Maximum Likelihood Methods for Hierarchical Structure from Motion of Uncalibrated Video

by
Stuart Benjamin Heinrich

A dissertation submitted to the Graduate Faculty of
North Carolina State University
in partial fulfillment of the
requirements for the Degree of
Doctor of Philosophy

Computer Science

Raleigh, North Carolina

2011

APPROVED BY:

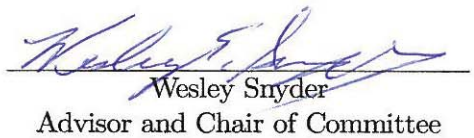
Jan-Michael Frahm
External Co-Advisor

Ben Watson

Chris Healey

Siamak Khorram

Griff Bilbro



Wesley Snyder
Advisor and Chair of Committee

ABSTRACT

HEINRICH, STUART BENJAMIN. Maximum Likelihood Methods for Hierarchical Structure from Motion of Uncalibrated Video. (Under the direction of Wesley Snyder.)

This dissertation addresses the general problem of reconstructing static 3D scene structure and camera parameters from an uncalibrated video or image series using structure from motion. Using these techniques, reconstructions can be made from video or still images taken with any hand held camera. We make no prior assumptions about the camera except that it takes images that are not stretched or skewed, which is true for all modern cameras, and require no additional information beyond the images. The advancements proposed in this dissertation include a novel overall system architecture that is designed to maximize robustness to noise and outliers, analysis of the theoretical instabilities of traditional methods of autocalibration along with a maximum likelihood approach that overcomes these limitations, a maximum likelihood method of merging projective reconstructions that is invariant to the uncertainty in structure points, a parallel decomposition for bundle adjustment, a derivation of new internal constraints of the trifocal tensor, and several other contributions. Many of these advancements are useful not only to uncalibrated video reconstruction but also reconstructions from arbitrary photo collections as well as visual odometry systems for robot navigation.

BIOGRAPHY

Stuart B. Heinrich was born November 13, 1983 in Burlington, Vermont. He attended Champlain Valley Union High School in Hinesburg, Vermont, where he ran cross country on the varsity team and also participated in track and field. He was a Vermont Scholar at the University of Vermont where he received his Bachelor of Science in Computer Science with a minor in Electrical Engineering. He made the Dean's List all four years, and was the keynote speaker at his own graduation after winning second place in the Volney Giles Barbour Engineering Essay competition. He also designed the logo for the UVM Complex Systems Center.

He has participated in research internships at the VT Healthcare and Information Technology Education Center (HITEC), UVM Extension, NSF Research Experience for Undergraduates (REU) at Texas A&M University, and the Summer Research Program (SRP) at Argonne National Laboratory (ANL).

He was accepted to the doctoral program of computer science at North Carolina State University in 2006 where he completed his coursework with a 4.0 GPA and received his masters degree in computer science. While pursuing his PhD studies he was a computer science instructor for the Research Apprenticeship Program for High School Students at Shaw University for 2 years, and also made his theatrical debut in the performances of 'Amadeus', 'It's a Wonderful Life: A Live Radio Broadcast', and 'Macbeth' at the University Theater. He holds a black belt in Tae Kwon Do and has studied numerous other martial arts including Aiki Jitsu, Brazilian Jiu Jitsu, mantis style Kung Fu, Bagua and Wing Chun Kung Fu. His other interests include a lifelong passion for art and a fondness for fiction writing.

ACKNOWLEDGEMENTS

I would, first of all, like to thank my primary advisor Dr. Wesley Snyder, not only for giving me this opportunity but also allowing me the freedom to explore completely new directions, and sticking with me through the thick and thin of it. I am truly grateful for his involvement, and have many fond memories of our conversations at Sammy's Tap and Grill.

I would also like to thank the rest of my committee, starting with Dr. Jan-Michael Frahm, whose experience and confidence gave me a second-wind when I needed it most, and for being so open about sharing the work of his 3D computer vision group at the University of North Carolina (UNC). Thanks also to Dr. Siamak Khorram for always pointing out new perspectives from remote sensing, to Dr. Ben Watson and Dr. Chris Healey for opening my eyes to new perspectives on light, perception and image formation, and to Dr. Griff Bilbro for all our interesting conversations.

I must also thank Dr. Ademola Ejire for giving me the opportunity to help make an impression on so many promising young students through the Research Apprenticeship Program for High School Students at Shaw University, a truly life-changing experience that also helped support me during my own studies.

In retrospect, I would also like to recognize Dr. Yoonsuck Choe for providing me my first real research opportunity through the NSF Research Experience for Undergraduates (REU) program at Texas A&M University, because our research on receptive field development in the visual cortex first got me thinking about the human visual system; also, Dr. Randal C. Nelson from Rochester University, whose thought provoking conversations inspired me to pursue a degree in 3D computer vision.

Last but not least, the support of my family has been invaluable; in particular I must give special thanks to my mother Dr. Margaret Eppstein for her never-ending willingness to read and review my work, and for never faltering to provide whatever kind of support was needed. I would also like to thank my father, Dr. Bernd Heinrich, who never much liked computers, for keeping me partially grounded in more Earthly things, and for always having faith in me. Thanks to my step father Peter Gillette for sharing his wisdom on so many things electrical and mechanical, and to my brother Dan, who has always been as much a friend as a brother to me.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1

Introduction

This dissertation addresses the general problem of reconstructing static 3D scene structure and camera parameters from an uncalibrated video or image series using structure from motion (SfM). By uncalibrated we mean that no additional information is available beyond the images themselves, and no prior assumptions about the cameras are made other than what is known about all modern cameras (e.g., that the images are not stretched or skewed). The reconstructed camera parameters include relative pose of the camera as well as the intrinsic parameters, most notably focal length, which is unknown and possibly be varying.

We note that with less general assumptions, the SfM problem becomes considerably simpler. For example, estimation of the relative camera movement from a sequence of images (called visual odometry) is commonly performed for robotic navigation systems using a pair of calibrated stereo cameras [Levin and Szeliski, 2004; Takaoka et al., 2004; Olson et al., 2001; Zhang and Faugeras, 1992; Weng et al., 1992a; Olson et al., 2000; Matthies and Shafer, 1987]. Because the relative pose between the stereo cameras is known, dense correspondences can be directly triangulated to obtain the depth of any pixel, or back-projected to yield a 3D point cloud. It is relatively straight-forward to register successive point clouds using the Iterative Closest Point (ICP) algorithm [Besl and McKay, 1992; Chen and Medioni, 1991], thereby revealing the relative movement (or egomotion) of the stereo camera pair.

From a monocular video input, the visual odometry problem is more difficult because structure points cannot be triangulated without knowing the relative pose between successive camera views. Assuming a calibrated camera with unvarying intrinsic parameters, an elegant solution was recently proposed in Nistér [2004], used for example in Pollefeys et al. [2008]. However, the problem is still much more difficult than the stereo case because the estimation of relative pose is not guaranteed to be well-conditioned (depending on how the camera has moved).

Although the above techniques are effective for robotic navigation problems, they cannot be used to make reconstructions from videos that are recorded using hand-held cameras or from

collections of snap-shots that lack camera calibration. This more general uncalibrated problem is considerably more difficult because the reconstruction must be done initially in projective space where familiar concepts such as angles and distance are meaningless. In projective space, an infinite value may correspond to a finite value in the underlying metric space, or a finite value may correspond to an infinite value; thus, even the convex hulls of shapes are not preserved.

Theoretically this projective ambiguity can be removed via autocalibration, but there are certain mathematical degeneracies that must be accounted for, and the process is inherently sensitive to noise; thus, one can never be sure if the ambiguity has been removed completely, and as a consequence it is often unsafe to measure distance in the reconstructed space.

Despite these difficulties, enough of the necessary mathematical ground-work has been worked out for several proof-of-concept systems to be built [Beardsley et al., 1997; Fitzgibbon and Zisserman, 1998; Nister, 2001a; Pollefeys et al., 2004; Repko and Pollefeys, 2005]. Still, none of these systems are completely satisfactory in their ability to produce robust results under all practical circumstances. In particular, previously proposed methods have been very sensitive to initialization because the result of one ill-conditioned estimate will be used as input to another estimation problem, and errors become amplified.

The fundamental goal of this research is to design a system that is more stable and less sensitive to initialization by minimizing the degree to which intermediate estimates are trusted; rather, we try to pose each estimation problem in terms of the original correspondence measurements so as to prevent estimation errors from being compounded.

Due to the large number of disparate topics and related work within each topic that must be integrated into this overall system, we postpone our in-depth review of related work to the context of each chapter. These chapters are organized roughly in the order that they are applied in the context of the overall system, which is described in greater detail in Chapter 2.

One of the major contributions of this work is the overall architecture of the system (Chapter 2). In addition, there are several other major contributions including a new maximum likelihood method for robustly merging projective triplets that is invariant to the ill-estimated structure points and avoids measuring distance in projective space (Chapter 5), and a maximum likelihood method of autocalibration that avoids the instabilities of previous heuristic approaches (Chapter 6).

Other less significant contributions include a new method of keyframe detection (Section 3.1), a survey and analysis of the most robust and precise methods for reconstructing the trifocal tensor (Chapter 4), including the discovery of the final three internal constraints of a trifocal tensor, and a parallel implementation of bundle adjustment (Chapter 7), which is the *de facto* standard way to improve a reconstruction in any SfM problem, whether in projective or metric space.

Chapter 2

System Architecture

In this chapter we discuss the overall architecture of the proposed reconstruction system. We begin by explaining the fundamental components that may be used as building blocks to form a larger reconstruction system and then discuss how these components have been used to design systems in the past. We analyze the theoretical advantages and disadvantages of these different architectures as well as other potential architectures in order to justify our chosen architecture.

We begin with the big picture: the end goal is to reconstruct a detailed surface mesh of the scene geometry, but this is much easier to do once the relative pose of cameras has been established because then any corresponding image points can be triangulated into 3D structure points, and surface reconstruction algorithms can then be used to convert this point cloud into a surface mesh. Thus, the first problem is to accurately estimate the relative pose of each view in the video or image series.

This will be accomplished by using structure from motion (SfM), which uses the fact that the motion of objects in the image plane from a camera translation (called motion parallax) is dependent on depth, to reconstruct all the properties of the camera corresponding to each view simultaneously with the 3D structure points associated with each tracked image point.

From a theoretical standpoint, it is relatively straight-forward to nonlinearly improve a reconstruction of camera parameters and structure points in order to agree with the measured image correspondences (see bundle adjustment in Chapter 7 for the maximum likelihood method). However, there are an astonishingly large number of local minima in any real problem, and thus in order for this nonlinear improvement to converge to the true solution with a reasonable degree of certainty, it will always be necessary to have some direct method of obtaining a good initial solution.

Most typically, direct initial solutions are obtained using either minimal solutions found by clever algebraic rearrangement [Quan, 1995; Carlsson and Weinshall, 1998; Hartley and Debunne, 1998; Hartley and Dano, 2000] or over-determined linear least squares [Hartley, 1995;

Shashua and Werman, 1995]. Almost all linear and minimal solutions that are applicable to SfM have been formulated in terms of projection matrices, which are a compact representation of camera view parameters. These projection matrices can later be factored into more meaningful components representing the camera rotation, translation and intrinsic calibration parameters.

However, real life cameras have certain constraints on their calibration parameters, meaning that projection matrices have more degrees of freedom than real camera views. Unfortunately, these constraints cannot generally be enforced onto the estimation of projection matrices using either the minimal or linear methods, and any reconstruction that does not explicitly enforce these constraints will be subject to what is called the *projective ambiguity*. This projective ambiguity can be resolved in a second phase, called *autocalibration*, by attempting to find the projective warping (a.k.a. rectifying homography) that causes the internal projection matrix constraints to be satisfied for each view. However, autocalibration is not a well-posed problem because, in general, there will not exist a projective warping that causes all such constraints to be exactly satisfied. This is an issue that we address later with a maximum likelihood solution in Chapter 6.

Needless to say, before any SfM algorithm can be applied it is necessary to identify corresponding image points. That is, points in different images that are the projections of the same physical point in 3D. These correspondences can be identified based on local image similarity, although some mismatches will occur; thus, SfM algorithms must be made robust to outliers.

Because the observed motion parallax is a function of both object distance and camera pose, it is necessary to simultaneously estimate the sparse scene structure (i.e., the 3D points associated with each observed image correspondence) along with the projection matrices. The first structure from motion technique applicable to n-view systems was the factorization approach of Tomasi and Kanade [1991], so named because it was shown that an estimate of all structure points and projection matrices could be solved using a single Singular Value Decomposition (SVD) (a type of matrix factorization).

Unfortunately, factorization is not usually a practical solution for several reasons. First, it uses an orthographic approximation to perspective projection, which is extremely inaccurate, especially for points further than a few feet away from the camera. This was partially addressed with nonlinear improvements to correct for perspective distortion [Poleman and Kanade, 1991; Christy and Horaud, 1996; Sturm and Triggs, 1996], but fundamentally there is still no guarantee that the initialization will be good.

The second problem with factorization is that, due to the nature of attempting to solve the entire problem at once, the inevitable presence of outliers will skew the result without indicating which points were outliers. This problem was partially addressed in Jia and Martinez [2009] with a method for detecting dominant outliers, but will always remain a fundamental problem.

By far the greatest problem with factorization is that it cannot handle 'missing data'; that

is, it can only be used to estimate the 3D location of points that have been observed in all n-views. Thus, although the method is not specifically limited to any number of views, it is practically limited by the fact that no feature point can be expected to be observable in all views of a long image sequence.

Because SfM is an inherently sensitive problem, the only way to obtain true robustness to the inevitable outliers is to detect and ignore them. This is typically done by wrapping smaller estimation problems within a Random Sample Consensus (RANSAC) [Fischler and Bolles, 1981] framework, whereby the estimation problem is repeated on random minimal subsets of the data in an attempt to simultaneously estimate the desired parameters and the largest sample consensus of inliers.

It would be ineffective and computationally infeasible to apply RANSAC to extremely large estimation problems, such as a global SfM factorization, because a single outlier in a long track of otherwise good correspondences would make the entire track an outlier. However, if the overall problem is broken down into smaller fixed size problems where only one to four views are estimated at a time, then these can be handled with relative efficiency within the RANSAC framework. Not only does this approach overcome the problem of outliers but it also overcomes the problem of 'missing data' because a correspondence must only be tracked over the small number of views in a partial reconstruction. The partial reconstructions can then be merged together to obtain larger reconstructions spanning any arbitrary number of views.

The challenge in designing a good SfM system is in merging these partial reconstructions together in a way that is stable, robust to noise, robust to outliers, efficient, and robust to degenerate or quasi-degenerate configurations. This is further complicated by the fact that these partial reconstructions must be estimated from nearby views, but the correspondences between nearby views often do not encode for sufficient motion parallax to make the reconstruction problem well-posed.

2.1 Theoretical Comparison of Architectures

The simplest approach, which we call incremental resectioning, was proposed in Beardsley et al. [1997] and has been commonly used since [Pollefeys et al., 2002a, 2004; Engels et al., 2006]. The basic idea is to start by estimating a view pair (using the fundamental matrix [Hartley, 1992; Faugeras, 1992]) and then iteratively extend some correspondences into the next view which are used to estimate the next projection matrix (a process called resectioning). In order to remain robust to outliers, this resectioning should be done using RANSAC. With the addition of each projection matrix, the overall projective reconstruction can be nonlinearly improved to stabilize any potential errors (e.g., using projective bundle adjustment, see Chapter 7). After completing the projective reconstruction, autocalibration is used to resolve the projective ambiguity

yielding a metric reconstruction, which is finally improved using metric bundle adjustment. A flow diagram for this approach is given in Fig. 2.1.

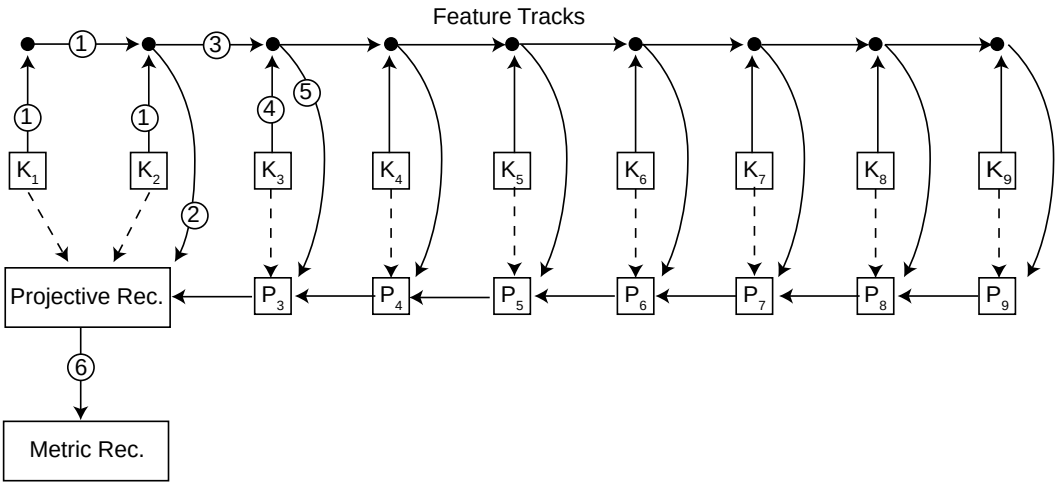


Figure 2.1. Incremental resectioning. (1) find initial pairwise correspondences between keyframes K_1, K_2; (2) estimate fundamental matrix that defines initial projective reconstruction; (3) extend existing feature tracks into K_3; (4) start new feature tracks in K_3; (5) resection view corresponding to K_3 to get the projection matrix P_3. Repeat until all views have been incorporated into a global projective reconstruction. Finally, (6) autocalibrate.

