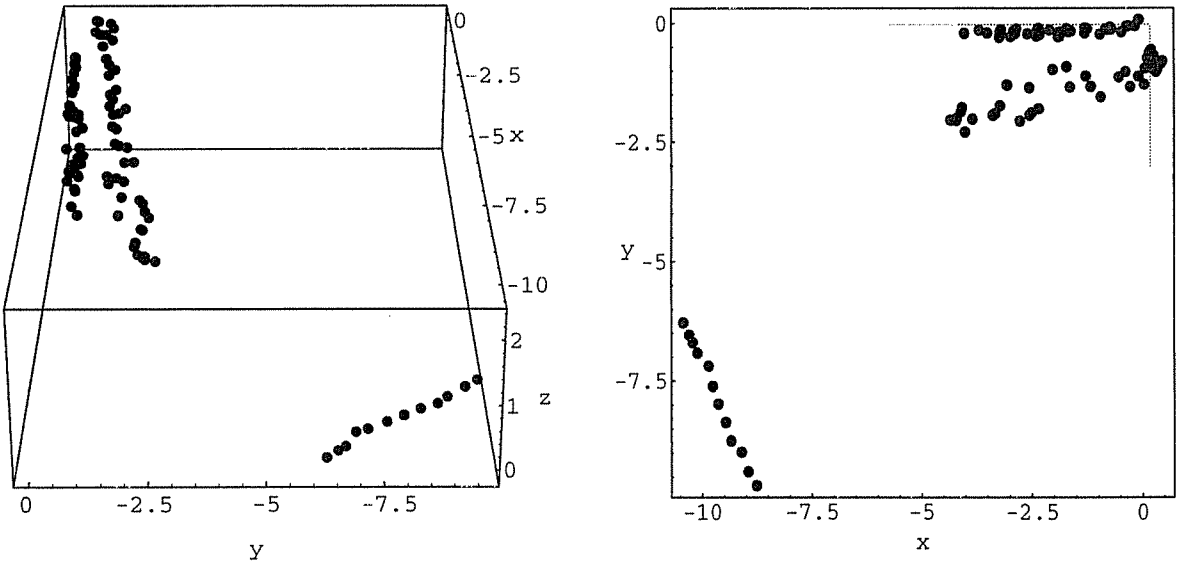
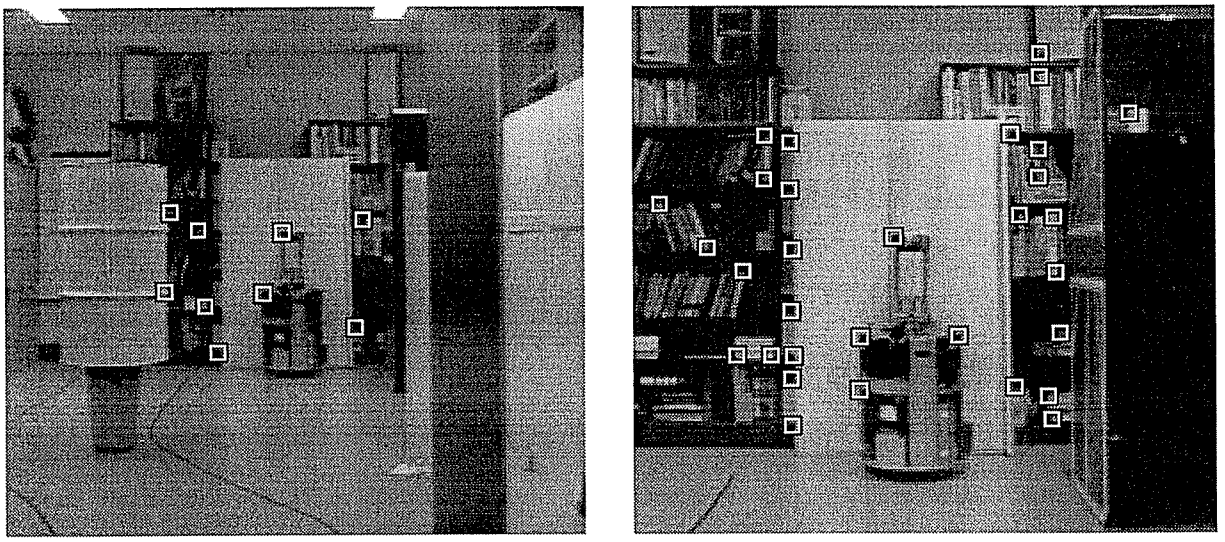


Figure 9.21: (a) The recovered positions of the feature points (black) and camera centers (gray) from the building image sequence. (b) A top-down view of the recovered scene.

scene changes as the camera moves and the shape of the scene is foreshortened in depth. Projected Error Refinement is able to accurately recover scene structure and camera motion in this example because it models perspective projection.

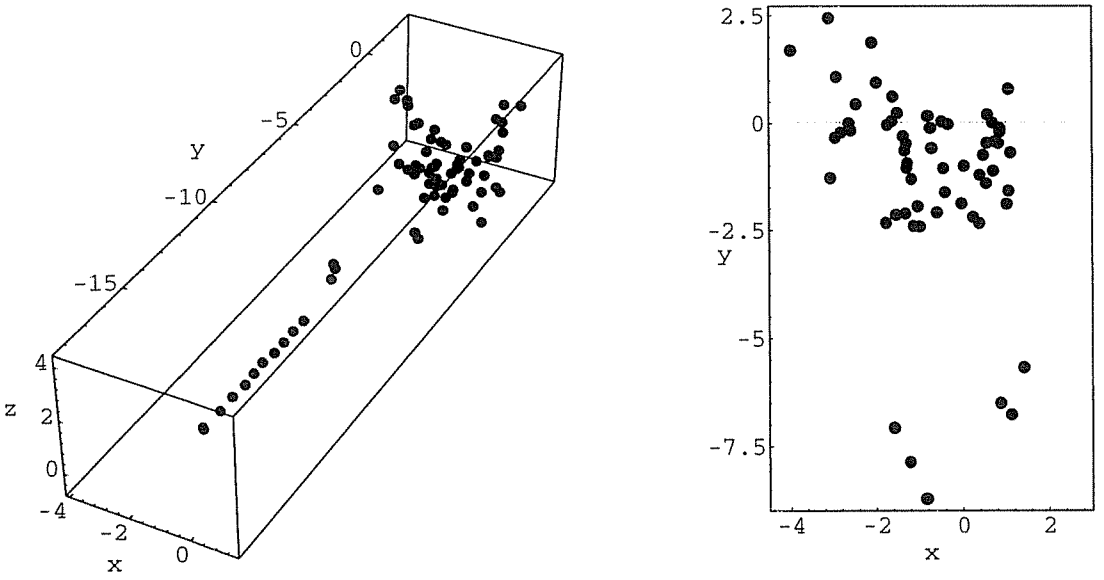
9.2.5 Image Sequence 5: Indoor Lab

The lab sequence contains 11 images of an indoor scene taken by a forward moving camera. The first and last images are shown in Figure 9.22. Because the camera moved forward, many features in the scene moved out of view and new features appeared on the rear wall as the camera moved closer. A total of 61 features were tracked in the sequence, with substantial occlusion. Camera motion was smooth and the feature points were tracked reliably. A few feature points were caused by depth discontinuities but they were only present in a few frames and did not affect the solution. This image sequence involves significant perspective effects and, as a result, absolutely requires a perspective projection camera model. Parallel approximations, such as weak perspective and para-perspective projection, are unsuitable for this application because the depth of the scene changes considerably. The reconstructed 3D posi-