Incremental resectioning is attractive for its simplicity, but it is inherently sensitive to initialization because when resectioning new views using RANSAC, any points that violate the existing reconstruction would be treated as outliers. If the initial reconstruction is poor, this would cause any points that contain sufficient information to correct the current reconstruction to be treated as an outlier (false negative). As one moves further from the initial view pair this effect would increase and could eventually result in failure to find sufficient correspondences for resectioning.

An alternative to incremental resectioning is to compute many small partial reconstructions and then attempt to merge these together into a larger reconstruction. The advantage of a merging approach is that because each partial reconstruction is computed independently there is no propagation of errors or outliers from one to the next, and hence the problem of false negatives remains insignificant.

One of the disadvantages of merging is that each of the partial reconstructions, which are computed independently, could be erroneous if there is not enough camera motion to make the estimation problem well-conditioned. Thus, it is necessary to identify keyframes when sufficient motion parallax has accumulated so that the reconstruction will not be be ill-posed. This is not particularly difficult to do (see Section 3.1) but it does bring about increased computational

expense.

There are many possible variations of merging approaches that could be used, depending on the size of the partial reconstructions (i.e., should one attempt to merge pairs, triplets, or quadruples?), the number of overlapping views to use for each independent reconstruction, the method of merging that is used, whether or not merging is performed incrementally or hierarchically, and how long autocalibration is delayed.

For example, Fitzgibbon and Zisserman [1998] proposed to merge projective triplets with two views of overlap hierarchically (Fig. 2.2), while Nister [2001a] proposed a similar hierarchical merging of triplets but with 1-view overlap, and Repko and Pollefeys [2005] proposed to merge metric triplets with 2-views of overlap by solving an absolute orientation problem of corresponding structure points (Fig. 2.3).

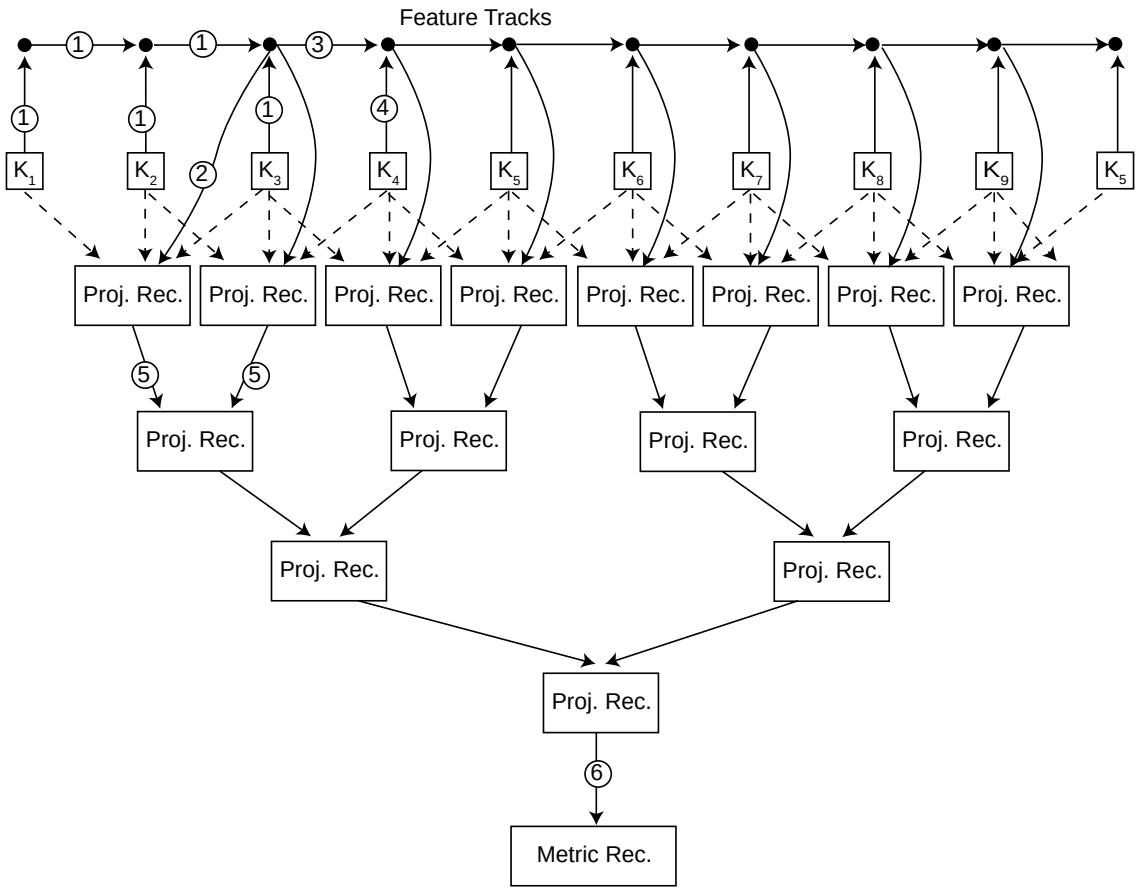


Figure 2.2. Hierarchical merging of projective triplets using 2-view overlap. (1) find initial triplet correspondences between keyframes $K_1 \ldots K_3$; (2) estimate trifocal tensor; (3) extend existing feature tracks into K_4; (4) identify new features in K_4; (5) projective merging with 2-view overlap. After completing the hierarchical projective reconstruction, (6) autocalibrate.

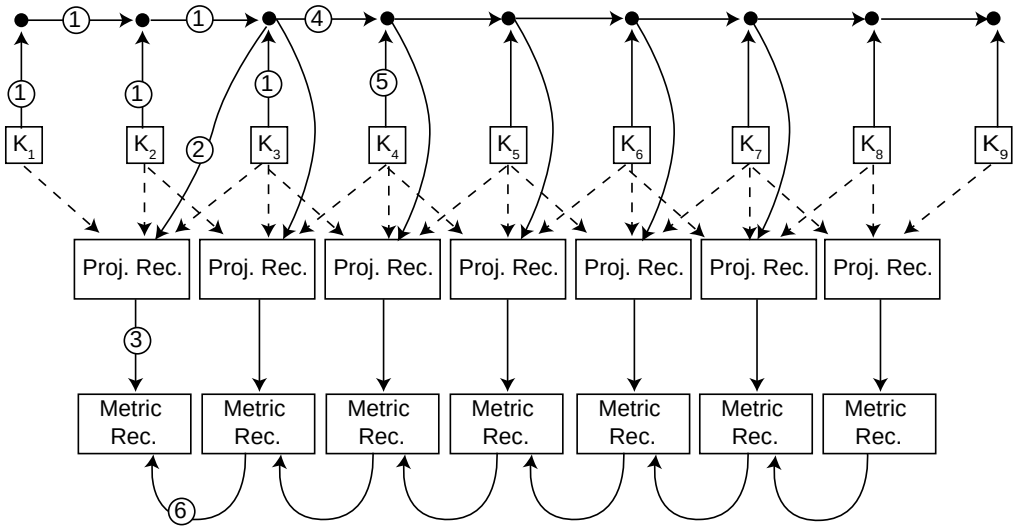


Figure 2.3. Incremental merging of metric triplets using 2-view overlap. (1) find initial triplet correspondences between keyframes $K_1 \ldots K_3$; (2) estimate trifocal tensor that defines projective reconstruction; (3) autocalibrate; (4) extend existing feature tracks into K_4; (5) identify new features in K_4; (6) metric merging with 2-view overlap.

In general, when using m-view partial reconstructions, merging could be performed using anywhere from 0 to $m-1$ views of overlap (see Table 2.1); merging could be performed in projective space with autocalibration deferred to the end, or autocalibration could be performed on each partial reconstruction so that merging can be done in metric space; the reconstruction could be done incrementally, or it could be done hierarchically.

Table 2.1. Theoretical comparison between various potential merging approaches using all combinations of subset views and overlap views. Subset views is the number of views in each partial reconstruction. Overlap views is the number of overlapping views between successive partial reconstructions. Min track length is the minimum number of consecutive views that a feature track must persist through in order to provide merging information. Merge DOF is the number of degrees of freedom in the merging homography that remain after accounting for overlapping view constraints. A negative DOF indicates that the merging homography is over-determined from view constraints alone, and hence the partial reconstructions will be 'mangled' during the merging operation.

Subset Views	**Overlap Views**	**Min Track Length**	**Merge DOF**
Resection			
1	n/a	3	11
Fundamental Matrix			
2	0	4	15
2	1	3	4
Trifocal Tensor			
3	0	4	15
3	1	3	4
3	2	3	-7
Quadfocal Tensor			
4	0	4	15
4	1	4	4
4	2	4	-7
4	3	4	-18

The size of the partial reconstructions effects the length of correspondence tracks that are needed for estimation purposes, as well as overall performance and robustness. Smaller partial reconstructions are easier to estimate because there are fewer internal constraints and, in general, more correspondence data available. However, with fewer views the estimation may be more ambiguous and less well-conditioned. For example, the relationship between two views is represented by the 3×3 fundamental matrix which has just 1 internal constraint, but its estimation is ill-conditioned if the scene structure is planar. If one instead uses triplets, the relationship is represented by the $3 \times 3 \times 3$ trifocal tensor, which has 8 internal constraints, and can be estimated even when the scene geometry is planar.

Merging is possible using constraints derived from view and/or structure point constraints, although the specifics are largely dependent on the number of overlapping views. If the reconstructions are metric then the alignment is defined by a similarity transformation having 7 dof, and can be determined up to a scale factor by a single overlapping view. However, view constraints are not strictly necessary because the similarity transform can also be solved in

closed form corresponding metric structure points [Horn et al., 1988; Umeyama, 1991; Matei and Meer, 1999]; a method that is often used in practice [Repko and Pollefeys, 2005; Farenzena et al., 2009; Frahm et al., 2010].

The problem with merging in metric space is that it becomes necessary to autocalibrate each individual partial reconstruction, but autocalibration is a very sensitive procedure that is unlikely to successfully remove all of the projective distortion when applied to a small reconstruction. Thus, the alignment between two autocalibrated partial reconstructions is unlikely to be well-modeled by a similarity transform.

More generally, when merging two projective reconstructions (or quasi-metric reconstructions), the alignment is represented by a homography having 15 dof. Each overlapping view provides 11 constraints, so a single overlap leaves 4 dof remaining while 2 views of overlap makes the problem over-determined. Over-determined problems are usually good, but not in this case because it means the two views can't be aligned perfectly, and must be discarded during the merge anyway because there cannot be more than one projection matrix per view. Thus, if there are two or more views of overlap, the merging operation will necessarily increase reprojection errors, and this increase in error is completely *unbounded* because it depends only on the location of structure points. These problematic over-determined cases are indicated as negative dof in Table 2.1.

In order to avoid this potentially drastic increase in error, one must use only zero or one view of overlap, meaning that correspondences between structure points are necessary to resolve the remaining ambiguity. In order for a structure point to exist in a partial reconstruction it must have been visible in at least two views to be triangulated. Thus, if there are zero views of overlap, then only correspondence tracks of length four or greater can be used in merging. If there is one view of overlap, then tracks need not be longer than three views. This is a significant difference because longer correspondence tracks are exponentially more difficult to acquire.

We can further conclude that three is the theoretical minimum length of correspondence tracks usable by any approach, because the only way to obtain structure point correspondences from tracks of length two would be to have two views of overlap, but this would require the partial reconstructions to be at least length three. Even the incremental resectioning approach has a minimum track length of three because it requires correspondences between structure points in the previous reconstruction and image points in the new view, and in order for a structure point to exist in the previous reconstruction it must have been visible in at least two views.

Thus, we make the following conclusions: (1) merging is best done in projective space using one view of overlap, with autocalibration reserved to the final stage, and (2) it is preferable to use triplets as the fundamental unit of reconstruction rather than pairs, because the minimum

track length is already three, and triplets provide stronger geometric constraints than pairs and are not degenerate to planar surfaces.

2.2 Proposed Architecture

Our approach differs from conventional tracking-based approaches in the way that we manage correspondences (Chapter 3). Rather than building long feature tracks in an independent module, we search for triplet correspondences on-demand using a guided matching approach (Chapter 3.4.1) whenever a triplet needs to be reconstructed. This allows us to consistently find a sufficient number of correspondences for reliable reconstruction, assuming the images are sufficiently detailed.

In order to ensure that the estimation of each independent triplet is not ill-conditioned, we detect keyframes (Section 3.1) and reconstruct only from those key views using SfM. Once the overall reconstruction of keyframes is complete, the intermediate views can be resectioned.

As motivated by the theoretical concerns discussed in Section 2.1, the proposed SfM architecture uses a hierarchical merging of projective view triplets (Chapter 4), and merging is done using single-view overlap in projective space (Chapter 5). After each initial reconstruction, and after each merge, the reconstruction will be stabilized by using a nonlinear improvement. We always use bundle adjustment (Chapter 7) because it is the maximum likelihood method. Once the projective reconstruction is completed, autocalibration is used to upgrade the projective result to metric (Chapter 6), and improved with metric bundle adjustment. Finally, a surface mesh of the scene geometry can be computed (Chapter 8). A flow chart for the SfM process (which includes all but the final surface reconstruction) is shown in Fig. 2.4, and is explained below.

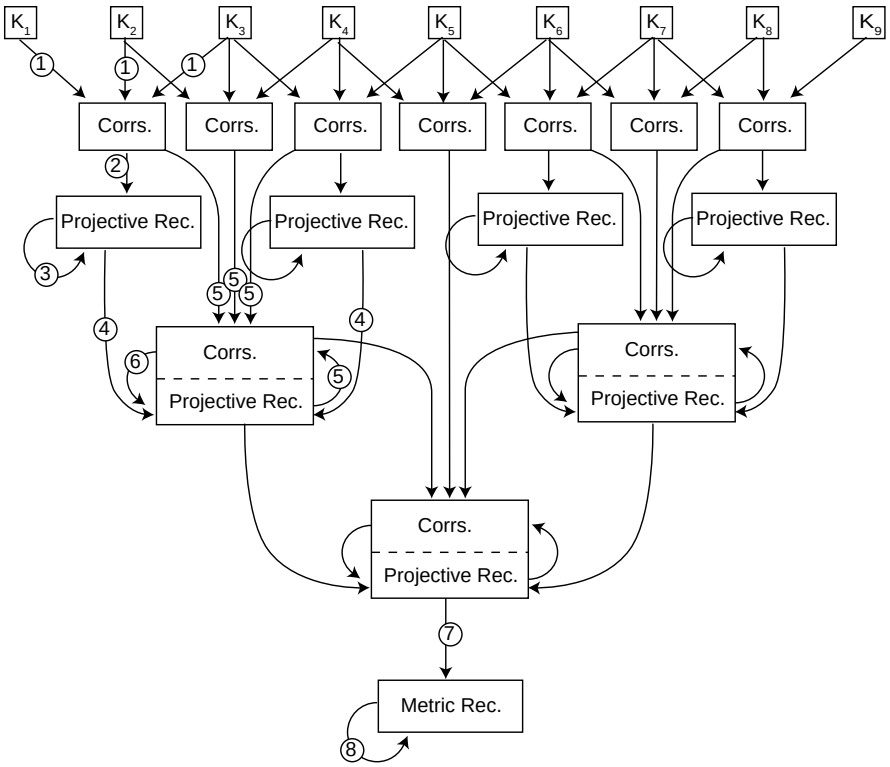


Figure 2.4. Proposed merging architecture using hierarchical merging of projective triplets and correspondences using 1-view overlap. (1) find triplet correspondences between keyframes $K_1 \ldots K_3$; (2) estimate trifocal tensor that defines projective reconstruction; (3) projective bundle adjustment; (4) projective merging with 1-view overlap; (5) merge correspondences; (6) projective bundle adjustment. After completing the hierarchical projective reconstruction, (7) autocalibrate; (8) metric bundle adjustment.