(a) (b)

Figure 9.22: (a) Frame 1 and (b) Frame 11 (of 11) of the lab image sequence.



(a) (b)

Figure 9.23: (a) The recovered positions of the feature points (black) and camera centers (gray) from the lab image sequence. (b) A top-down view of the recovered feature points (Note: The rear wall is shown at y=0, therefore any feature points where y>0 are incorrect).

tions of the feature points and camera centers are shown in Figure 9.23.

This was the most a difficult image sequence to recover scene structure and camera

motion because the field of view of the camera was small and because many features were only visible over a narrow range of views. As a result, the projectors defined for these feature points are close to parallel, making their estimated positions especially sensitive to image noise. This is reflected by the poor reconstruction of many feature points, as shown in Figure 9.23(b). In this example there were few features detected on the periphery of the images, increasing the reliance on features close to the direction of motion. These features are more sensitive to noise because their projectors remain close to parallel. This experiment shows that for predominantly forward motion, it is important to detect features near the edges of the images for maximum angular variation in the observed projectors. It also shows that while Projected Error Refinement is able to recover approximate scene structure and camera motion under difficult viewing conditions, it is limited by the quality of the available projected image data.

9.3 Summary

Experimental results are important for measuring the accuracy of the recovered scene structure and camera motion under different viewing conditions. Unfortunately, the quantitative analysis and comparison of different SFM techniques is not widespread in this field. This is due in part to the lack of standardized, calibrated image sequences and also the variety of feature tracking algorithms that are employed. For these reasons, synthetic image data is predominantly used for quantitative error analysis.

The synthetic image experiments described in this chapter examined the effect of simulated viewing conditions on the projected image error and the resulting structure and motion errors of the solution. The first experiment confirmed that minimizing the angular projection error of the projectors has the desired effect of reducing the error in reconstruction. The second experiment showed that the projected error after refinement is closely related to the original image noise; in particular, the mean angular projection error is of approximately the same order as the angular image noise. This indicates that the angular projection error after refine-

ment can be used to estimate the image noise when it is not known *a priori*. The third experiment showed that reconstruction from the minimal number of features and images is poor and that *both* additional feature points and additional images are required for accurate reconstruction. This experiment also indicated that a good working set size is approximately 10 features projected to 5 images. The fourth experiment showed that while occlusion does affect the quality of reconstruction, if the visible projectors remain well distributed then scene structure and camera motion can still be accurately recovered. The last experiment showed that detecting and pruning outliers based on their residual error in the solution is effective and improves the accuracy of reconstruction. However, the ability to detect outliers is limited by the accuracy of the initial refined solution (with outliers included) and pruning outliers is less effective when the number of outliers is large or when outliers have a large magnitude.

The real image sequences demonstrated the performance of Projected Error Refinement in real applications. The first image sequence illustrated that scene structure and camera motion can be recovered accurately under favorable viewing conditions even when the change in viewpoint is small. The second experiment showed how Projected Error Refinement can be adapted to the problem of direct scene reconstruction or camera calibration when either the structure or motion parameters are known *a priori*, and that the desired motion or structure parameters can be found in a single iteration. The third image sequence illustrated recovering structure and motion for a moving camera and shows that, qualitatively at least, reconstruction is more accurate when there is a large change in viewpoint. The fourth image sequence illustrated reconstruction from an outdoor image sequence taken by a hand-held camera. These results showed that scene structure and camera motion can be accurately recovered despite significant occlusion, and that perspective effects do not pose a problem for Projected Error Refinement because it models perspective projection. This experiment also illustrates how Projected Error Refinement optimizes scene structure and camera motion based on *all* the observed feature points and images, unlike linear methods which must first extract a complete subset of features and images. The last image sequence showed that accurate reconstruction is

most difficult and least accurate when the camera motion is along the optical axis. This was compounded by the lack of features detected near the edges of the images. This experiment indicates that a large change in lateral viewpoint and/or wide field of view is required for accurate reconstruction, otherwise the projectors defined for each feature point are close to parallel and the solution is more sensitive to noise.

Chapter 10
Conclusions and Future Work

This thesis described a new non-linear optimization-based SFM technique called *Projected Error Refinement*. Research in SFM has primarily focussed on developing efficient linear methods and addressing the issue of noise. Efficiency is desirable, however it is not itself sufficient for general-purpose SFM reconstruction. Efficient non-linear optimization methods are better suited to this task because they are scalable and can examine all the available features and images. They are also recursive, which allows features and images to be added or removed arbitrarily, making occlusion natural and simplifying the handling of outliers.

10.1 Major Contributions

Projected Error Refinement fulfills many of the requirements of a general-purpose SFM technique. It is scalable, supports occlusion and detects outliers in a well-defined manner. Perspective projection is supported and no further constraints are imposed on the camera motion or scene structure other than rigidity. This technique minimizes the *angular projection error* of the visible projectors to obtain an optimal estimate of the structure and motion parameters. This approach is recursive and new images can be added at any time. Although other SFM techniques share some of these properties, Projected Error Refinement is unique in that *all* are supported using a *projector-based* camera model - a geometric model of inverse projection based on projectors rather than image coordinates.

Projected Error Refinement uses an efficient *parallel iterative refinement* algorithm that takes an initial estimate of the scene structure and camera motion and alternately refines

the structure and motion parameters. This reduces the complexity of non-linear optimization to (a) optimizing the camera's pose with respect to known features and (b) optimizing a feature's position with respect to known camera poses. Features and images are refined *in parallel* and projectors can be added or removed at any time and the refinement of the solution simply continues. Parallel iterative refinement also allows the precision of the solution to be determined by the available processing time, which is important for real-time applications.