When two partial reconstructions need to be merged, we first find triplet correspondences and reconstruct a robust triplet spanning the gap between them; this ensures that we obtain a large number of correspondences for merging that are almost all guaranteed to be inliers.

Longer feature tracks are obtained via merging of the initial triplet correspondences, which is done every time we merge two reconstructions (Section 5.6). The advantage of merging correspondences as opposed to incrementally building long feature tracks is that the problem of feature drift is avoided and the presence of potential outlier matches does not cause existing good structure points to be dropped from the reconstruction.

The choice of incremental vs. hierarchical merging is mostly computational. If bundle adjustment is performed after each merge in order to ensure stability, then the size of the full bundle adjustment problem grows with each merge; the computational complexity of adding a single view to a reconstruction of n points and m views is therefore $O((3n + 12(m-1))^3)$ (see Section 7.3), and this quickly becomes impractical for large numbers of views.

In order to put an upper bound on the computational complexity of adding a single view, only a partial windowed bundle adjustment can be used on the most recent views, as in Engels et al. [2006]. In contrast, if merging is performed hierarchically, then the same number of bundle adjustments will be needed but on average they will be much smaller, and thus the need to resort to windowed bundle adjustment is not as great. Moreover, hierarchical merges can be done in parallel (as is implemented in our system), whereas this cannot be done using an incremental approach.

Chapter 3

Finding Correspondences and Detecting Keyframes

In order to make a reliable reconstruction over a set of views one must have correspondences between those views. It is important to have a large number of correspondences so that the reconstructed camera parameters and structure will be strongly over-determined, making their estimation from imprecise measurements more accurate, and allowing RANSAC [Fischler and Bolles, 1981] (Section 4.3) to be used to effectively remove outliers.

Because our hierarchical merging approach relies upon reconstructing many independent triplet reconstructions, the correspondences between views used for initial triplets must demonstrate sufficient motion parallax or the estimated camera poses will be ill-conditioned. Therefore, the detection of views with sufficient motion parallax, which we call keyframes, is closely related to the problem of finding correspondences.

In our work we have adopted a novel method for keyframe detection (Section 3.1), as well as two different front-end methods for acquiring correspondences: one based on KLT tracking for high frame rate video (Section 3.3), and another using guided matching that is more robust for image series with wider baseline (Section 3.4). In the former, the detection of keyframes is integrated into the tracking algorithm. We detect feature points for the purpose of both tracking and wide baseline matching in the same way (Section 3.2).

These methods are largely inspired by previous work but contain some of our own improvements as well. However, it should be stressed that the focus of this dissertation is not on correspondence finding, but rather on the reconstruction methods that follow. Furthermore, despite the large amount of prior research, we still do not consider feature tracking to be a well-solved problem. In particular, the convergence of KLT is often less than ideal, especially for small patches, and this may force one to use patches that are larger than desired (which reduces precision and performance).

3.1 Identifying Keyframes

The amount of camera translation that is necessary for stable reconstruction is not an absolute quantity but is relative to the geometry of the scene and the precision of correspondences. Because the relative error for more distant correspondences are greater, more distant geometry requires a wider absolute baseline for reliable reconstruction. Thus, it is not sufficient to use evenly spaced keyframes in an arbitrary video even when frame rate or camera movement is known to be constant. Instead, keyframes must be detected from the correspondences by attempting to quantify the amount of usable motion parallax.

In general, any set of corresponding image points $\mathbf{x} \leftrightarrow \mathbf{x}' \in \mathbb{P}^2$ in two views must satisfy the epipolar constraint, which depends upon the 3×3 *fundamental matrix* $\mathbf{F}$,

$$\mathbf{x}'^{\mathsf{T}}\mathbf{F}\mathbf{x} = \mathbf{0}. \tag{3.1}$$

The epipolar constraint arises from the fact that any point $\mathbf{x}$ in the first view defines a ray, and this ray is imaged as a line $\mathbf{F}\mathbf{x}$ in the second view that the corresponding point $\mathbf{x}'$ must lie on, regardless of depth. Thus, the fundamental matrix encodes for the relative projective pose between views, known as the epipolar geometry of the scene.

If the scene geometry is coplanar, or if the cameras have the same focal point so that there is no motion parallax, then the transformation between corresponding image points will be perfectly described by a 3×3 homography $\mathbf{H}$,

$$\mathbf{x}' \propto \mathbf{H}\mathbf{x}. \tag{3.2}$$

Thus, when the relative distance between view focal points (known as baseline) is small in comparison to the depth of scene geometry, motion parallax becomes negligible and the homography becomes an accurate model. If the correspondences can be well-modeled by a homography, estimation of the fundamental matrix will not be well-conditioned because the structure of the scene is indeterminate.

The problem of keyframe detection is to select views from the image series that do not suffer from degeneracies that would make the subsequent estimation of projective geometry (e.g., using the fundamental matrix) ill-conditioned. Because we use the trifocal tensor in our hierarchical reconstruction framework, the issue of planar geometry does not pose a specific problem, but it is still important that frames have sufficient baseline to display motion parallax.

Because $\mathbf{F}$ encodes for the epipolar geometry that we wish to reconstruct, and $\mathbf{H}$ works precisely when $\mathbf{F}$ does not, one way to detect keyframes is to use the Geometric Robust In-

formation Criterion (GRIC) [Torr, 1997, 2002] to detect when **F** fits the measurements better than **H**, as was done in [Repko and Pollefeys, 2005; Pollefeys et al., 2002a].

However, it may take more frames than necessary before **F** is preferred, and this would effectively reduce the number of available correspondences that could be found because it becomes more difficult to find correspondences for more widely separated views. Thus, there is a very sensitive tradeoff: if the baseline is too small, then estimation of **F** will be inaccurate because the relative noise in correspondences becomes large in comparison to motion parallax. Conversely, if the baseline is too large, then fewer correspondences will be found and hence estimation of **F** will either be less over-determined, or there may be insufficient data to make an estimate, causing the reconstruction to be prematurely cut short.

For the purposes of reconstruction, our interest is not really to assess which model is *better*, but rather to assess whether there is *enough* information available to estimate **F** with reasonable accuracy. This is in the same spirit as the approach of Beder and Steffen [2006], where keyframes were detected by analyzing the roundness of the uncertainty ellipsoids of triangulated points.

Our solution to this problem is to identify a set of correspondences that is uncontaminated by outliers and then explicitly test the correspondences for planar degeneracy. In general, a set of correspondences that is free from outliers can be found by estimating **F** using RANSAC [Fischler and Bolles, 1981]. It is important to note that even though **F** is not *uniquely* determined from homography correspondences, it is still possible to find an **F** such that homography correspondences are satisfied by the epipolar constraint; thus, it can still be used to detect outliers even when its estimation is not well-posed. We then fit a homography to the set of inliers, and if the data is well fit by a homography then the estimate of **F** must have been ill-conditioned. Thus, our keyframe detection amounts to a threshold on the mean residual value, which is implemented using a sequential search from each previous keyframe (see Algorithm 1). This is demonstrated graphically in Fig. 3.1.

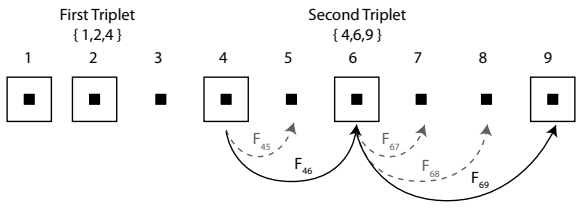


Figure 3.1. A hypothetical example showing how keyframes might be identified in an image series. In this example, frames 1,2,4 have already been established as keyframes, and frame 4 is initially the current frame. The system estimates the fundamental matrix between the last keyframe (view 4) and the next view (5), but the residual error from the homography is too low to be classified as a keyframe. The process is repeated between frames 4 and 6, and this time the threshold is exceeded, so frame 6 is marked as a keyframe. Successive frames are now checked against frame 6 because it is the previous keyframe, and the final keyframe is found at frame 9.

This is also conceptually similar to some other methods that explicitly test for degeneracy of the fundamental matrix, such as QDEGSAC [Frahm and Pollefeys, 2006] or the method of Chum et al. [2005]; however, the objective of those algorithms is to make a robust estimate of $\mathbf{F}$ in the presence of (quasi)-degenerate data, and this is not a significant concern for keyframe detection because we are only concerned with finding a set of inliers rather than making a good estimate of $\mathbf{F}$.

Algorithm 1 Detect all Keyframes

Require: $keyThresh$ is a threshold on the minimum amount of motion parallax demanded between each keyframe, and $minCorrs$ is the minimum number of correspondences needed for reliable estimation.

Ensure: $keyframes$ is a list of keyframes.

```
lastkey ← 1
keyframes ← {lastkey}
hErr ← 0
repeat
    i ← lastkey + 1
    corrs ← FindCorresondences(lastkey, i)
    inliersF ← InliersFromRobustF(corrs)
    if size(inliersF) < minCorrs then
        return
    end if
    H ← EstimateHomography(inliersF, corrs)
    hErr ← MeanFitError(H, inliersF, corrs)
    if hErr > keyThresh then
        Add i to keyframes.
    end if
until i + 1 > totalFrames
```

We find that a fixed threshold for $keyThresh$ of about 0.6% of the image width is effective. Thus, for a 640×480 image, a threshold of about 3.84 pixels is sufficient. This may at first seem like a very small displacement; however, one must remember that this is a threshold on the distance *remaining* after having factored out all the motion that could be explained by a homography. A homography can usually account for the vast majority of disparity (e.g., 90%), so a threshold of 3.84 could easily correspond to an image displacement on the order of 40 or

more pixels, and because this is only a threshold on the mean value, there would probably be some correspondences with larger (e.g., 100 pixels) displacement.

When using the feature tracking method to generate correspondences (Section 3.3), Algorithm 1 is combined with tracking to form a single algorithm. However when using the wide baseline matching approach Section 3.4, Algorithm 1 is executed as an independent algorithm.

3.2 Feature Point Detection

Good features to track are those that can be uniquely and reliably identified in both spatial dimensions of the image plane in spite of rotation, perspective distortion, illumination change and image noise. Primarily these points can be identified as either corners (the intersections of two edges or endpoint of a line) or as roughly elliptical blobs of color.

We detect feature points in much the same way as was the original approach laid out by Harris and Stephens [1988], by looking for points where the *structure tensor* has two equally large eigenvalues. The structure tensor of image $f(x, y)$ is given by

$$\mathbf{S} = \begin{bmatrix} \left(\frac{\partial f}{\partial x}\right)^2 & \frac{\partial f}{\partial x}\frac{\partial f}{\partial y} \\ \frac{\partial f}{\partial x}\frac{\partial f}{\partial y} & \left(\frac{\partial f}{\partial x}\right)^2 \end{bmatrix}, \tag{3.3}$$

and its eigenvalues are given by

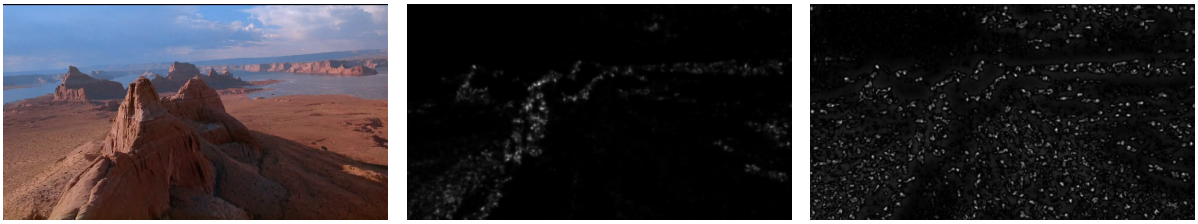
$$\{\lambda_1, \lambda_2\} = \mathbf{S}_{11} + \mathbf{S}_{22} \pm \sqrt{4\mathbf{S}_{11}\mathbf{S}_{22} + (\mathbf{S}_{11} - \mathbf{S}_{22})^2}, \tag{3.4}$$

where $\mathbf{S}_{ij}$ are the elements of $\mathbf{S}$. The original heuristic used by Harris was designed specifically to avoid the square root in (3.4) because at the time it was deemed too computationally intensive. This is no longer the case, and it has since been noticed [Tomasi and Kanade, 1991] that $\min(\lambda_1, \lambda_2)$ is a better heuristic. This heuristic responds strongly to any point in the image plane that can be uniquely localized in both spatial dimensions based on the local image gradients. When dealing with color images we simply sum this heuristic over all three color channels.

Feature points can be detected at the maxima of the heuristic response, but there are usually too many maxima and many of them are not good, so we employ a minimum threshold on the heuristic as well. Although the threshold is good at selecting strong corners, it can also cause large regions to be made devoid of corners. Ideally we would like corners to be uniformly distributed across the image plane because clusters of corners that are very close together are largely redundant. Therefore we have adopted a scheme for locally normalizing the heuristic prior to thresholding based on the standard deviation of corner responses in a larger local neighborhood. Specifically, if $h(x, y) = \min(\lambda_1, \lambda_2)$, then we use

$$h'(x,y) = \frac{h(x,y) - \mu}{\max(\sigma, \sigma_{min})}, \tag{3.5}$$

where μ, σ are the local mean and standard deviation of $h(x,y)$, and σ_{min} is a constant that damps the response in homogeneous areas to prevent small levels of image noise from being detected as feature points. The threshold can then be specified in terms of σ; ie, 1.96σ. An example of the heuristic response using this method is shown in Fig. 3.2.

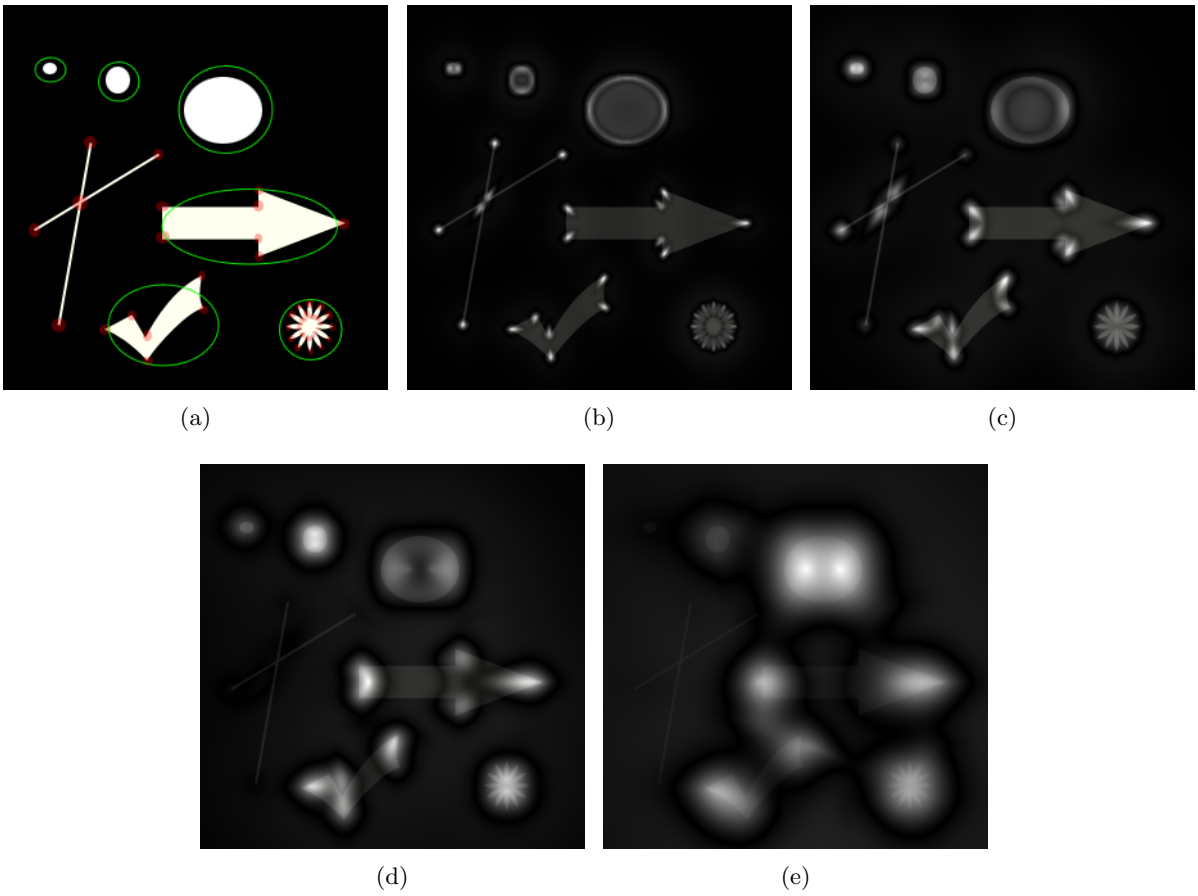


(a) (b) (c)

Figure 3.2. Example heuristic interest point response function. (a) input image; (b) corner heuristic from Tomasi and Kanade [1991] summed over all color channels; (c) corner heuristic after incorporating our weighting to promote a uniform distribution.