The major contributions of this thesis can be summarized as

1. An intuitive geometric model of Structure From Motion based on projectors and angular projection error.

2. A new minimal data solution to the inverse projection problem for 2D and 3D perspective projection based on projectors.

3. An efficient non-linear optimization-based SFM technique for 2D and 3D perspective projection that is recursive, scalable and handles occlusion and outliers.

10.2 Future Research

There are several aspects of Projected Error Refinement that can be improved or extended to make the technique more useful and reliable.

10.2.1 Representation Using Projective Geometry

Projected Error Refinement performs non-linear optimization of the camera's poses and feature positions. The projection equations involved may have a simpler representation in *projective geometry.* In particular, the objective error function measures the angular projection error of each projector, which is the angle between two lines in Euclidean space. In projective geometry, this angle can be more simply described as the distance between two homogeneous

points. Any simplification of the objective error function will directly improve the efficiency of refining camera pose and feature position. Projective geometry also has the desirable property that perspective and parallel projection are treated alike. Projective geometry is presently used by several existing SFM techniques, although principally for the purpose of intrinsic camera calibration and projective reconstruction, rather than for non-linear optimization.

10.2.2 Improve Efficiency

Parallel iterative refinement simplifies the inverse projection problem by separately the refinement of feature position and camera pose. Presently, non-linear methods are used for both estimating the initial camera locations from four feature points, and later on for optimizing camera pose with respect to all the feature points. Huang and Faugeras [28] gave a linear solution to the location determination problem that examined additional feature points and lines. Alternative solutions to the location determination problem that simplify optimizing the camera locations should be investigated because this is computationally far more expensive than optimizing the feature positions. Efficiency may also be improved by using more sophisticated optimization methods with faster convergence properties. In particular, Levenberg-Marquardt optimization converges faster near the global minimum and has been successfully used by other optimization-based SFM techniques [55]. At present, Projected Error Refinement uses a simple gradient descent method [73].

10.2.3 Intrinsic Camera Calibration

Projected Error Refinement assumes the intrinsic camera parameters are known or obtained by other means. Several existing SFM techniques based on projective geometry recover the intrinsic camera parameters as well as scene structure and camera motion [34], [41]. Although independent calibration algorithms exist, it would be preferable to allow the intrinsic camera parameters to change dynamically over the image sequence, in particular the camera's focal length. For example, Pollefeys *et al.* [41] recently proposed a SFM technique based on projective geometry that allows the camera's focal length to vary in each image, as might occur for a

camera with a zoom lens.

Adding focal length as an variable parameter will require some modification to the projector model. Presently, only the *relative* directions of the projectors in an image are important and therefore the optical axis of the camera is ignored. However, if the focal length may vary, then the projectors become parameterized vectors measured relative to the optical axis (see Chapter 4.2). This also introduces an additional parameter to the optimization of camera pose, namely the focal length of the camera. It is not obvious what, if any, effect this parameter will have on the convergence properties of optimization. Formulating Projected Error Refinement in terms of projective geometry may facilitate adding focal length as an variable parameter. As described in Chapter 3, projectors are equivalent to homogeneous image coordinates used in projective geometry. Some of the results from projective geometry for intrinsic camera calibration may therefore be transferable to the projector model.

10.2.4 Improve Outlier Detection

Projected Error Refinement currently employs a simple heuristic for outlier detection based on the residual error of projectors after refinement. Experimental results showed this was not effective for large numbers of outliers or outliers with a large magnitude. In these situations a more sophisticated outlier detection mechanism is required; for example, *M-estimators* or *least median of squares* (LMedS). M-estimators is a relatively simple extension of least-squares minimization that adjusts the weight of each error term according to the magnitude of the residual error. LMedS is a more complex non-linear method that searches for the solution with the smallest *median* projected error. Both approaches to outlier detection are more reliable than pruning outliers based on their residual error, at the expense of increased complexity.

10.2.5 Extend to Long Image Sequences

Projected Error Refinement is recursive and new images can be added at any time without re-

computing the solution from scratch. However, long image sequences nonetheless will monotonically increase the complexity of optimization because all the previous features and images will continue to be refined each iteration. In long image sequences, the visible portion of the scene changes significantly. Therefore, rather than continuing to optimize *all* the feature points and images, it would be more efficient to only optimize the feature points and images currently of interest. Such a *dynamic* working set of current features and images can be obtained by adjusting the weight of projectors based on the 'age' of the images or other criteria. For example, the projectors in the most recent images are more important than those in earlier images and therefore should be given a larger weight. An open question is how to decide when projectors are no longer of interest and should be permanently removed from consideration.

Appendix A
Triangulation

Projected Error Refinement requires an initial estimate of the structure and motion parameters to begin optimization. This obtained from the minimal data solution described in Chapter 4, although any suitable SFM technique could equally be used. In most cases, the initial estimate will be computed from a subset of the features and images. Therefore, prior to refinement, the positions of the remaining feature points are estimated by triangulation from two of the recovered images. For each feature point, the intersection of any two of its projectors provides an estimate of its position. This position will be subsequently refined with respect to all the images.

In the case of 2D perspective projection, two projectors in $\Re^2$ will precisely intersect unless they are parallel. The point of intersection of two projectors, with direction vectors $\mathbf{v}_1$ and $\mathbf{v}_2$, for a feature point $\mathbf{p}$ that is projected to two images, with optical centers $\mathbf{o}_1$ and $\mathbf{o}_2$, is expressed by

$$\begin{aligned} \mathbf{p} &= \mathbf{o}_1 + \Omega_1 \mathbf{v}_1 \\ \mathbf{p} &= \mathbf{o}_2 + \Omega_2 \mathbf{v}_2. \end{aligned} \qquad \text{A.1}$$

Substituting $\mathbf{o}_j = \begin{bmatrix} x_j \\ y_j \end{bmatrix}$ and $\mathbf{v}_i = \begin{bmatrix} u_j \\ v_j \end{bmatrix}$ gives

$$\begin{bmatrix} x_1 \\ y_1 \end{bmatrix} + \Omega_1 \begin{bmatrix} u_1 \\ v_1 \end{bmatrix} = \begin{bmatrix} x_2 \\ y_2 \end{bmatrix} + \Omega_2 \begin{bmatrix} u_2 \\ v_2 \end{bmatrix}$$

$$\begin{bmatrix} \Omega_1 u_1 - \Omega_2 u_2 \\ \Omega_1 v_1 - \Omega_2 v_2 \end{bmatrix} = \begin{bmatrix} x_2 - x_1 \\ y_2 - y_1 \end{bmatrix} \qquad \text{A.2}$$

$$\begin{bmatrix} u_1 & -u_2 \\ v_1 & -v_2 \end{bmatrix} \begin{bmatrix} \Omega_1 \\ \Omega_2 \end{bmatrix} = \begin{bmatrix} x_2 - x_1 \\ y_2 - y_1 \end{bmatrix}$$

which is linear in Ω_1 and Ω_2. Solving for either Ω_1 or Ω_2 gives the position of $\mathbf{p}$ along the associated projector.