After computing the heuristic at each point in the image the local maxima indicate the location of feature points. This approach can be used to detect features of any desired size by blurring the image using the *differentiation scale* σ_D, and then blurring the gradient images using the *integration scale* σ_I. The purpose of the differentiation scale is to improve the robustness of the finite difference approximation to the local image gradient, whereas the integration scale represents the size of the neighborhood to integrate over, and hence the scale of the feature being detected. It is recommended to couple these two scales [Mikolajczyk and Schmid, 2004] with $\sigma_D = 0.7\sigma_I$ so that there is effectively only one free scale parameter.

An example of the heuristic response over a range of scales is shown in Fig. 3.3, where it can be seen that both corners and blobs are detected using this approach.



(a) (b) (c)

(d) (e)

Figure 3.3. Example of multi-scale feature detection. In each image we have overlaid the heuristic corner/blob response function with 85% opacity on top of the input image. (a) input image, with human classified corner features dotted in red, and blob features circled in green; (b) heuristic feature response at scale of 2.0 pixels; (c) heuristic feature response at scale of 4.0 pixels; (d) heuristic feature response at scale of 8.0 pixels; (e) heuristic feature response at scale of 16.0 pixels.

Blob-like features are commonly detected as points that maximize the response of a Laplacian-of-Gaussian (LoG)

$$LoG(\mathbf{x}, \sigma) = \frac{\partial^2}{\partial x^2} G(\mathbf{x}, \sigma) + \frac{\partial^2}{\partial y^2} G(\mathbf{x}, \sigma) \tag{3.6}$$

$$= \frac{1}{\pi\sigma^4} \left(\frac{x^2 + y^2}{2\sigma^2} - 1 \right) \exp\left(-\frac{x^2 + y^2}{2\sigma^2} \right), \tag{3.7}$$

or the more computationally efficient Difference-of-Gaussians (DoG) [Marr and Hildreth, 1980] approximation. Because the LoG has a negatively weighted core surrounded by positive weights

it responds highly to blobs of color of the appropriate size. Thus, each blob feature has a particular scale at which it responds most highly to the LoG, known as the *characteristic scale* Lindeberg [1998].

However, the LoG also responds highly along edges, and therefore it must be used in combination with some other heuristic criterion in order to avoid the many spurious local maxima in an image. For example, the Harris-Laplace detector [Mikolajczyk and Schmid, 2004] looks for points in position-scale space that are simultaneously local maxima of the multi-scale Harris corners at their characteristic scales.

The LoG is also frequently used in affine invariant feature detectors (see Mikolajczyk et al. [2005] for thorough empirical comparisons). These affine invariant features can be useful in very wide baseline matching but are not necessary for finding good points to track. We have investigated using feature points based on the LoG but opted in favor of multi-scale Harris points because they are more robust and can also be used to detect blobs (as opposed to simply corners) by simply increasing the scale parameter.

3.3 Feature Tracking

In the seminal work of Lucas and Kanade [1981], a generic technique for nonlinear image registration under a translation was proposed. This was extended to an affine warping by Shi and Tomasi [1994], and Kanade-Lucas-Tomasi (KLT) feature tracking has been the *de facto* standard way of tracking large numbers of correspondences in video ever since.

A number of related KLT variations were summarized in the unifying framework of Baker and Matthews [2004], in which it was concluded that the *inverse compositional* derivation is the most efficient for feature tracking. This was used to implement the first real time KLT tracker by Jin et al. [2001], which utilized an affine photometric model to increase the reliability of tracking under illumination changes due to reflection angle and camera automatic gain control. More recently, real time KLT tracking has been performed on the GPU [Hedborg et al., 2007; Sinha et al., 2007; Ohmer and Redding, 2008; Zach et al., 2008; Hwangbo et al., 2009; Kim et al., 2009; Phull et al., 2010], and this can also be done with an affine photometric model [Zach et al., 2008; Hwangbo et al., 2009; Kim et al., 2009].

Our feature tracking algorithm (Algorithm 3.4) dynamically locates new features, tracks them, and identifies keyframes that have sufficient motion parallax by integrating with Algorithm 1. Feature points are identified as described in Section 3.2 and then tracked as described in Section 3.3.1.

We use two methods of outlier detection on the feature tracks. First, we reject tracks when the Normalized Cross Correlation (NCC) between the original template and the most recent match falls below a threshold of 0.95. This is similar to the Sum of Squared Differences (SSD)

dissimilarity measure used in Shi and Tomasi [1994], but we find that the NCC is a more reliable measure in the presence of illumination changes, and has yielded more reliable performance for us than the X84 rejection rule [Tommasini et al., 1998; Fusiello et al., 1999]. Secondly, we test the epipolar constraint against the fundamental matrix that is estimated for keyframe detection.

Combined Feature Tracking and Keyframe Detection Algorithm

1. Register the images using a perspective photometric model with KLT.
2. For each feature, initialize the affine photometric model with the overall perspective photometric model and then register the template using KLT.
3. Terminate all feature tracks where the NCC falls below a threshold τ_1.
4. Compute a robust estimate of the fundamental matrix $\mathbf{F}$ between the current frame and the previous keyframe using all tracks.
5. Terminate all feature tracks that violate the epipolar constraint by more than τ_2 using $\mathbf{F}$.
6. Estimate the homography $\mathbf{H}$ from all fundamental matrix inliers. If the mean residual error is greater than τ_3 then this is a new keyframe.
7. Detect corners in the new image as local maxima of the multi-scale Harris detector. For each corner point, lookup the nearest feature track. If the distance is greater than τ_4, then use the corner to initialize a new feature.

Figure 3.4. Summary of our algorithm for combined feature tracking and keyframe detecting.

Some examples of the feature tracks found using our tracker are shown in figures 3.5 and 3.6. A few outliers can be seen in these examples, but these do not pose a significant problem because they will be filtered out later on by the trifocal tensor constraints during reconstruction.

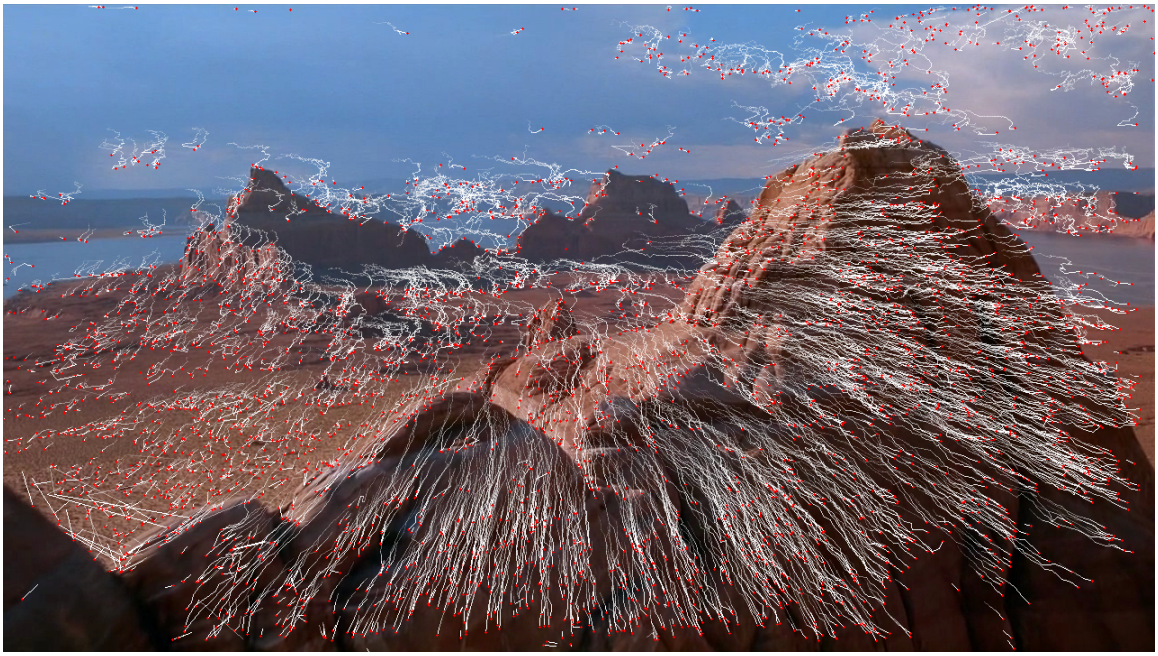


Figure 3.5. Active feature tracks on frame 296 of an aerial helicopter video. The centroid of each feature in the current frame is shown as a red dot, with a white trail indicating past history.

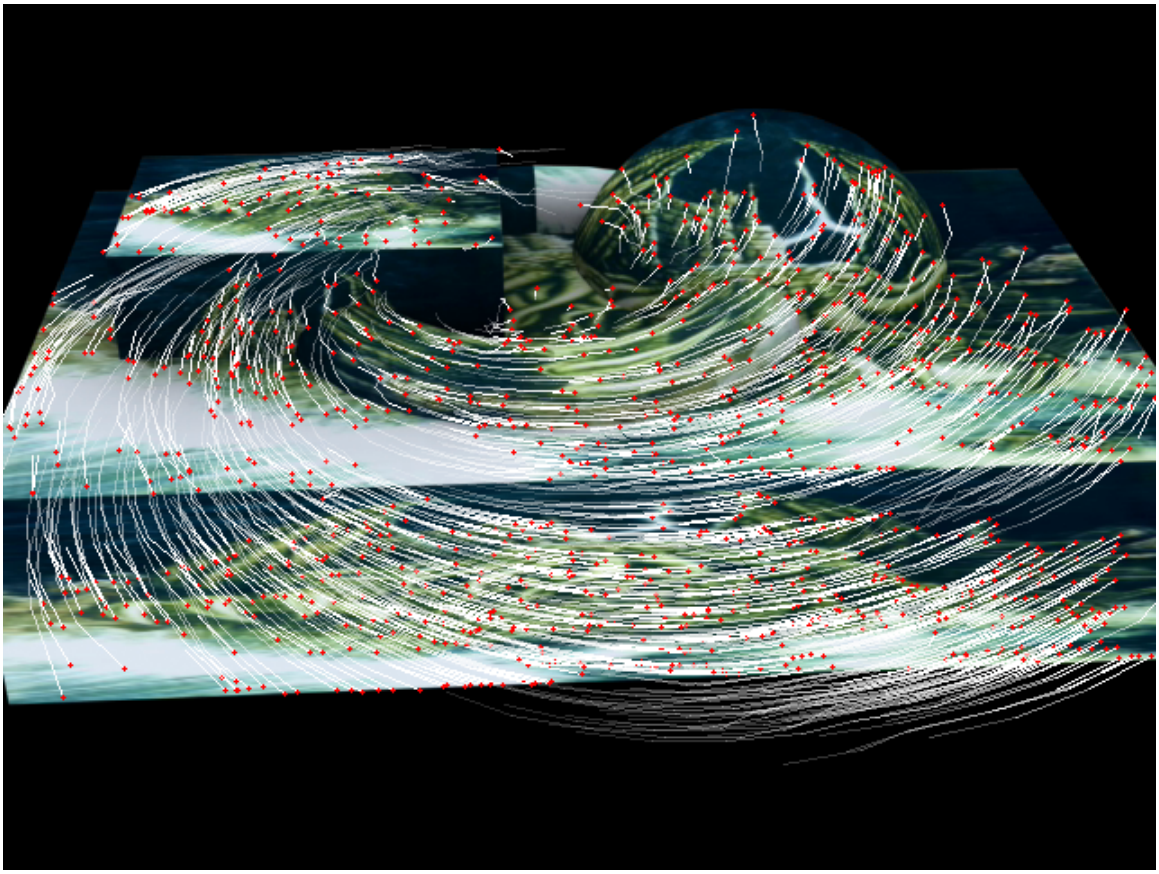


Figure 3.6. Active feature tracks on frame 14 of a synthetic video. The centroid of each feature in the current frame is shown as a red dot, with a white trail indicating past history.

3.3.1 Lucas-Kanade Image Registration

Given a parameterized model of allowable transformations $\mathbf{W}(\mathbf{x};\mathbf{p})$, the goal of the Lucas and Kanade [1981] image registration algorithm is to find the parameter vector $\mathbf{p}$ that minimizes the Sum of Squared Differences (SSD) between a template image $T(\mathbf{x})$ and a search image $I(\mathbf{x})$ transformed back into the reference frame of the template,

$$\sum_{\mathbf{x}\in\mathcal{A}} \left(I(\mathbf{W}(\mathbf{x};\mathbf{p})) - \mathbf{T}(\mathbf{x})\right)^2, \tag{3.8}$$

where $\mathcal{A}$ is the area of the template image. The algorithm assumes that $\mathbf{p}$ is approximately known and then iteratively computes an update $\Delta\mathbf{p}$ for nonlinear improvement. Thus, each step approximately minimizes

$$\sum_{\mathbf{x}\in\mathcal{A}} \left(I(\mathbf{W}(\mathbf{x};\mathbf{p}+\Delta\mathbf{p})) - \mathbf{T}(\mathbf{x})\right)^2. \tag{3.9}$$

Equation (3.9) is linearized using a first-order Taylor expansion to give

$$\sum_{\mathbf{x}\in\mathcal{A}} \left(I(\mathbf{W}(\mathbf{x};\mathbf{p})) + \Delta I \frac{\partial \mathbf{W}}{\partial \mathbf{p}} \Delta\mathbf{p} - \mathbf{T}(\mathbf{x})\right)^2, \tag{3.10}$$

where $\Delta I = \left(\frac{\partial I}{\partial x}, \frac{\partial I}{\partial y}\right)$ is the gradient of I evaluated at $\mathbf{W}(\mathbf{x};\mathbf{p})$, and $\frac{\partial \mathbf{W}}{\partial \mathbf{p}}$ is the Jacobian of $\mathbf{W}(\mathbf{x};\mathbf{p})$. The partial derivative of (3.10) with respect to $\Delta\mathbf{p}$ is

$$\sum_{\mathbf{x}\in\mathcal{A}} \left(\Delta I \frac{\partial \mathbf{W}}{\partial \mathbf{p}}\right)^{\mathsf{T}} \left(I(\mathbf{W}(\mathbf{x};\mathbf{p})) + \Delta I \frac{\partial \mathbf{W}}{\partial \mathbf{p}} \Delta\mathbf{p} - \mathbf{T}(\mathbf{x})\right). \tag{3.11}$$

Setting (3.11) to zero and solving gives the closed form solution for the approximate minimum as

$$\Delta\mathbf{p} = \mathbf{H}^{-1} \sum_{\mathbf{x}\in\mathcal{A}} \mathbf{J}^{\mathsf{T}}(T(\mathbf{x}) - I(\mathbf{W}(\mathbf{x};\mathbf{p}))), \tag{3.12}$$

where $\mathbf{J} = \Delta I \frac{\partial \mathbf{W}}{\partial \mathbf{p}}$ is the Jacobian of $I(\mathbf{W}(\mathbf{x};\mathbf{p}))$ and $\mathbf{H} = \mathbf{J}^{\mathsf{T}}\mathbf{J}$ is the Gauss-Newton approximation to the Hessian.

It was shown in Lucas and Kanade [1981] that the algorithm will converge for periodic functions when the initialization is within one-half period. Thus, it is recommended to minimize using a coarse to fine approach on an image pyramid, starting with low-frequency information and progressively refining the transformation as higher frequency content is added back in, thereby achieving high precision while also remaining relatively robust to local minima.

This algorithm can be used for registering large images or small templates, although it becomes progressively less reliable for smaller template sizes. The basic approach derived above is referred to as the *forward additive* model. There are a number of other derivations that have been compared in Baker and Matthews [2004], in which it was determined that the *inverse compositional* derivation is mathematically equivalent up to first order, but is more efficient for repeated template matching because the Hessian can be precomputed. Since Jin et al. [2001], this is the method used by most modern feature trackers.

3.3.1.1 Transformation Models

If we approximate the geometry of the template patch as being locally planar, then its projective transformation is described by a 3×3 homography of $\mathbb{P}^2$ denoted by $\mathbf{H}$. Using homogeneous coordinates, this transformation is given by

$$\mathbf{W}(\mathbf{x}; \mathbf{p}) = \mathbf{H}\mathbf{x}, \tag{3.13}$$

where $\mathbf{p}$ is an 8-vector containing the elements of $\mathbf{H}$ (less one that can be fixed for scale). This is the most general warping model that has been used within the KLT framework [Baker and Matthews, 2004], although modern feature trackers generally use an affine warping instead [Shi and Tomasi, 1994; Jin et al., 2001; Zach et al., 2008; Hwangbo et al., 2009; Kim et al., 2009] because it has fewer parameters, making it less prone to over-fitting for small templates. Using inhomogeneous coordinates, the affine transformation is given by

$$\mathbf{W}(\mathbf{x}; \mathbf{p}) = \mathbf{A}\mathbf{x} + \mathbf{b}, \tag{3.14}$$

and now $\mathbf{p}$ is a 6-vector containing the elements of $\mathbf{A}$ and $\mathbf{b}$. In order to deal with illumination changes due to camera automatic gain control and perspective-dependent surface reflection, an illumination model can be employed as well. This has been done using the affine photometric model [Jin et al., 2001; Zach et al., 2008; Hwangbo et al., 2009; Kim et al., 2009],

$$I(\mathbf{W}(\mathbf{x}; \mathbf{p})) = \beta + \alpha I(\mathbf{A}\mathbf{x} + \mathbf{b}), \tag{3.15}$$

where β represents an additive brightness term and α is a multiplicative gain factor that represents contrast change. In this case $\mathbf{p}$ is an 8-vector containing all photometric parameters. We use the affine photometric model for the same reasons described above.

3.3.2 Homography Initialization

Feature trackers that use the KLT typically assume an identity transformation for the initial estimate of the transformation parameters $\mathbf{p}$ of each feature. However, if one has a prior model for the overall image registration, this can be used to initialize the local feature patches for improved performance and reliability of convergence.

For example, in Hwangbo et al. [2009] and Kim et al. [2009] an external Inertial Motion Unit (IMU) sensor was used to get a rough estimate of the rotation between frames, and in Phull

et al. [2010] registration of each feature using a scale pyramid was abandoned in favor of using variable sized features that were registered in order of decreasing size, with the transformation of each matched feature being used to initialize the next smaller feature.

Our approach is simply to downsize the two images and then register them with KLT using a perspective photometric model. From this estimate of the overall image homography we initialize the local affine photometric parameters of each feature. This is more accurate than the method of Hwangbo et al. [2009] because it uses the precise image information rather than relying upon imprecise external measurements, and because a homography is a more general model than a rotation it can provide a better initialization. It is also simpler and more accurate than the method of Phull et al. [2010] because the features do not need to be multi-scale, and the overall image homography is more likely to provide a good initialization for any given feature than the local homography used by another small feature.

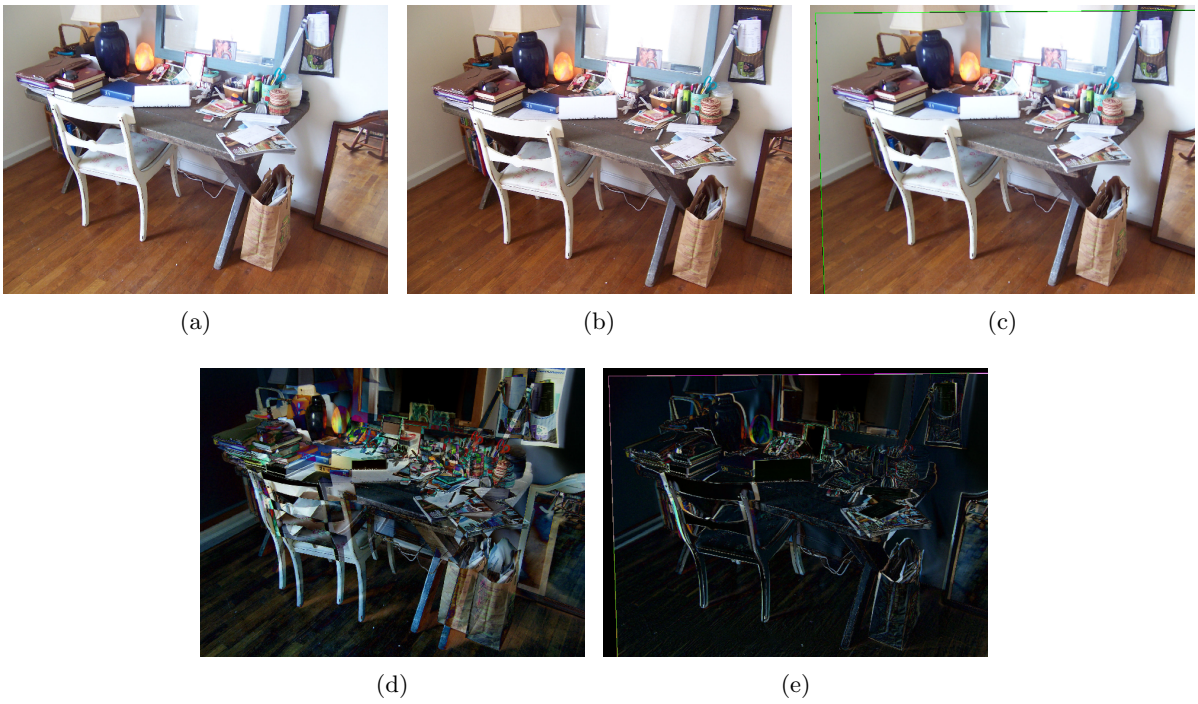
In order to initialize the local affine transformations from the global homography we recognize that each of the four corners $\mathbf{c}_i = (x_i, y_i)^\mathsf{T}, i = 1 \ldots 4$ of the template T should be transformed to

$$(x_i', y_i')^\mathsf{T} = \mathbf{H}\left(\mathbf{A}\mathbf{c}_i + \mathbf{b}\right), \tag{3.16}$$

where $\mathbf{H}$ represents the overall image registration, and $\mathbf{A}$ and $\mathbf{b}$ represent the affine transformation of the patch in the previous frame. Thus, each corner provides two linear constraints on the affine parameter vector $\mathbf{p}$ given by

$$\begin{bmatrix} x_i & y_i & 0 & 0 & 1 & 0 \\ 0 & 0 & x_i & y_i & 0 & 1 \end{bmatrix} \mathbf{p} = \begin{bmatrix} x_i' \\ y_i' \end{bmatrix}, \tag{3.17}$$

from which $\mathbf{p}$ can be solved using linear least squares. The photometric parameters of the patch are then directly copied from the photometric parameters of the overall perspective photometric image registration. An example of the registration computed using this approach on a pair of images with relatively wide baseline is shown in Fig. 3.7, where it can be seen from the difference images that the image alignment is much closer, and hence the nonlinear convergence of small patches will be improved.



(a) (b) (c)

(d) (e)

Figure 3.7. Example of image registration using a homography to reduce search distance. (a) previous keyframe; (b) current keyframe; (c) previous keyframe after being registered into the frame of current keyframe using a homography; (d) image difference between previous and current keyframes; (e) image difference between previous keyframe registered with current keyframe.

3.4 Wide Baseline Matching

Even when supplied with a good initial homography registration and relatively large templates, we have observed that KLT feature matching can be quite unstable when the images differ by more than a small amount. Therefore, we have adopted a more robust matching approach to deal with some image sequences.

As noted in Shi and Tomasi [1994]; Jin et al. [2001], feature tracks that are constructed by inter-frame matching are prone to the systematic accumulation of error (a.k.a. feature drift) that would prevent a long track from being triangulated into a single structure point, and cause any reconstruction attempt to become highly unstable. To cope with this issue we do not attempt to track features across intermediate frames; rather, we directly search for correspondences between keyframes when reconstructing trifocal tensors.

Given any two frames that we wish to find correspondences between (such as during keyframe detection in Algorithm 1), we first detect feature points in the first image (Section 3.2) and then match these features in the second image using guided matching (Section 3.4.1).

When it becomes time to estimate a trifocal tensor we will require triplet correspondences, and we find these by doing two successive guided matchings. In other words, if the keyframes are $\{k_1, k_2, k_3\}$, then we first search for features in k_1 and then extend these using a guided matching from $k_1 \rightarrow k_2$, and finally with a second guided matching from $k_2 \rightarrow k_3$. Although the second matching is susceptible to accumulated error from the first matching, we do not ever repeat the process into additional frames, and thus the error does not systematically accumulate.

Additionally, we have developed a novel system of merging inter-frame correspondences into longer feature 'tracks' that avoids the problem of feature drift (Section 5.6); however, unlike the conventional feature tracking algorithm which may be computed as a separate module, this requires a deeper level of integration into the overall reconstruction system.

3.4.1 Guided Matching

Because we wish to use the method for extending matches across triplets, our objective is to take an initial set of features in one image and output a set of matching points in a second image. Our approach is to first estimate the overall motion between views and then use this motion model to guide the search for our matches of interest. This is conceptually similar to the guided matching approach described in Pollefeys et al. [1998]; Hartley and Zisserman [2004], but different in that we do not actively identify new feature points during the process.

Because the correspondences between two images can be related in various degrees by either a homography or fundamental matrix (Section 3.1), both of these matrices can be used to restrict the search region, as shown in Fig. 3.8. The fundamental matrix restricts the search space to an epipolar line (within some small tolerance related to the precision of the estimate), and the homography restricts the search space to a point (with some large tolerance related to the relative baseline and effective motion parallax). Therefore, our first step is to compute an estimate of both of these matrices.

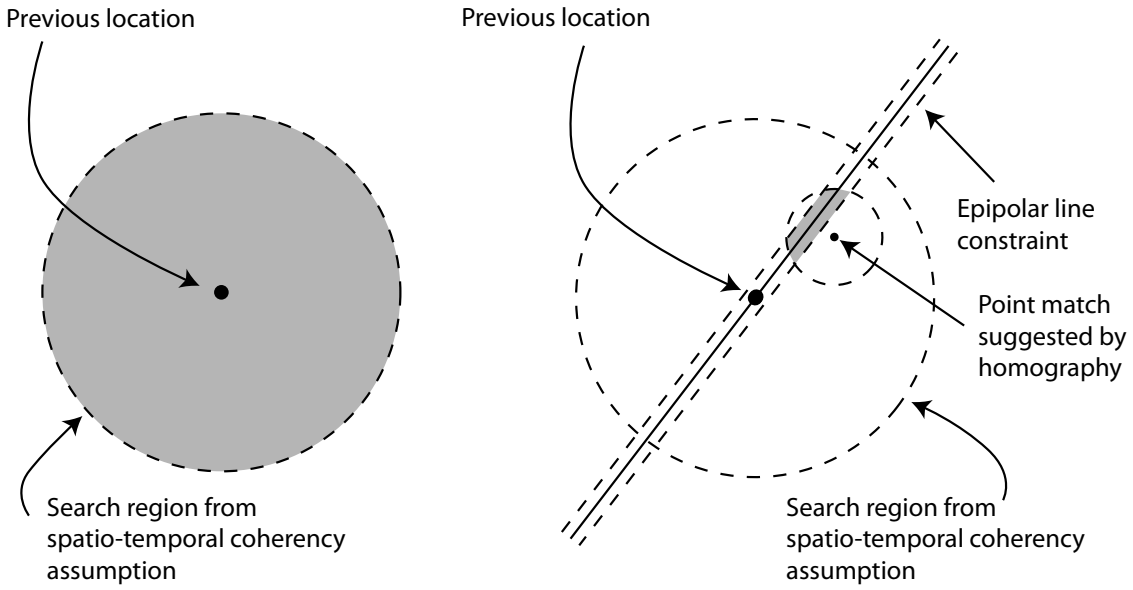


Figure 3.8. Guided matching search regions. Without any specific knowledge, the search region (shaded gray) for correspondences can only be limited by some weak assumption of spatio-temporal coherency (left). If the fundamental matrix is known, the search is restricted to the epipolar line (within some small tolerance); if an estimate of the homography is also known, then one obtains an approximate point match that defines a new circular search region. By intersecting all three constraints we obtain a much smaller search region.

The homography can be estimated nonlinearly using KLT with the projective photometric model, as shown in Fig. 3.7. However, the nonlinear KLT method is very sensitive and may fail to find the best homography for medium baseline pairs. In this case, a much more reliable method is to identify sparse correspondences and linearly solve for the homography from these point matches. We use the second approach. In order to find correspondences, we first down-sample the images for improved performance and then identify feature points in both images (Section 3.2). Feature points in the second image are then stored in a regular grid structure that supports amortized O(1) lookup of all feature points within a specified radius of some point (assuming uniform density of feature points).

For each feature point in the first image we query the regular grid for the list of feature points in the right image within the specified radius. For each candidate match, we compute a rotation insensitive matching score by evaluating the Normalized Cross Correlation (NCC) at all angles in 10° increments (the choice of increment is largely arbitrary and should be chosen based on computational demands as well as the size of the features, as larger features will benefit from finer discretization). We have found experimentally that this explicit search over rotation space is more robust than nonlinear minimization of the KLT using the affine photometric model (3.15). Matches that maximize the NCC, and have NCC greater than 0.95,

are taken as correspondences.

From this set of correspondences we compute a robust estimate of the fundamental matrix **F** using RANSAC, and then from the set of inliers we estimate the homography **H**. An appropriate search radius from the point match suggested by the homography is then determined adaptively by looking at the residual errors; the more motion parallax there is present, the larger the search radius will need to be. Specifically, we use a search radius that is the maximum of the 80th percentile of the residuals and a threshold to enforce a minimum search radius, because if the fundamental matrix was estimated from quasi-degenerate (i.e., mostly coplanar) data then we do not want to search only for more planar matches.

Finally, we repeat the above matching process on the full scale images to match the set of input features in the second image. From each set of candidate match points, we test only against the potential matches that are within both the epipolar threshold and automatically determined radius from the homography. The restricted search space improves performance because there are fewer potential matches that need to be tested, and also reduces the number of outliers because the probability of matching an outlier is reduced because the total number of potential matches has been reduced without eliminating the correct match.

Example correspondences found using this guided matching are shown in Fig. 3.9, Fig. 3.10, and Fig. 3.11. As can be seen from the figures, the feature point detection and matching is very robust, even under moderate changes in perspective.

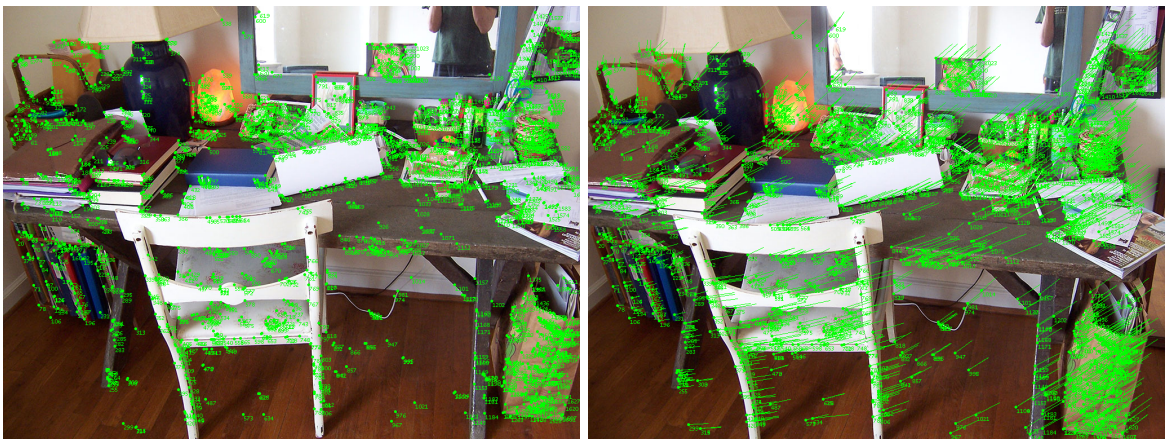


Figure 3.9. Example correspondences. There are 1451 correspondences and a fundamental matrix was fit with mean squared reprojection error of 0.14 pixels. (a) left image with numbered features; (b) right image with matched features and optical flow vectors.

Figure 3.10. Example correspondences. There are 1822 correspondences and a fundamental matrix was fit with mean squared reprojection error of 0.106 pixels. (a) left image with numbered features; (b) right image with matched features and optical flow vectors.

Figure 3.11. Example correspondences. There are 452 correspondences and a fundamental matrix was fit with mean squared reprojection error of 0.105 pixels. (a) left image with numbered features; (b) right image with matched features and optical flow vectors.

Chapter 4

Projective Triplet Reconstruction

In this chapter we discuss the issue of computing an accurate and reliable projective reconstruction from three views. It is well known that the trifocal tensor can be estimated either minimally from 6 points [Quan, 1995; Carlsson and Weinshall, 1998; Hartley and Debunne, 1998; Hartley and Dano, 2000] or linearly from 7 or more points [Hartley, 1995; Shashua and Werman, 1995; Hartley, 1998a]. The linear method is over-determined, which provides robustness to noise, but does not enforce internal constraints so the result is not geometrically consistent. In contrast, the minimal algorithm implicitly enforces all internal constraints and requires fewer points, which theoretically means fewer iterations will be required when used within a RANSAC framework for robust estimation.