In the case of 3D perspective projection, two projectors in $\Re^3$ will rarely intersect. Therefore, the position of the feature point is estimated to be the midpoint of the shortest vector $\mathbf{w}_{min}$ between the two projectors. A vector $\mathbf{w}$ connecting a point on each line is given by

$$\begin{aligned} \mathbf{w} &= (\mathbf{o}_1 + \Omega_1 \mathbf{v}_1) - (\mathbf{o}_2 + \Omega_2 \mathbf{v}_2) \\ |\mathbf{w}|^2 &= (\mathbf{o}_1 - \mathbf{o}_2 + \Omega_1 \mathbf{v}_1 - \Omega_2 \mathbf{v}_2) \cdot (\mathbf{o}_1 - \mathbf{o}_2 + \Omega_1 \mathbf{v}_1 - \Omega_2 \mathbf{v}_2) \end{aligned} \qquad \text{A.3}$$

which is quadratic in Ω_1 and Ω_2. The minimum length vector $\mathbf{w}_{min}$ is found by solving for its partial derivatives set equal to 0, i.e.,

$$\frac{\partial}{\partial \Omega_1} |\mathbf{w}|^2 = 0, \quad \frac{\partial}{\partial \Omega_2} |\mathbf{w}|^2 = 0, \qquad \text{A.4}$$

which is a simple non-linear minimization problem. The midpoint of minimum length vector $\mathbf{w}_{min}$ connecting the two projectors gives the estimated position of $\mathbf{p}$; i.e.,

$$\mathbf{p} = (\mathbf{o}_1 + \Omega_1 \mathbf{v}_1) + \frac{\mathbf{w}_{min}}{2}. \qquad \text{A.5}$$

Appendix B
The Beacon Problem and Location Determination Problem

The initial estimate of the structure and motion parameters may not contain all the feature points and images. As described in Appendix A, the positions of the missing feature points are estimated by triangulation from the recovered images. Similarly, the positions and rotations of the missing images can be estimated from the recovered feature points. This is called the *beacon problem* for 2D perspective projection [24], or the *location determination problem* for 3D perspective projection [28], [15]. In both cases, if a set of known *beacons* (i.e., the recovered feature points) are observed from an unknown location, then the position of the observer can be computed from the relative directions of the beacons. In other words, the position of the camera's optical center can be estimated by examining the relative direction of a set of recovered feature points. The 2D beacon problem and the 3D location determination problem are defined differently and are therefore examined separately.

B.1 The Beacon Problem

In $\Re^2$ the observed angle $\alpha_{1,2}$ between two feature points $\mathbf{p}_1$ and $\mathbf{p}_2$, with unit direction vectors $\mathbf{v}_1$ and $\mathbf{v}_2$, respectively, is given by

$$\sin\alpha_{1,2} = |\mathbf{v}_1 \times \mathbf{v}_2|. \qquad \text{B.1}$$

The Law of Cosines states that the $\alpha_{1,2}$ defines a circle passing through $\mathbf{p}_1$, $\mathbf{p}_2$ and the optical center $\mathbf{o}$ of the image, as shown in Figure B.1. That is, at all points along the perimeter of

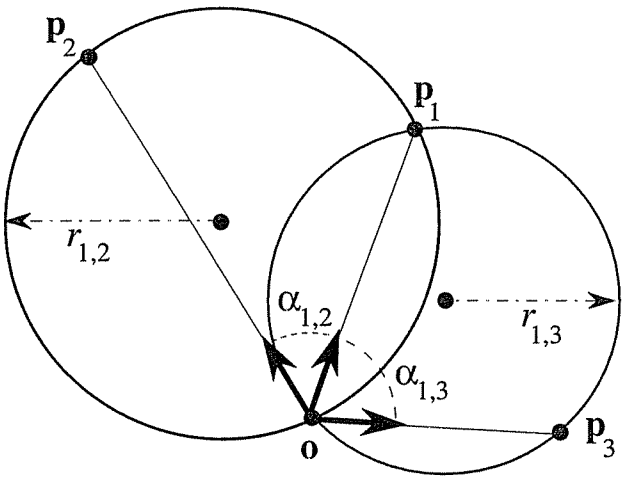


Figure B.1: The 2-D beacon problem. The observed angle $\alpha_{1,2}$ between two recovered feature points $\mathbf{p}_1$ and $\mathbf{p}_2$ defines a circle of radius $r_{1,2}$ on which the observer $\mathbf{o}$ must lie. A third feature point $\mathbf{p}_3$ defines another such circle. The intersection of these two circles gives the position of $\mathbf{o}$.

---

this circle the observed angle between $\mathbf{p}_1$ and $\mathbf{p}_2$ is the same. The radius of this circle is given by

$$r_{1,2} = \frac{|\mathbf{p}_1 - \mathbf{p}_2|}{\sin\alpha_{1,2}}. \qquad \text{B.2}$$

There are two opposing circles of radius $r_{1,2}$ passing through two points $\mathbf{p}_1$ and $\mathbf{p}_2$. The correct circle is determined by examining the sign of the cross-product $\mathbf{v}_1 \times \mathbf{v}_2$.

The observed angle between two feature points $\mathbf{p}_1$ and $\mathbf{p}_2$ defines a circle on which the observer $\mathbf{o}$ must lie. Examining a third feature point $\mathbf{p}_3$ defines a second such circle. As shown in Figure B.1, the intersection of these two circles gives the unique position of the camera's optical center. Once the location of the camera is found, any one of the feature points can be used to determine the appropriate rotation of the image around the optical center.

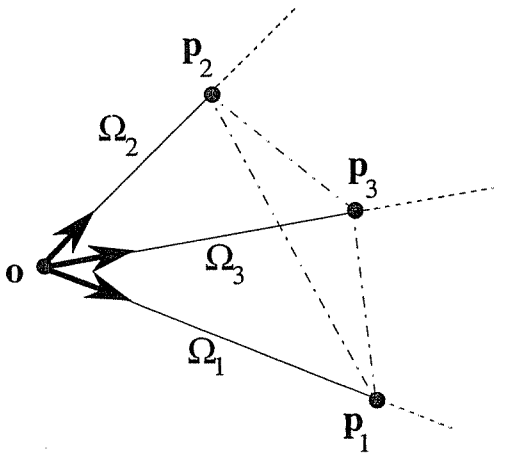


Figure B.2: The 3-D *location determination problem*. The locations of the three feature points $\overline{\mathbf{p}_1}, \overline{\mathbf{p}_2}, \overline{\mathbf{p}_3}$ along the three image projectors must match the distance between the same three points in the solution.