The importance of using minimal methods within RANSAC has been stressed [Fischler and Bolles, 1981], and in particular it has been concluded that the 6 point method should be used when estimating the trifocal tensor [Torr and Zisserman, 1997; Hartley and Zisserman, 2004], with empirical results showing that the 6 point method produces substantially lower error [Torr, 1995; Torr and Zisserman, 1997]. However, it has been noticed that using a linear initial estimate from a non-minimal subset increases the robustness to noise [Chum et al., 2003], and this may lead to improved convergence characteristics.

More recently developed quasi-linear methods improve the performance of the linear method by enforcing internal constraints and were not considered in the previous studies. The purpose of this research was to determine whether or not the minimal or linear algorithm is better to use within RANSAC when state of the art techniques are employed; and, if the linear method is superior, then we also wanted to know which variation was most effective, and how many points to use for optimal performance.

We begin by introducing some basic mathematical background by showing how the tensor can be derived from corresponding line constraints in three images (Section 4.1) and how it relates to projection matrices (Section 4.1.1).

There is still no direct method for over-determined estimation of the tensor that takes into account all of the internal tensor constraints (Section 4.1.2), which are themselves still not fully understood. Therefore, we felt that this was a subject that merited further research. Although we were not able to find such an ideal algorithm, we have at least managed to derive the final three internal constraints (Section 4.1.2.5), and shown how these constraints can be used to define a new parameterization (Section 4.1.2.5.1).

We then discuss existing trifocal tensor estimation algorithms (Section 4.2), beginning with the minimal 6 point solution (Section 4.2.1), in which we introduce some minor tricks for improving robustness and disambiguiting between the multiple solutions. Next we introduce the basic linear method (Section 4.2.2), and discuss three alternative ways for enforcing the trilinear constraints (Section 4.2.2.1), as well as four methods for quasi-linear reestimation to enforce internal consistency constraints (Section 4.2.2.2). We also provide a discussion of additional estimation algorithms and explain why they were not included in our comparison (Section 4.2.3).

Out experiments (Section 4.4) begin with several tests designed to first find the best linear variation (Section 4.4.1) which we then compare to the minimal algorithm to see which has better performance (Section 4.4.2). Finally we investigate performance in RANSAC as a function of the number of points used, on both synthetic and real data (Section 4.4.3).

Our experimental results indicate several things: (a) we show that an older, lesser used, method of quasi-linear enforcement of the internal constraints actually performs best; (b) we could find no difference in performance between the various methods of trilinear constraint representation, which leads us to believe that it is best to stick with the simplest and fastest method; (c) we show that the best linear variation provides a substantially more accurate estimate than the minimal method, and is nearly a maximum likelihood estimate when estimated from more than 10 points; (d) contrary to popular belief, we show that using larger subset size in RANSAC is actually better because it allows a larger final consensus size to be reached, and in a shorter overall runtime, despite the fact that runtime for the minimal method by itself is substantially faster.

4.1 The Trifocal Tensor

The constraints on corresponding lines in three views were first derived for calibrated cameras in [Spetsakis and Aloimonos, 1990; Weng et al., 1992b]. These constraints were generalized to the uncalibrated case in [Shashua, 1995], and formulated in terms of a trifocal tensor in [Hartley, 1995; Shashua and Werman, 1995; Triggs, 1995]. It was shown in [Hartley, 1997b] that point constraints could also be represented using the tensor. In this section we summarize the derivation of the tensor from line constraints as described in [Hartley and Zisserman, 2004].

Without loss of generality, the first projection matrix can be assumed canonical, so that the set of projection matrices for three views can be written as

$$\mathbf{P} = [\mathtt{I}|\mathbf{0}] \tag{4.1}$$

$$\mathbf{P}' = [\mathbf{a}_1 \dots \mathbf{a}_4] = [\mathbf{A}|\mathbf{a}_4] \tag{4.2}$$

$$\mathbf{P}'' = [\mathbf{b}_1 \dots \mathbf{b}_4] = [\mathbf{B}|\mathbf{b}_4]. \tag{4.3}$$

The tensor will be derived based on a correspondence between images of a line in 3D space. Let the three corresponding lines in the image plane be denoted as $\mathbf{l} \leftrightarrow \mathbf{l}' \leftrightarrow \mathbf{l}''$. The back projection of each line yields a plane,

$$\pi = \mathbf{P}^\mathsf{T}\mathbf{l} = (\mathbf{l}^\mathsf{T}, 0)^\mathsf{T} \tag{4.4}$$

$$\pi' = \mathbf{P}'^\mathsf{T}\mathbf{l}' = \begin{bmatrix} \mathbf{A}^\mathsf{T}\mathbf{l}' \\ \mathbf{a}_4^\mathsf{T}\mathbf{l}' \end{bmatrix} \tag{4.5}$$

$$\pi'' = \mathbf{P}''^\mathsf{T}\mathbf{l}'' = \begin{bmatrix} \mathbf{B}^\mathsf{T}\mathbf{l}'' \\ \mathbf{b}_4^\mathsf{T}\mathbf{l}'' \end{bmatrix}. \tag{4.6}$$

Because the lines were all images of a single 3D line, these back-projected planes must all intersect in a single 3D line that we write parametrically as a linear combination of two points $\mathbf{X}_1$ and $\mathbf{X}_2$,

$$\mathbf{X}(t) = t\mathbf{X}_1 + (1-t)\mathbf{X}_2. \tag{4.7}$$

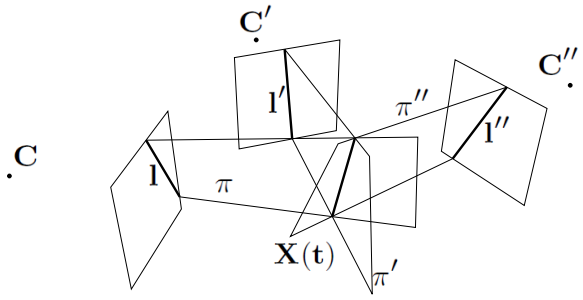


Figure 4.1. Diagram of trifocal line constraints. The first camera center is denoted by $\mathbf{C}$. A parametric 3D line in space is given by $X(t)$. This line projects onto the first image plane as $\mathbf{l}$. The line $\mathbf{l}$ back-projects to the plane π. Notation is similar with respect to the other two views.

This incidence relation is diagrammed in Fig. 4.1. Clearly, $\mathbf{X}(t)$ must be a point on each back-projected plane equation, so

$$\pi^{\mathsf{T}}\mathbf{X}(t) = \pi'^{\mathsf{T}}\mathbf{X}(t) = \pi''^{\mathsf{T}}\mathbf{X}(t) = 0. \tag{4.8}$$

If we concatenate these planes into a 4×3 matrix $\mathbf{M} = [\pi|\pi'|\pi'']$, then $\mathbf{M}^{\mathsf{T}}\mathbf{X}(t) = 0$. Substituting (4.4-4.6) into $\mathbf{M}$, we obtain

$$\mathbf{M} = \begin{bmatrix} \mathbf{l} & \mathbf{A}^{\mathsf{T}}\mathbf{l}' & \mathbf{B}^{\mathsf{T}}\mathbf{l}'' \\ 0 & \mathbf{a}_4^{\mathsf{T}}\mathbf{l}' & \mathbf{b}_4^{\mathsf{T}}\mathbf{l}'' \end{bmatrix}. \tag{4.9}$$

Because $\mathbf{M}^{\mathsf{T}}\mathbf{X}_1 = 0$ and $\mathbf{M}^{\mathsf{T}}\mathbf{X}_2 = 0$, $\mathbf{M}$ must have at least a 2-dimensional null space and is therefore at most rank 2 by the rank-nullity theorem. Thus, it follows that the first column can be written as a linear combination of the second two columns, so $\pi = \alpha\pi' + \beta\pi''$. From the bottom row we obtain

$$0 = \alpha\mathbf{a}_4^{\mathsf{T}}\mathbf{l}' + \beta\mathbf{b}_4^{\mathsf{T}}\mathbf{l}'', \tag{4.10}$$

which implies that $\alpha = k\mathbf{b}_4^{\mathsf{T}}\mathbf{l}''$ and $\beta = -k\mathbf{a}_4^{\mathsf{T}}\mathbf{l}'$ for some scalar k. Making these substitutions back into the top half of $\mathbf{M}$ provides a homogeneous equivalence constraint between the lines,

$$\mathbf{l} = \mathbf{b}_4^{\mathsf{T}}\mathbf{l}''\mathbf{A}^{\mathsf{T}}\mathbf{l}' - \mathbf{a}_4^{\mathsf{T}}\mathbf{l}'\mathbf{B}^{\mathsf{T}}\mathbf{l}'' \tag{4.11}$$

$$= \mathbf{l}''^{\mathsf{T}}\mathbf{b}_4\mathbf{A}^{\mathsf{T}}\mathbf{l}' - \mathbf{l}'^{\mathsf{T}}\mathbf{a}_4\mathbf{B}^{\mathsf{T}}\mathbf{l}''. \tag{4.12}$$

Introducing the notation $\mathbf{l} = (l_1, l_2, l_3)^{\mathsf{T}}$ and

$$\mathtt{T}_i = \mathbf{a}_i\mathbf{b}_4^{\mathsf{T}} - \mathbf{a}_4\mathbf{b}_i^{\mathsf{T}}, \tag{4.13}$$

it can be verified that (4.12) is equivalent to

$$l_i = \mathbf{l}'\mathtt{T}_i\mathbf{l}'' \; \forall i. \tag{4.14}$$

Thus, the relationship between cameras has been completely described by $\{\mathtt{T}_1, \mathtt{T}_2, \mathtt{T}_3\}$. These

three matrices, known as the *correlation slices*, can be represented by a single $3 \times 3 \times 3$ tensor $\mathcal{T}$, allowing the above relations to be written equivalently in tensor notation as

$$\mathcal{T}_i^{jk} = a_i^j b_4^k - a_4^j b_i^k \tag{4.15}$$

$$l_i = l'_j l''_k \mathcal{T}_j^{jk}. \tag{4.16}$$

It should be noted that, similar to the fundamental matrix, the views are treated asymmetrically by the trifocal tensor. In other words, there are three different trifocal tensors for any trio of views depending on the order in which the views are considered. In the remainder of this work, we assume an implicit ordering of these views.

4.1.1 Relationship to Projection Matrices

Because the trifocal tensor provides a complete description of the epipolar geometry for three views, it must be possible to extract a suitable set of projection matrices. However, it is not immediately obvious how one could factor a given tensor into the form of (4.13) to get back the original camera matrices. An algorithm is given in [Hartley and Zisserman, 2004, Alg. 15.1] and is summarized here.

One begins by calculating the epipoles $\mathbf{e}'$ and $\mathbf{e}''$, which are the images of the focal point of the first camera in the other two views. This is achieved in two steps. First, denote the left and right null spaces of each $\mathtt{T}_i$ as $\mathbf{v}_i$ and $\mathbf{u}_i$ in

$$\mathtt{T}_i \mathbf{v}_i = \mathbf{0}, \quad i = 1 \dots 3 \tag{4.17}$$

$$\mathtt{T}_i^\mathsf{T} \mathbf{u}_i = \mathbf{0}, \quad i = 1 \dots 3. \tag{4.18}$$

Next, denote $\mathbf{U} = [\mathbf{u}_1 | \mathbf{u}_2 | \mathbf{u}_3]^\mathsf{T}$ and $\mathbf{V} = [\mathbf{v}_1 | \mathbf{v}_2 | \mathbf{v}_3]^\mathsf{T}$. Then the epipoles are given by the null spaces of $\mathbf{U}$ and $\mathbf{V}$,

$$\mathbf{U}\mathbf{e}' = \mathbf{0} \tag{4.19}$$

$$\mathbf{V}\mathbf{e}'' = \mathbf{0}. \tag{4.20}$$

Once the epipoles have been determined, one can recover the fundamental matrix between the first two views. Recall that the tensor was defined based on a correspondence between lines $\mathbf{l} \leftrightarrow \mathbf{l}' \leftrightarrow \mathbf{l}''$ in each image. If the third line $\mathbf{l}''$ back projects into a plane π'', then this plane induces a planar-homography mapping the first line $\mathbf{l}$ to the second line $\mathbf{l}'$.

A homography that transfers points according to $\mathbf{x}' = \mathbf{H}\mathbf{x}$ transfers lines according to $\mathbf{l}' = \mathbf{H}^{-\mathsf{T}}\mathbf{l}$. According to this definition, (4.14) implies that the homography transferring a line from the first to the second image induced by a line in the third image is given by

$$\mathbf{H}_{12} = [\mathtt{T}_1, \mathtt{T}_2, \mathtt{T}_3]\mathbf{l}'', \tag{4.21}$$

where the notational convention of writing $\mathbf{A}[\mathbf{B}, \mathbf{C}, \mathbf{D}]\mathbf{E}$ is used as a shorthand for $[\mathbf{ABE}|\mathbf{ACE}|\mathbf{ADE}]$.

Given a point $\mathbf{x}$ in the first view, it is therefore transferred to $\mathbf{x}' = \mathbf{H}_{12}\mathbf{x}$ in the second view. The line between two points is given by the cross product, so the epipolar line $\mathbf{l}'_e$ corresponding to $\mathbf{x}$ is given by

$$\mathbf{l}'_e = \mathbf{e}' \times [\mathtt{T}_1, \mathtt{T}_2, \mathtt{T}_3]\mathbf{l}''\mathbf{x}. \tag{4.22}$$

Thus, the fundamental matrix $\mathbf{F}_{12}$ from the first to the second view is given by

$$\mathbf{F}_{12} = [\mathbf{e}']_\times [\mathtt{T}_1, \mathtt{T}_2, \mathtt{T}_3]\mathbf{l}''. \tag{4.23}$$

This formula holds for any $\mathbf{l}''$ as long as $\mathbf{l}''$ is not in the null space of any $\mathbf{T}_i$. One choice that avoids this degeneracy is $\mathbf{e}''$. Thus, one obtains

$$\mathbf{F}_{12} = [\mathbf{e}']_\times [\mathtt{T}_1, \mathtt{T}_2, \mathtt{T}_3]\mathbf{e}''. \tag{4.24}$$

It is known that the fundamental matrix corresponding to a pair of cameras given by $\mathbf{P} = [\mathtt{I}|\mathbf{0}]$ and $\mathbf{P}' = [\mathbf{M}|\mathbf{m}]$ is equal to $[\mathbf{m}]_\times\mathbf{M}$. Therefore, a suitable choice for the first two camera matrices consistent with the tensor is given by

$$\mathbf{P} = [\mathtt{I}|\mathbf{0}] \tag{4.25}$$

$$\mathbf{P}' = [[\mathtt{T}_1, \mathtt{T}_2, \mathtt{T}_3]\mathbf{e}''|\mathbf{e}']. \tag{4.26}$$

The third camera matrix can now be determined from (4.13). Using the notation of (4.3),

$$\mathbf{a}_i = \mathtt{T}_i \mathbf{e}'', \quad i = 1 \ldots 3 \tag{4.27}$$

$$\mathbf{a}_4 = \mathbf{e}' \tag{4.28}$$

$$\mathbf{b}_4 = \mathbf{e}'' \tag{4.29}$$

and substituting into (4.13) we obtain

$$\mathtt{T}_i = \mathtt{T}_i \mathbf{e}'' \mathbf{e}''^{\mathsf{T}} - \mathbf{e}' \mathbf{b}_i^{\mathsf{T}} \tag{4.30}$$

$$\mathbf{e}' \mathbf{b}_i^{\mathsf{T}} = \mathtt{T}_i (\mathbf{e}'' \mathbf{e}''^{\mathsf{T}} - \mathtt{I}). \tag{4.31}$$

If we choose the scale of $\mathbf{e}'$ such that $\mathbf{e}'^{\mathsf{T}} \mathbf{e}' = ||\mathbf{e}''|| = 1$, then we can left multiply by $\mathbf{e}'^{\mathsf{T}}$ to get

$$\mathbf{b}_i^{\mathsf{T}} = \mathbf{e}'^{\mathsf{T}} \mathtt{T}_i (\mathbf{e}'' \mathbf{e}''^{\mathsf{T}} - \mathtt{I}) \tag{4.32}$$

$$\mathbf{b}_i = (\mathbf{e}'' \mathbf{e}''^{\mathsf{T}} - \mathtt{I}) \mathtt{T}_i^{\mathsf{T}} \mathbf{e}'. \tag{4.33}$$

Thus, a consistent choice for the third camera matrix is given by

$$\mathbf{P}'' = [(\mathbf{e}'' \mathbf{e}''^{\mathsf{T}} - \mathtt{I})[\mathtt{T}_1^{\mathsf{T}}, \mathtt{T}_2^{\mathsf{T}}, \mathtt{T}_3^{\mathsf{T}}] \mathbf{e}' | \mathbf{e}'']. \tag{4.34}$$

4.1.2 Internal Tensor Constraints

A reconstruction from projection constraints alone is, at best, ambiguous up to an arbitrary projective transform having 15 degrees of freedom (dof). Each camera matrix has 11 dof, so there are $11m - 15$ dof to the projective-invariant geometry (henceforth referred to as *epipolar geometry*) of m unique cameras [Hartley and Zisserman, 2004, sec. 17.5].

In the case of 2 views, the epipolar geometry has 7 dof and may be conveniently represented by the 3×3 fundamental matrix $\mathbf{F}$. The additional 2 parameters in the matrix $\mathbf{F}$ may be attributed to an ambiguous overall scale factor and a single internal constraint that $\operatorname{rank} \mathbf{F} = 2$. This is usually relaxed to $\operatorname{rank} \mathbf{F} \leq 2$, making it equivalent to $\det \mathbf{F} = 0$, which can be written as a simple polynomial.

Similarly, we have seen in Section 4.1 that the geometric relationship between three views can be conveniently represented by the $3 \times 3 \times 3$ trifocal tensor, denoted by $\mathcal{T}$. Again by the argument above, it is clear that the epipolar geometry of three views has 18 dof, although

the tensor $\mathcal{T}$ has 27 parameters. Thus, not all tensors are a consistent representation of some epipolar geometry. One parameter may be attributed to the overall scale factor, meaning that a geometrically meaningful tensor must satisfy 8 independent algebraic constraints.

Unlike the fundamental matrix, the trifocal constraints are not straightforward and have not previously been fully understood [Faugeras and Mourrain, 1995; Laveau, 1996; Torr and Zisserman, 1997] [Hartley and Zisserman, 2004, sec. 15.1]. The earliest known constraints are the three rank constraints (3rd order) and two epipolar constraints (5th order) [Hartley, 1997b; Faugeras and Papadopoulo, 1998; Papadopoulo and Faugeras, 1998], which are fairly straightforward to identify from the definition (Section 4.1.2.1). However, this set of five constraints is insufficient because they leave three additional degrees of freedom unaccounted for.

An additional 27 axes constraints (Section 4.1.2.2) of 6th order were discovered in Faugeras and Papadopoulo [1998]. These axes constraints are not fully independent from one another, although they are independent from the previous rank and epipolar constraints. Thus, all 32 constraints were needed to constrain the tensor at that time.

It was later shown in Papadopoulo and Faugeras [1998] that there are a set of 10 extended rank constraints (Section 4.1.2.3), three of which are equivalent to the original rank constraints. These are 3rd order, and independent from the epipolar constraints. Thus, the epipolar constraints can be taken with the extended rank constraints to yield a set of 12 sufficient constraints.

Finally, Canterakis [2000] discovered a set of 8 constraints (of up to 12th degree) based on the concept of generalized eigenvalues (Section 4.1.2.4). This is currently the only minimal and sufficient set of constraints that is known. However, a somewhat unsatisfactory property of these constraints is that they are not so simple as the well known rank and epipolar constraints.

From the counting argument, it is clear that there must exist a set of 3 constraints that can be taken with the well-known rank and epipolar constraints to yield an alternative set of constraints that is both minimal and sufficient. This set would likely be simpler and preferable for the purpose of constrained estimation than the generalized eigenspace constraints which are rather cumbersome to deal with. The identification of these final constraints has remained an open problem until now (Section 4.1.2.5).

4.1.2.1 Rank and Epipolar Constraints

The rank and epipolar constraints are a set of 5 independent constraints on the tensor that have been known for some time [Hartley, 1997b; Faugeras and Papadopoulo, 1998; Papadopoulo and Faugeras, 1998], and can be deduced from (4.17-4.34). Specifically, in order for the null spaces in (4.17-4.18) to exist, each $\mathtt{T}_i$ must be of rank 2. This may also be deduced from the fact that $\mathtt{T}_i$ was defined as the sum of two outer products in (4.13). Thus, the first three constraints may be taken as,

$$\text{rank } \mathtt{T}_i = 2, \quad i = 1 \ldots 3. \tag{4.35}$$

These are the *rank constraints*. They arise as a direct result of the assumption that corresponding lines must back-project into planes that intersect in a common 3D line.

Similarly, in order for the epipoles to exist as null spaces in (4.19-4.20), the rank of the matrix of left null vectors, as well as the rank of the matrix of right null vectors, must be 2. Thus, an additional two constraints, called the *epipolar constraints*, are given by

$$\text{rank } \mathbf{U} = 2 \tag{4.36}$$

$$\text{rank } \mathbf{V} = 2. \tag{4.37}$$

These epipolar constraints enforce the property that all projection rays emanating from one camera's focal point must intersect at a common epipolar point in the image plane of another camera.

Constraints of the form $\text{rank}\,\mathbf{M} = 2$ on 3×3 matrices are usually treated as being equivalent to $\det \mathbf{M} = 0$ to be enforced as polynomials. However, it should be noted that degenerate matrices having rank < 2 do not represent valid trifocal tensors, although the inequality constraint that rank > 1 does not affect the dof.

4.1.2.2 Axes Constraints

There are 27 axes constraints [Faugeras and Papadopoulo, 1998; Faugeras et al., 2001], 9 for each dimension of the tensor. These are 6th degree polynomials in the tensor elements, independent from the rank and epipolar constraints. Recall that the trifocal tensor elements are indexed as $\mathcal{T}_i^{jk}$ where i, j, k index the correlation slice, row, and column dimensions, respectively.

In order to denote these constraints, we introduce the notation that an asterisk in any dimension means all the values in that dimension will be collected into a *column* vector. In other words, $\mathcal{T}_i^{*k}$ denotes the kth column of the ith correlation slice $\mathtt{T}_i$, $\mathcal{T}_i^{j*}$ denotes the *transpose* of the jth row of $\mathtt{T}_i$, and $\mathcal{T}_*^{jk}$ denotes the column vector formed by taking the (j,k)th element from each correlation slice.

Without further ado, the *vertical constraints* are

$$\left|\mathcal{T}_*^{ik}, \mathcal{T}_*^{il}, \mathcal{T}_*^{jl}\right| \left|\mathcal{T}_*^{ik}, \mathcal{T}_*^{jk}, \mathcal{T}_*^{jl}\right| - \tag{4.38}$$

$$\left|\mathcal{T}_*^{jk}, \mathcal{T}_*^{il}, \mathcal{T}_*^{jl}\right| \left|\mathcal{T}_*^{ik}, \mathcal{T}_*^{jk}, \mathcal{T}_*^{il}\right| = 0, \tag{4.39}$$

the *horizontal column* constraints are

$$\left|\mathcal{T}_i^{*k}, \mathcal{T}_i^{*l}, \mathcal{T}_j^{*l}\right| \left|\mathcal{T}_i^{*k}, \mathcal{T}_j^{*k}, \mathcal{T}_j^{*l}\right| - \tag{4.40}$$

$$\left|\mathcal{T}_j^{*k}, \mathcal{T}_i^{*l}, \mathcal{T}_j^{*l}\right| \left|\mathcal{T}_i^{*k}, \mathcal{T}_j^{*k}, \mathcal{T}_i^{*l}\right| = 0, \tag{4.41}$$

and the *horizontal row* constraints are

$$\left|\mathcal{T}_i^{k*}, \mathcal{T}_i^{l*}, \mathcal{T}_j^{l*}\right| \left|\mathcal{T}_i^{k*}, \mathcal{T}_j^{k*}, \mathcal{T}_j^{l*}\right| - \tag{4.42}$$

$$\left|\mathcal{T}_j^{k*}, \mathcal{T}_i^{l*}, \mathcal{T}_j^{l*}\right| \left|\mathcal{T}_i^{k*}, \mathcal{T}_j^{k*}, \mathcal{T}_i^{l*}\right| = 0, \tag{4.43}$$

where $i, j, k, l \in \{1, 2, 3\}$ and $i < j$ and $k < l$. Note that commas have been used in the above equations to denote the merging of three columns together to form a 3×3 matrix.

4.1.2.3 Extended Rank Constraints

It is known that any linear combination of the tensor slices must also have rank 2. Specifically, if $\mathbf{x} = (x_1, x_2, x_3)^\mathsf{T}$ then

$$\text{rank} \sum_i x_i \mathtt{T}_i = 2. \tag{4.44}$$

Geometrically, if $\mathbf{x}$ is a point in the first image then the left and right null spaces of $\sum_i x_i \mathtt{T}_i$ are the corresponding epipolar lines in the second and third views, respectively [Hartley and Zisserman, 2004].

The requirement that all linear combinations have rank 2 can be translated into a set of 10 algebraic constraints called the *extended rank constraints* [Papadopoulo and Faugeras, 1998; Faugeras et al., 2001] as follows. The rank constraint in (4.44) implies a zero determinant, which can be expanded to the polynomial

$$\begin{aligned}\det \sum_i x_i \mathtt{T}_i = &c_1 x_1^3 + c_2 x_2^3 + c_3 x_3^3 \\ &+ c_4 x_1^2 x_2 + c_5 x_1^2 x_3 \\ &+ c_6 x_2^2 x_1 + c_7 x_2^2 x_3 \\ &+ c_8 x_3^2 x_1 + c_9 x_3^2 x_2 \\ &+ c_{10} x_1 x_2 x_3 = 0.\end{aligned} \tag{4.45}$$

In order for this polynomial to be zero for *all* choices of $\mathbf{x}$, it must be the case that the coefficients $c_i = 0 \, \forall i$. These are the ten extended rank constraints. The first three coefficients are $c_i = \det \mathtt{T}_i$ for $i = 1 \ldots 3$, so these are just the original rank constraints, but the remaining 7 are independent from the basic rank and epipolar constraints. They may be expanded to

$$c_4 = \left|\mathcal{T}_1^{*1}, \mathcal{T}_1^{*2}, \mathcal{T}_2^{*3}\right| + \left|\mathcal{T}_1^{*1}, \mathcal{T}_2^{*2}, \mathcal{T}_1^{*3}\right| + \left|\mathcal{T}_2^{*1}, \mathcal{T}_1^{*2}, \mathcal{T}_1^{*3}\right| = 0 \tag{4.46}$$

$$c_5 = \left|\mathcal{T}_1^{*1}, \mathcal{T}_1^{*2}, \mathcal{T}_3^{*3}\right| + \left|\mathcal{T}_1^{*1}, \mathcal{T}_3^{*2}, \mathcal{T}_1^{*3}\right| + \left|\mathcal{T}_3^{*1}, \mathcal{T}_1^{*2}, \mathcal{T}_1^{*3}\right| = 0 \tag{4.47}$$

$$c_6 = \left|\mathcal{T}_2^{*1}, \mathcal{T}_2^{*2}, \mathcal{T}_1^{*3}\right| + \left|\mathcal{T}_2^{*1}, \mathcal{T}_1^{*2}, \mathcal{T}_2^{*3}\right| + \left|\mathcal{T}_1^{*1}, \mathcal{T}_2^{*2}, \mathcal{T}_2^{*3}\right| = 0 \tag{4.48}$$

$$c_7 = \left|\mathcal{T}_2^{*1}, \mathcal{T}_2^{*2}, \mathcal{T}_3^{*3}\right| + \left|\mathcal{T}_2^{*1}, \mathcal{T}_3^{*2}, \mathcal{T}_2^{*3}\right| + \left|\mathcal{T}_3^{*1}, \mathcal{T}_2^{*2}, \mathcal{T}_2^{*3}\right| = 0 \tag{4.49}$$

$$c_8 = \left|\mathcal{T}_3^{*1}, \mathcal{T}_3^{*2}, \mathcal{T}_1^{*3}\right| + \left|\mathcal{T}_3^{*1}, \mathcal{T}_1^{*2}, \mathcal{T}_3^{*3}\right| + \left|\mathcal{T}_1^{*1}, \mathcal{T}_3^{*2}, \mathcal{T}_3^{*3}\right| = 0 \tag{4.50}$$

$$c_9 = \left|\mathcal{T}_3^{*1}, \mathcal{T}_3^{*2}, \mathcal{T}_2^{*3}\right| + \left|\mathcal{T}_3^{*1}, \mathcal{T}_2^{*2}, \mathcal{T}_3^{*3}\right| + \left|\mathcal{T}_2^{*1}, \mathcal{T}_3^{*2}, \mathcal{T}_3^{*3}\right| = 0 \tag{4.51}$$

$$\begin{aligned}c_{10} = &\left|\mathcal{T}_1^{*1}, \mathcal{T}_2^{*2}, \mathcal{T}_3^{*3}\right| + \left|\mathcal{T}_1^{*1}, \mathcal{T}_3^{*2}, \mathcal{T}_2^{*3}\right| + \left|\mathcal{T}_2^{*1}, \mathcal{T}_1^{*2}, \mathcal{T}_3^{*3}\right| + \\ &\left|\mathcal{T}_2^{*1}, \mathcal{T}_3^{*2}, \mathcal{T}_1^{*3}\right| + \left|\mathcal{T}_3^{*1}, \mathcal{T}_1^{*2}, \mathcal{T}_2^{*3}\right| + \left|\mathcal{T}_3^{*1}, \mathcal{T}_2^{*2}, \mathcal{T}_1^{*3}\right| = 0.\end{aligned} \tag{4.52}$$

4.1.2.4 Generalized Eigenspace Constraints

It was shown in [Canterakis, 2000] that a minimal set of 8 necessary and sufficient constraints for a trifocal tensor could be derived by ensuring that the generalized eigenspaces between each pair of tensor slices $\mathtt{T}_i$ intersect in a common point, and that there exists a common one-dimensional generalized eigenspace of all pairs. Specifically, the following conditions must be satisfied:

1. The polynomial $\det(\mathtt{T}_2 - \lambda \mathtt{T}_1)$ must have a single root λ_1 and double root λ_2, and $\operatorname{rank}(\mathtt{T}_2 - \lambda_2 \mathtt{T}_1) = 1$.

2. The polynomial $\det(\mathtt{T}_3 - \mu \mathtt{T}_1)$ must have a single root μ_1 and a double root μ_2, and $\operatorname{rank}(\mathtt{T}_3 - \mu_2 \mathtt{T}_1) = 1$.

3. If $\mathbf{a}, \mathbf{b}, \mathbf{a}', \mathbf{b}'$ are the generalized eigenvectors corresponding to the eigenvalues λ_1, λ_2, μ_1 and μ_2 (respectively), then $\mathbf{a} \propto \mathbf{a}'$.

These constraints are not independent from any of the previous constraints. The requirement that $\mathbf{a} \propto \mathbf{a}'$ amounts to checking for equality between two inhomogeneous 2-vectors and therefore provides two algebraic constraints.

In order for a 3rd degree polynomial

$$p(\lambda) = a\lambda^3 + b\lambda^2 + c\lambda + d \tag{4.53}$$

to have a double root, it must be the case that

$$B^2 - 4AC = 0, \tag{4.54}$$

where $A = b^2 - 3ac$, $B = bc - 9ad$ and $C = c^2 - 3bd$. In this case, the roots are given by

$$\lambda_1 = B/A - b/a \tag{4.55}$$
$$\lambda_2 = -B/(2A). \tag{4.56}$$

The two conditions of this type therefore provide two constraints. In terms of the tensor elements, the coefficients of $\det(\mathtt{T}_2 - \lambda\mathtt{T}_1)$ are 3rd degree, A, B, C are 6th degree, and the double root constraint is therefore 12th degree.

The constraint $\mathrm{rank}(\mathtt{T}_2 - \lambda_2\mathtt{T}_1) = 1$ is equivalent to

$$(\mathtt{T}_2 - \lambda_2\mathtt{T}_1)(\mathbf{a} \times \mathbf{b}) \propto \mathtt{T}_1\mathbf{a}, \tag{4.57}$$

(and similarly for the second pair). These requirements provide the final four constraints.

4.1.2.5 Circular Constraints

The final 3 constraints can be obtained by substituting the recovered camera matrices from (4.26) and (4.34) back into (4.13). This may at first seem like circular logic because the camera matrices were derived from (4.13). However, the presence of singular outer-product matrices prevents the resulting equations from being simplified down to a trivial result such as $\mathtt{I} = \mathtt{I}$, and real constraints arise. We will refer to these as the *circular constraints*.