B.2 The Location Determination Problem

The 2D beacon problem has a 3D analogue, except that now the observed angle $\alpha_{1,2}$ between two feature points $\mathbf{p}_1$ and $\mathbf{p}_2$ defines a *surface* on which **o** must lie. In principle, the intersection of three such surfaces, derived from the three different pairings of feature points, $f(\mathbf{p}_1, \mathbf{p}_2)$, $f(\mathbf{p}_1, \mathbf{p}_3)$ and $f(\mathbf{p}_2, \mathbf{p}_3)$, will uniquely determine the location of **o**. However, this approach requires solving a complex 4th order polynomial. A simpler approach was described by Fischler and Bolles [15], called the *location determination problem*. This method is non-linear and examines three feature points, with a fourth point used to resolve ambiguity. Huang and Faugeras [28] also described a linear solution to the location determination problem that examined six feature points.

If three feature points $\mathbf{p}_1$, $\mathbf{p}_2$ and $\mathbf{p}_3$ are projected to an image **o**, with direction vectors $\mathbf{v}_1$, $\mathbf{v}_2$ and $\mathbf{v}_3$, then the position of each feature point along its respective projector must match the distance between the recovered points in the solution, as shown in Figure B.2 That is,

$$\begin{aligned}\mathbf{p}_1 &= \mathbf{o} + \Omega_1 \mathbf{v}_1 \\ \mathbf{p}_2 &= \mathbf{o} + \Omega_2 \mathbf{v}_2 \\ \mathbf{p}_3 &= \mathbf{o} + \Omega_3 \mathbf{v}_3,\end{aligned} \tag{B.3}$$

with the constraint that

$$\begin{aligned}|\mathbf{p}_1 - \mathbf{p}_2| &= d_{1,2} \\ |\mathbf{p}_1 - \mathbf{p}_3| &= d_{1,3} \\ |\mathbf{p}_2 - \mathbf{p}_3| &= d_{2,3}\end{aligned} \tag{B.4}$$

where d is the distance between the two points in the solution. Eq. B.4 is quadratic in three unknowns, $\Omega_1, \Omega_2, \Omega_3$, and has up to eight solutions [15]. However, for every positive solution there is a geometrically isomorphic negative solution behind the camera, so there are at most four actual solutions to consider. To resolve the ambiguity, the procedure is repeated substituting a fourth point $\mathbf{p}_4$ in place of $\mathbf{p}_3$ to identify the unique solution.

Eq. B.3 gives the relative distances Ω_i of each feature point $\mathbf{p}_i$ from the focal point $\mathbf{o}$. The actual position of $\mathbf{o}$ with respect to the feature points is found by solving

$$\begin{aligned}|\mathbf{o} - \mathbf{p}_1| &= \Omega_1 \\ |\mathbf{o} - \mathbf{p}_2| &= \Omega_2 \\ |\mathbf{o} - \mathbf{p}_3| &= \Omega_3\end{aligned} \tag{B.5}$$

which has two solutions, one on each side of the triangle with vertices $\mathbf{p}_1$, $\mathbf{p}_2$ and $\mathbf{p}_3$. The correct position of $\mathbf{o}$ is determined by examining the triple scalar product $(\mathbf{v}_1 \times \mathbf{v}_2) \cdot \mathbf{v}_3$.

Appendix C
Measuring Structure and Motion Error

The recovered positions of the feature points and the cameras' optical centers are accurate only up to a scale factor and a rigid translation and rotation because the original scene's coordinate system is unknown. Measuring the accuracy of the solution is therefore difficult because determining the optimal coordinate transformation mapping the solution to the original scene is a non-trivial *data fitting* or *shape registration* problem [1], [22]. For example, fitting two sets of data points in $\Re^3$ is a non-linear optimization problem involving seven independent parameters.

In order to simplify mapping the solution to the original coordinate system, the synthetic images described in this thesis were generated from scenes where the feature points lie on a *unit circle* for 2D perspective projection, or a *unit sphere* for 3D perspective projection, as shown in Figure 9.1. For example, Figure C.1 shows a synthetic scene in $\Re^2$ containing eight feature points projected to five images. Because the original feature points lie on a circle, so should the recovered feature points. Therefore, a circle is fitted through the features points in the solution, which is defined as

$$(x_i - \mathbf{c}_x)^2 + (y_i - \mathbf{c}_y)^2 = r^2, \qquad \text{C.1}$$

where $\mathbf{c} = \{\mathbf{c}_x, \mathbf{c}_y\}$ is the center of the circle and r is the radius. Similarly, for 3D perspective projection, the recovered feature points should lie on a sphere, which is defined as

$$(x_i - \mathbf{c}_x)^2 + (y_i - \mathbf{c}_y)^2 + (z_i - \mathbf{c}_z)^2 = r^2, \qquad \text{C.2}$$

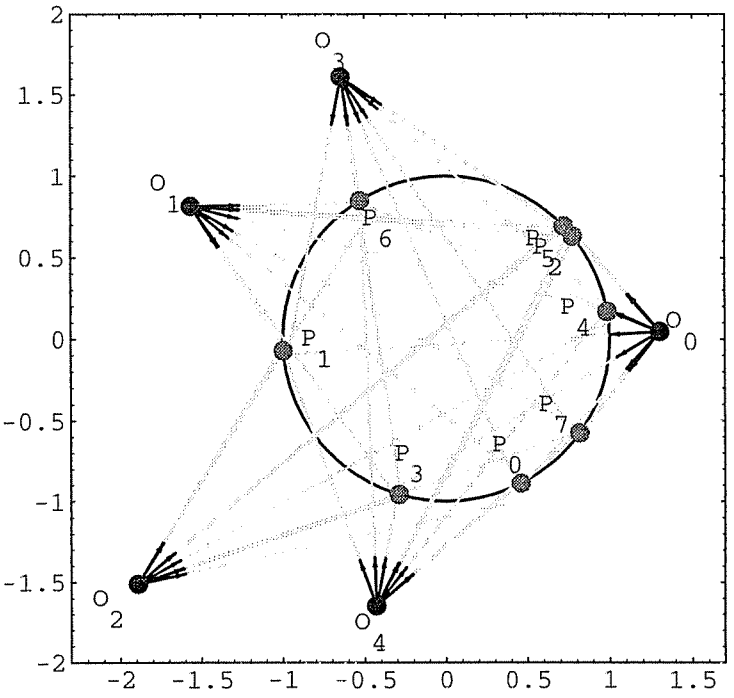


Figure C.1: Example of a synthetic 2D scene with eight feature points $\mathbf{p}_i$ lying on a circle that are projected to five images with optical centers $\mathbf{o}_j$. 1° angular noise has been added to the projectors in the images.

---

where $\mathbf{c} = \{\mathbf{c}_x, \mathbf{c}_y, \mathbf{c}_z\}$ is the center of the sphere. A circle or sphere is fitted to the feature points in the solution to minimizes the least-squares distance between the points and the perimeter of the circle or sphere. The corresponding objective error function is

$$\varepsilon = \sum_i \left[\sqrt{(x_i - \mathbf{c}_x)^2 + (y_i - \mathbf{c}_y)^2} - r^2 \right] \qquad \text{C.3}$$

for the case of a circle, or

$$\varepsilon = \sum_i \left[\sqrt{(x_i - \mathbf{c}_x)^2 + (y_i - \mathbf{c}_y)^2 + (z_i - \mathbf{c}_z)^2} - r^2 \right] \qquad \text{C.4}$$

for the case of a sphere. The radius r is initially estimated to be 1, and the center-of-mass $\mathbf{m}$ of the feature points provides the initial estimate of $\mathbf{c}$ and is given by

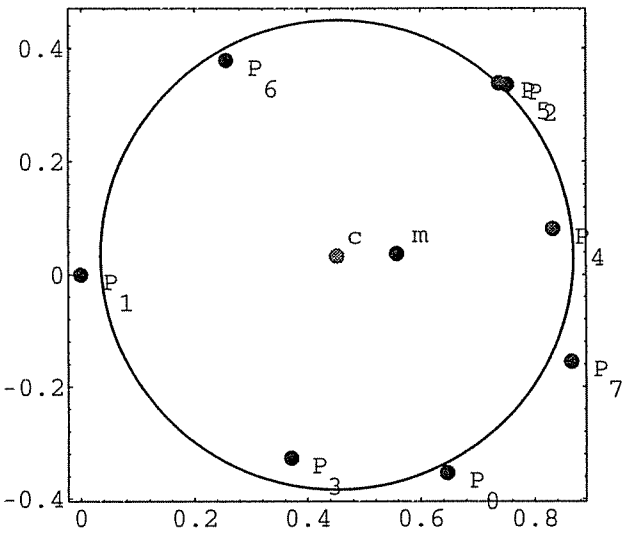


Figure C.2: A circle fitted to the feature points in the initial (unrefined) solution to the images projected in Figure C.1. The center of the circle **c** gives the translation vector from the solution to the original scene's coordinate system, and the radius of the circle gives the scale change. **m** is the center-of-mass of the feature points.

$$\mathbf{m} = \frac{\sum_{i=0}^{I-1} \mathbf{p}_i}{I}, \qquad \text{C.5}$$

where I is the number of feature points. Figure C.2 shows the circle fitted to the feature points in the solution to the images projected in Figure C.1. This circle gives the translation and scale factor mapping the solution to the original scene, leaving only the rotation transformation to be determined.

The optimal rotation that aligns the translated and scaled solution with the original scene minimizes the least-squares distances ε_i between the feature points $\mathbf{p}_i$ in the solution and their positions $\tilde{\mathbf{p}}_i$ in the original scene. In $\Re^2$ this distance is given by

$$\varepsilon_i = \left| \tilde{\mathbf{p}}_i - \mathbf{p}_i \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \right|, \qquad \text{C.6}$$

where θ specifies the rotation of the circle around the origin. In $\Re^3$ this distance is given by

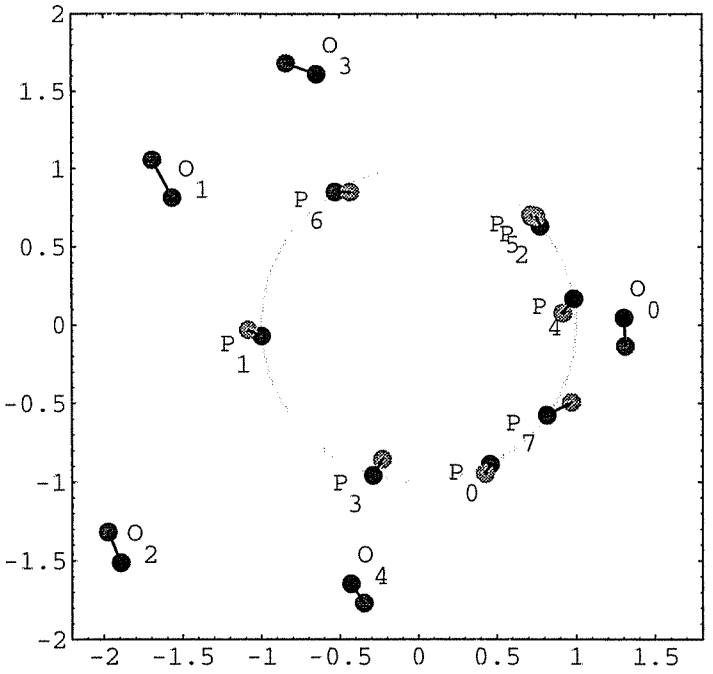


Figure C.3: The feature points in the solution (gray) are finally rotated to find the best fit with the points in the original scene (black). The distance between the recovered and original feature points is called the *structure error*, and the distance between the recovered and original optical centers is called the *motion error*.

$$\varepsilon_i = \left| \tilde{\mathbf{p}}_i - \mathbf{p}_i \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\delta & \sin\delta \\ 0 & -\sin\delta & \cos\delta \end{bmatrix} \cdot \begin{bmatrix} \cos\varphi & 0 & \sin\varphi \\ 0 & 1 & 0 \\ -\sin\varphi & 0 & \cos\varphi \end{bmatrix} \cdot \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \right|, \qquad \text{C.7}$$

where $\{\theta, \varphi, \delta\}$ specify the rotation of the sphere around the origin. Eqn. C.6 and Eqn. C.7 are solved to find the optimal rotation parameters. Figure C.3 shows the final positions of the feature points in the solution after scaling, translation and rotation to the original coordinate system.

The average distance between the feature points in the final solution and their positions in the original scene is called the *structure error* ε_S and is defined as

$$\varepsilon_S = \frac{\sum_{i=0}^{I-1} \varepsilon_i}{I}, \qquad \text{C.8}$$

where I is the number of features. The average distance between the cameras' optical centers in the final solution and their positions in the original scene is called the *motion error* ε_M and is defined as

$$\varepsilon_M = \frac{\sum_{j=0}^{J-1} \varepsilon_j}{J}, \qquad \text{C.9}$$

where J is the number of images. In this example, the structure error is 0.09 and the motion error is 0.20. Both errors are unit-less distances measured relative to the circle or sphere having radius 1 on which the original feature points lie.

An important consequence of this method for measuring the structure and motion error is that the cameras' optical centers are ignored when determining the optimal registration transformation. That is, registration is based on an optimal fit of the feature points only. As a result, the measured motion error, i.e., the error in the recovered cameras' optical centers, is typically larger than the structure error. This is a by-product of the registration method and is not because camera motion is less reliably recovered by Projected Error Refinement than scene structure, and a different technique for fitting the solution to the original scene will give a different structure and motion error [1], [22].

Bibliography

[1] K.S. Arun, T.S. Huang and S.D. Blostein, "Least-squares fitting of two 3-D point sets," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 9, no. 5, 1987, pp. 698-700.

[2] A. Azarbayejani and A.P. Pentland, "Recursive estimation of motion, structure and focal length," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 17, no. 6, 1995, pp. 562-575.

[3] J.M. Babcock and R.A. Jarvis, "Wire-frame modelling of polyhedral objects from rangefinder data," *Robotica*, vol. 12. 1994, pp. 65-75.

[4] S. Birchfield, "Derivation of Kanade-Lucas-Tomasi tracking equation," *unpublished notes*, 1997.

[5] T.E. Boult and L.G. Brown, "Factorization-based segmentation of motions," *Proc. IEEE Workshop on Visual Motion*, 1991, pp. 179-186.

[6] F. Chaumette, S. Boukir, P. Bouthemy and D. Juvin, "Structure from controlled motion," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, no. 5, 1996, pp. 492-504.

[7] H.H. Chen and T.S. Huang, "Matching 3-D line segments with applications to multiple-object motion estimation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, no. 10, 1990, pp. 1002-1008.

[8] G.T. Chou and S. Teller, "Multi-image correspondence using geometric and structural constraints," *Proc. Image Understanding Workshop*, vol. 2, 1997, pp. 869-874.

[9] S. Christy and R. Horaud, "Euclidean shape and motion from multiple perspective view by affine iterations," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, no. 11, 1996, pp. 1098-1104.

[10] R. Cipolla, Y. Okamoto and Y. Kuno, "Robust structure from motion using motion parallax," *Proc. International Conference in Computer Vision*, 1993, pp. 374-382.

[11] S. Coorg and S. Teller, "Matching and pose refinement with camera pose estimates," *Proc. Image Understanding Workshop*, vol. 2, 1997, pp. 857-862.

[12] U.R. Dhond and J.K. Aggarwal, "Structure from stereo - a review," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 19, no. 6, 1989, pp. 1489-1510.

[13] O. Faugeras, *Three-Dimensional Computer Vision: A Geometric Viewpoint*, MIT Press, Cambridge, Mass., 1993.

[14] O.D. Faugeras and S. Maybank, "Motion from point matches: Multiplicity of solutions," *Proc. IEEE Workshop on Visual Motion*, 1989, pp. 248-255.

[15] 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*, vol. 24, no. 6, 1981, pp. 381-395.

[16] J.J. Gibson, *The Ecological Approach to Visual Perception*, Houghton Mifflin Company, Boston, 1979.

[17] R.I. Hartley, "Projective reconstruction and invariants from multiple images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 16, no. 10, 1994, pp. 1036-1041.

[18] R.I. Hartley, "Self-calibration from multiple views with a rotating camera," *Proc. Third European Conference on Computer Vision*, vol. 1, 1994, pp. 471-478.

[19] R.I. Hartley, "Kruppa's equations derived from the fundamental matrix," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 2, 1997, pp. 133-135.

[20] R.J. Holt and A.N. Netravali, "Uniqueness of solutions to three perspective views of four points," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 17, no. 3, 1995, pp. 303-307.

[21] B.K.P. Horn, "Motion fields are hardly ever ambiguous," *International Journal of Computer Vision*, vol. 1, 1987, pp. 259-274.

[22] K. Kanatani, "Analysis of 3-D rotation fitting," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 16, no. 5, 1994, pp. 543-549.

[23] J.J. Koenderink and A.J. Van Doorn, "Invariant properties of the motion parallax field due to the movement of rigid bodies relative to an observer,"

Optica Acta, vol. 22, no. 9, 1975, pp. 773-791.

[24] E. Krotkov, "Mobile robot localization using a single image," *Proc. IEEE International Conference on Robotics and Automation*, 1989, pp. 978-983.

[25] K.N. Kutulakos and C.R. Dyer, "Global surface reconstruction by purposive control of observer motion," *Artificial Intelligence*, vol. 78, 1995, pp. 147-177.

[26] R. Laganière and A. Mitiche, "On combining points and lines in an image sequence to recover 3D structure and motion," *Proc. IEEE Workshop on Visual Motion*, 1989, pp. 221-228.

[27] J.-C. Latombe, *Robot Motion Planning*, Kluwer Academic Publishers, Boston, 1991.

[28] Y. Liu, T.S. Huang and O.D. Faugeras, "Determination of camera location from 2-D to 3-D line and point correspondences," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, no. 1, 1990, pp. 28-37.

[29] H.C. Longuet-Higgins, "A computer algorithm for reconstructing a scene from two projections," *Nature*, vol. 293, 1981, pp. 133-135.

[30] S.J. Maybank and O.D. Faugeras, "A theory of self-calibration of a moving camera," *International Journal of Computer Vision*, vol. 8, 1992, pp. 123-152.

[31] S. Maybank, *Theory of Reconstruction from Image Motion*, Springer Series in Information Sciences, vol. 28, Springer-Verlag, New York, NY, 1993.

[32] P.F. McLauchlan and D.W. Murray, "Active camera calibration for a head-eye platform using the variable state-dimension filter," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, no. 1, 1996, pp. 15-22.

[33] D.P. McReynolds and D.G. Lowe, "Rigidity checking of 3D point correspondences under perspective projection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, no. 12, 1996, pp. 1174-1185.

[34] T. Moons, L. Van Gool, M. Proesmans and E. Pauwels, "Affine reconstruction from perspective images pairs with a relative object-camera translation in between," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, no. 1, 1996, pp. 77-83.

[35] J.L. Mundy and A. Zisserman, *Geometric Invariance in Computer Vision*, MIT Press, Boston, Mass., 1992.

[36] A. Naeve and J.-O. Eklundh, "On projective geometry and the recovery of 3-D structure," *Technical Report*, CAR-TR-154, Center for Automation Research, University of Maryland, 1985.

[37] S.K. Nayar and Y. Nakagawa, "Shape from focus," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 16, no. 8, 1994, pp. 824-831.

[38] S. Negahdaripour, "Multiple interpretations of the shape and motion of objects from two perspective images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, no. 11, 1990, pp. 1025-1039.

[39] J. Oliensis, "A critique of structure from motion algorithms," *Technical Report*, NEC Research Institute, 1997.

[40] J. Philip, "Estimation of three-dimensional motion of rigid objects from noisy observations," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, no. 1, 1991, pp. 61-66.

[41] M. Pollefeys, R. Koch and L. Van Gool, "Self-Calibration and metric reconstruction in spite of varying and unknown internal camera parameters," *Proc. Sixth International Conference on Computer Vision*, 1998, pp. 90-95.

[42] C.J. Poelman and T. Kanade, "A paraperspective factorization method for shape and motion recovery," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 3, 1997, pp. 206-218.

[43] W.H. Press, S.A. Teukolsky, W.T. Vetterling and B.P. Flannery, *Numerical Recipes in C*, Cambridge University Press, Cambridge, Mass., 1992.

[44] L. Quan, "Invariants of six points and projective reconstruction from three uncalibrated images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 17, no. 1, 1995, pp. 34-46.

[45] H.S. Sawhney, J. Oliensis and A.R. Hanson, "Image description and 3-D reconstruction from image trajectories of rotational motion," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 15, no. 9, 1993, pp. 885-898.

[46] S.M. Seitz and C.R. Dyer, "Complete scene structure from four point correspondences," *Proc. International Conference on Computer Vision*, 1995,

pp. 330-337.

[47] S.M. Seitz and C.R. Dyer, "Photorealistic scene reconstruction by voxel coloring," *Proc. Computer Vision and Pattern Recognition*, 1997, pp. 1067-1073.

[48] L.S. Shapiro, "Affine analysis of image sequences," *Ph.D. Thesis*, Department of Engineering Science, University of Oxford, 1993.

[49] H. Shariat and K.E. Price, "Motion estimation with more than two frames," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, no. 5, 1990, pp. 417-434.

[50] A. Shashua, "Projective structure from two uncalibrated images: Structure from motion and recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 16, no. 8, 1994, pp. 778-790.

[51] A. Shashua and N. Navab, "Relative affine structure: Canonical model for 3D from 2D geometry and applications," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, no. 9, 1996, pp. 873-883.

[52] J. Shi and C. Tomasi, "Good features to track," *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 1994. pp. 593-600.

[53] M. Spetsakis, "Optimal motion estimation," *Proc. IEEE Workshop on Visual Motion*, 1989, pp. 229-237.

[54] R. Szeliski and S.B. Kang, "Shape ambiguities in structure from motion," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 5, 1997, pp. 506-512.

[55] R. Szeliski and S.B. Kang, "Recovering 3D shape and motion from image streams using non-linear least squares," *Journal of Visual Communication and Image Representation*, vol. 5, no. 1, 1994, pp. 10-28.

[56] C. Taylor, D. Kriegman and P. Anandan, "Structure and motion from multiple images: A least-squares approach," *Proc. IEEE Workshop on Visual Motion*, 1991, pp. 242-248.

[57] E.H. Thompson, "A rational algebraic formulation of the problem of relative orientation," *Photogrammetric Record*, vol. 3, no. 14, 1959, pp. 152-159.

[58] W.B. Thompson, P. Lechleider and E.R. Stuck, "Detecting moving objects using the rigidity constraint," *IEEE Transactions on Pattern Analysis and*

Machine Intelligence, vol. 15, no. 2, 1993, pp. 162-166.

[59] C. Tomasi, "Shape and motion from image streams: A factorization method," *Ph.D. Thesis*, CMU-CS-91-172, Carnegie Mellon University, 1991.

[60] C. Tomasi and T. Kanade, "Factoring image sequences into shape and motion," *Proc. IEEE Workshop on Visual Motion*, 1991, pp. 21-28.

[61] C. Tomasi, "Input redundancy and output observability in the analysis of visual motion," *Proc. Sixth Symposium on Robotics Research*, 1993, pp. 213-222.

[62] C. Tomasi and J. Zhang, "Is structure-from-motion worth pursuing?" *Proc. Seventh International Symposium on Robotics Research*, 1995, pp. 391-400.

[63] P.H.S. Torr and D.W. Murray, "The development and comparison of robust methods for estimating the fundamental matrix," *International Journal of Computer Vision*, vol. 24, no. 3, 1997, pp. 271-300.

[64] R.Y. Tsai and T.S. Huang, "Uniqueness and estimation of three-dimensional motion parameters of rigid objects with curved surfaces," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 6, no. 1, 1984, pp. 13-27.

[65] R.Y. Tsai, "A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses," *IEEE Journal of Robotics and Automation*, vol. 3, no. 4, 1987, pp. 323-344.

[66] S. Ullman, *The Interpretation of Visual Motion*, MIT Press, Cambridge, Mass., 1979.

[67] S. Ullman, "Computational studies on the interpretation of structure and motion: Summary and extension," *Artificial Intelligence Memo*, Artificial Intelligence Laboratory, MIT, no. 706, 1983.

[68] S. Ullman, "Maximizing rigidity: The incremental recovery of 3-D structure from rigid and non-rigid motion," *Perception*, vol. 13, 1984, pp. 255-274.

[69] Y.F. Wang, "Characterizing three-dimensional surface structures from visual images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, no. 1, 1991, pp. 52-60.

[70] D. Weinshall and C. Tomasi, "Linear and incremental acquisition of invariant shape models from image sequences," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 17, no. 5, 1995, pp. 512-517.

[71] J. Weng, T.S. Huang, and N. Ahuja, "Motion and structure from line correspondences: Closed-form solution, uniqueness, and optimization," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 3, 1992, pp. 318-336.

[72] J. Weng, N. Ahuja and T.S. Huang, "Optimal motion and structure estimation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 15, no. 9, 1993, pp. 864-884.

[73] S. Wolfram, *Mathematica: A System for Doing Mathematics by Computer*, 2nd ed., Addison-Wesley, Redwood, CA, 1991.

[74] B.L. Yen and T.S. Huang, "Determining 3-D motion and structure of a rigid body using the spherical projection," *Computer Vision, Graphics, and Image Processing*, vol. 21, 1983, pp. 21-32.

[75] Z. Zhang and O. Faugeras, *3D Dynamic Scene Analysis: A Stereo Based Approach*, Springer, Berlin, Heidelberg, 1992.

[76] 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," *Artificial Intelligence Journal*, vol. 78, 1995, pp. 87-119.