In Section 4.1.1 we have already derived

$$\mathbf{a}_i = \mathtt{T}_i \mathbf{e}'', \quad i = 1 \ldots 3 \tag{4.58}$$

$$\mathbf{a}_4 = \mathbf{e}' \tag{4.59}$$

$$\mathbf{b}_i^{\mathsf{T}} = \mathbf{e}'^{\mathsf{T}} \mathtt{T}_i (\mathbf{e}'' \mathbf{e}''^{\mathsf{T}} - \mathtt{I}), \quad i = 1 \ldots 3 \tag{4.60}$$

$$\mathbf{b}_4 = \mathbf{e}'', \tag{4.61}$$

which can be substituted back into (4.13) to obtain

$$\mathtt{T}_i = \mathtt{T}_i \mathbf{e}'' \mathbf{e}''^{\mathsf{T}} - \mathbf{e}' (\mathbf{e}'^{\mathsf{T}} \mathtt{T}_i (\mathbf{e}'' \mathbf{e}''^{\mathsf{T}} - \mathtt{I})) \tag{4.62}$$

$$0 = \mathtt{T}_i (\mathbf{e}'' \mathbf{e}''^{\mathsf{T}} - \mathtt{I}) - \mathbf{e}' \mathbf{e}'^{\mathsf{T}} \mathtt{T}_i (\mathbf{e}'' \mathbf{e}''^{\mathsf{T}} - \mathtt{I}) \tag{4.63}$$

$$0 = (\mathtt{I} - \mathbf{e}' \mathbf{e}'^{\mathsf{T}}) \mathtt{T}_i (\mathbf{e}'' \mathbf{e}''^{\mathsf{T}} - \mathtt{I}). \tag{4.64}$$

Because outer product matrices and $\mathtt{T}_i$ are all singular, nothing further can be canceled out. However, one must be careful because this result was derived under the assumption that the epipoles were normalized, and only holds under that assumption. Thus, it may be generalized to

$$0 = (||\mathbf{e}'||^2 \mathtt{I} - \mathbf{e}' \mathbf{e}'^{\mathsf{T}}) \mathtt{T}_i (\mathbf{e}'' \mathbf{e}''^{\mathsf{T}} - ||\mathbf{e}''||^2 \mathtt{I}). \tag{4.65}$$

Because (4.65) is a 3×3 matrix equation which holds for each choice of $i = 1 \ldots 3$, it provides a total of 27 constraints that must be satisfied by $\mathcal{T}$ in order to remain consistent with our definitions. We will first explain our findings pertaining to the independence of these constraints.

To begin with, we denote the individual constraint equations arising from the matrix equation (4.65) as $\mathcal{C}_i^{jk}$, $ijk \in \{1, 2, 3\}$ corresponding to the (j, k)th equality using $\mathtt{T}_i$. Our first finding is that these constraints are not trivially satisfied, and that they are independent from the rank and epipolar constraints. Secondly, constraints on $\mathtt{T}_i$ are independent from constraints on $\mathtt{T}_j$ for $i \neq j$. Thus, from the counting argument, it can be inferred that if any constraint $\mathcal{C}_i^{jk}$ is satisfied, then $\mathcal{C}_i^{jk}$ $\forall jk$ are satisfied. These two findings are written formally as

$$\left(\begin{array}{l}\det\ \mathbf{U}=0\wedge\\ \det\ \mathbf{V}=0\wedge\\ \det\ \mathtt{T}_i=0\,\forall i\end{array}\right)\ \not\Rightarrow\ \left(\mathcal{C}_i^{jk}=0\right)\quad\forall ijk \tag{4.66}$$

$$\left(\mathcal{C}_i^{jk}=0\right)\ \Rightarrow\ \left(\mathcal{C}_i^{jk}=0\,\forall jk\right)\quad\forall i, \tag{4.67}$$

Therefore, one choice of the final three independent constraints may be taken as $\mathcal{C}_i^{22}\ \forall i$. Assuming the epipoles have been normalized, $\mathcal{C}_i^{22}$ expands to

$$\begin{array}{c}\mathbf{e}'_1\mathbf{e}'_2\mathbf{e}''_1\mathbf{e}''_2\mathcal{T}_i^{11}+\mathbf{e}'_1\mathbf{e}'_2(\mathbf{e}''^2_2-1)\mathcal{T}_i^{12}+\mathbf{e}'_1\mathbf{e}'_2\mathbf{e}''_2\mathbf{e}''_3\mathcal{T}_i^{13}+\\ \mathbf{e}''_1\mathbf{e}''_2(\mathbf{e}'^2_2-1)\mathcal{T}_i^{21}+(1-\mathbf{e}''^2_2)(1-\mathbf{e}'^2_2)\mathcal{T}_i^{22}+\mathbf{e}''_2\mathbf{e}''_3(\mathbf{e}'^2_2-1)\mathcal{T}_i^{23}+\\ \mathbf{e}'_2\mathbf{e}'_3\mathbf{e}''_1\mathbf{e}''_2\mathcal{T}_i^{31}+\mathbf{e}'_2\mathbf{e}'_3(\mathbf{e}''^2_2-1)\mathcal{T}_i^{32}+\mathbf{e}'_2\mathbf{e}'_3\mathbf{e}''_2\mathbf{e}''_3\mathcal{T}_i^{33}=0,\\ i\in\{1,2,3\},\end{array} \tag{4.68}$$

where the elements of the first epipole are denoted by $\mathbf{e}'=(\mathbf{e}'_1,\mathbf{e}'_2,\mathbf{e}'_3)^\mathsf{T}$, and similarly for $\mathbf{e}''$. This brings us to the following theorem,

Theorem 1. *Let $\mathcal{T}$ be any $3\times3\times3$ tensor. $\mathcal{T}$ is a trifocal tensor iff the following 8 internal constraints are satisfied:*

- *The three rank constraints of (4.35).*
- *The two epipolar constraints of (4.36-4.37).*
- *The three circular constraints of (4.68).*

Proof. Our derivation of the circular constraints given in (4.65) shows that these 27 equalities must be true for internal consistency, although we have not yet proven our claims in (4.66-4.67). In other words, we have not yet proven that these equalities are not implied by the rank and epipolar constraints, and that the subset in (4.68) are mutually independent.

If we assume that the circular constraints are dependent on the rank and/or epipolar constraints and then find a tensor that satisfies all the rank and epipolar constraints without satisfying the circular constraints, then we have reached a contradiction. Thus, finding such a tensor would prove that the circular constraints are, in general, independent from the rank and epipolar constraints.

It is easy to find an unlimited number of counter-examples of this type by using our parameterization in Section 4.1.2.5.1 to generate a random tensor that satisfies all constraints *except* for the circular constraint by simply omitting the last row from $\mathbf{C}_t$. It can then be verified

that the circular constraints are not satisfied. For example, we have used Maple with rational arithmetic to find the following tensor:

$$\mathtt{T}_1 = \begin{bmatrix} 357500/180469 & 200/251 & 475/251 \\ 1500/719 & 0 & 3 \\ 1700/719 & 2 & 1 \end{bmatrix} \tag{4.69}$$

$$\mathtt{T}_2 = \begin{bmatrix} 2050000/961197 & 200/401 & 1100/401 \\ 8000/2397 & 1 & 4 \\ 1500/799 & 0 & 3 \end{bmatrix} \tag{4.70}$$

$$\mathtt{T}_3 = \begin{bmatrix} 950000/480799 & 400/401 & 1100/401 \\ 2500/1199 & 0 & 5 \\ 4500/1199 & 4 & 1 \end{bmatrix}. \tag{4.71}$$

It may be verified that the above tensor slices are rank 2 (rank constraints). Extracting their null spaces and arranging them to form $\mathbf{U}$ and $\mathbf{V}$, we obtain

$$\mathbf{U} = \begin{bmatrix} -251/100 & 5/4 & 1 \\ -401/100 & 2 & 1 \\ -401/100 & 2 & 1 \end{bmatrix} \tag{4.72}$$

$$\mathbf{V} = \begin{bmatrix} -719/500 & 6/5 & 1 \\ -799/500 & 4/3 & 1 \\ -1199/500 & 2 & 1 \end{bmatrix}. \tag{4.73}$$

Again, these matrices are exactly rank 2 (epipolar constraints). Extracting their null spaces yields the epipoles,

$$\mathbf{e}' = (100, 200, 1)^{\mathsf{T}} \tag{4.74}$$

$$\mathbf{e}'' = (-500, -600, 1)^{\mathsf{T}}. \tag{4.75}$$

Finally, we evaluate (4.65) and observe that none of the circular constraints are zero. For brevity, we print only the values of the central constraints,

$$\mathcal{C}_1^{22} = -101022670792200/1834807869906823 \tag{4.76}$$

$$\mathcal{C}_2^{22} = -5236581973887/55211191885087 \tag{4.77}$$

$$\mathcal{C}_3^{22} = -14516209041800/698318420372419. \tag{4.78}$$

Thus, the tensor in (4.69-4.71) is not a valid trifocal tensor, and the circular constraints are indeed independent from the rank and epipolar constraints.

Proving that the three central constraints are independent from one another is trivial from the design of our algorithm for consistently parameterizing the tensor outlined in Section 4.1.2.5.1. Specifically, one notices that once $\mathbf{U}$ and $\mathbf{V}$ have been calculated, there are no further dependencies between the $\mathtt{T}_i$ matrices. Thus, it is not possible for $\mathcal{C}_i^{22}$ to have any dependency on $\mathcal{C}_j^{22}$ for $i \neq j$.

Having proven that $\mathcal{C}_i^{22}\ \forall i$ are mutually independent, and also independent from the rank and epipolar constraints, it can be seen from the counting argument that all degrees of freedom of the tensor have been accounted for. This proves our claim in (4.67). Indeed, if the circular constraint is added back into $\mathbf{C}_t$ in this example, then it can be verified that all of the constraints in (4.65) are satisfied exactly. □

To summarize, there are now four known sets of sufficient constraints that may be used to define a trifocal tensor, two of which are minimal. In order of discovery, they are

1. 3 rank + 2 epipolar + 27 axes
2. 2 epipolar + 10 extended rank
3. (minimal) 8 generalized eigenvalue
4. (minimal) 3 rank + 2 epipolar + 3 circular.

The newly found circular constraints were derived by a circular substituion; from the definition of the tensor, the general form for camera matrices was calculated and then substituted back into the definition of the tensor. This resulted in constraint equations that were nontrivially satisfied only because of the existence of rank 1 outer product matrices that prevented the equation from being simplified down to $\mathtt{I} = \mathtt{I}$.

Thus, the new constraints do not particularly represent any new geometrical restrictions, but are simply another result of algebraic consistency required from the original constraint that corresponding lines must back-project into planes that intersect in a common line in 3D space.

4.1.2.5.1 Circular Parameterization

Having identified the final circular constraints, it becomes possible to directly solve for the basis vectors of the components of a geometrically valid trifocal tensor, suggesting a mathematically elegant, although not necessarily practical, parameterization for the tensor by simply using the coordinates in these bases.

Specifically, four parameters can be used to describe the inhomogeneous coordinates of the epipoles $\mathbf{e}'$ and $\mathbf{e}''$. The epipolar constraint demands that the null spaces of $\mathbf{U}$ and $\mathbf{V}$ be the epipoles. Thus, if the elements of $\mathbf{U}$ are arranged into the column vector $\mathbf{u}$, then it must satisfy

$$\mathbf{C}_u\mathbf{u} = \mathbf{0}, \tag{4.79}$$

where $\mathbf{C}_u$ is a 3×9 constraint matrix given by

$$\mathbf{C}_u = \begin{bmatrix} \mathbf{e}'^{\mathsf{T}} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{e}'^{\mathsf{T}} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{e}'^{\mathsf{T}} \end{bmatrix}. \tag{4.80}$$

The space of all $\mathbf{u}$ satisfying this constraint is given by finding a basis $\mathbf{B}_u$ for the null space of $\mathbf{C}_u$ using Gauss-Jordan elimination. Because there are 9 parameters and 3 constraints, there is a 6 dimensional basis for the null space. One must be careful, however, to avoid the singular condition that arises if $\mathbf{U}$ has rank 1. This can be enforced by ensuring that each coordinate in the basis is non-zero. Also, the overall scale is irrelevant, so the last coordinate can be assumed equal to 1 and therefore $\mathbf{U}$ can be represented with 5 parameters (and similarly for $\mathbf{V}$).

The rank constraint demands that the left and right null spaces of each $\mathtt{T}_i$ are defined by $\mathbf{u}_i$ and $\mathbf{v}_i$. This provides six constraints on each $\mathtt{T}_i$. A seventh constraint is given by the circular constraint $\mathcal{C}_i^{22}$, and once again, the space of matrices satisfying these constraints is described by a basis for the null space of the constraint matrix.

To be precise, if the desired left and right null spaces of the slice $\mathtt{T}_i$ are $\mathbf{u}_i = (u_1, u_2, u_3)^{\mathsf{T}}$ and $\mathbf{v}_i = (v_1, v_2, v_3)^{\mathsf{T}}$, then the 7×9 constraint matrix is

$$\mathbf{C}_t = \begin{bmatrix} u_1 & 0 & 0 & u_2 & 0 & 0 & u_3 & 0 & 0 \\ 0 & u_1 & 0 & 0 & u_2 & 0 & 0 & u_3 & 0 \\ 0 & 0 & u_1 & 0 & 0 & u_2 & 0 & 0 & u_3 \\ v_1 & v_2 & v_3 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & v_1 & v_2 & v_3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & v_1 & v_2 & v_3 \\ a & b & c & d & e & f & g & h & i \end{bmatrix}, \tag{4.81}$$

where $a, b, c, \ldots, i$ are the coefficients of (4.68).

Although there are 7 constraints, there is a linear dependency between the first 6 so the dimension of the null space is 3. The first slice $\mathtt{T}_1$ can be represented with 2 parameters due to the overall scale ambiguity, but the remaining two slices require 3 parameters because they cannot be scaled independently. Thus, a total of 22 parameters are used to represent the tensor under this parameterization.

We now describe the reverse mapping for representing a given tensor in this parameterization. First, the epipoles are computed from the null spaces as shown in (4.17-4.20). If the tensor is not perfectly consistent, these null spaces may not exist so they should be extracted using the right singular vector, which provides a least squares estimate of a basis for the null space.

Once the epipoles have been found, the first four parameters are given by their inhomogeneous coordinates. Then the constraint matrix $\mathbf{C}_u$ can be formed and the basis $\mathbf{B}_u$ can be found. The coordinates $\mathbf{p}_u$ of $\mathbf{U}$ with respect to this basis may be found by solving a linear least squares system,

$$\mathbf{B}_u \mathbf{p}_u = \mathbf{u}. \tag{4.82}$$

After solving for $\mathbf{p}_u$ it should be normalized such that the last coordinate is equal to 1 in order to reduce the parameterization. The same process can be used to obtain the coordinates for $\mathbf{V}$, as well as each slice $\mathtt{T}_i$ of the tensor.

4.1.2.6 Polynomial Constraint Form

The rank, extended rank, and axes constraints are all formulated as polynomials on the tensor elements already. The epipolar constraints can also be written as polynomials on the tensor elements by expressing the null vectors $\mathbf{u}_i$ and $\mathbf{v}_i$ in terms of the tensor elements.

A basis for the null space of any $n \times n$ matrix having rank $n-1$ can be computed in closed

form by eliminating a row or column and taking $(n-1) \times (n-1)$ sub-determinants. Once the null spaces are known, it is straightforward to plug these into the rule of Sarrus. Because each element of $\mathbf{U}$ and $\mathbf{V}$ is quadratic in the tensor elements, and the determinant of a 3×3 is cubic in the elements, the epipolar constraints are therefore 5th degree.

However, it is often forgotten that it is necessary to eliminate a row or column that is linearly dependent on another row or column; otherwise, a zero vector will be extracted rather than a basis for the null space. Replacing any $\mathbf{u}_i$ with a zero vector only means that $\det \mathbf{U} = 0$ will be trivially satisfied. Thus, the particular polynomial that must be enforced by the 2 epipolar constraints must be chosen after identifying which rows are linearly independent.

Alternatively, the epipolar constraints could be translated into a larger set of equivalent polynomial constraints by considering all rank 2 possibilities of the $\mathtt{T}_i$ matrices. In particular, since only two of the three rows or columns will be linearly independent, it would be necessary to consider two possibilities for each $\mathtt{T}_i$. This leads to 2^3 possible choices for each of $\mathbf{e}'$ and $\mathbf{e}''$, so the 2 epipolar constraints can be represented by 16 *fixed* polynomial constraints.

The elements of $\mathbf{e}'$ and $\mathbf{e}''$ can be extracted in closed form using the same techniques, and their elements will be 4th degree polynomials because they are constructed from 2×2 determinants of a matrix having quadratic elements. Examining (4.68), the order of the circular constraints is therefore $4 + 4 + 4 + 4 + 1 = 17$.

The circular constraints are only trivially satisfied if *both* $\mathbf{e}'$ and $\mathbf{e}''$ are zero vectors, so an incorrect choice about which rows or columns of $\mathtt{T}_i$ are linearly independent would make the constraints in (4.68) appear to be violated, when they might actually be satisfied. Thus, it does not appear to be possible to enforce them as a set of fixed polynomials. However, the extended rank or axes constraints could be used instead to make a sufficient set of fixed polynomials, if that were desired for some reason.

4.2 Initial Tensor Estimation Algorithms

In this section we will review some of the known methods for estimating the trifocal tensor directly (i.e., without using nonlinear methods). In section Section 4.2.1 we describe the minimal algorithm for estimating a tensor from 6 points, in section Section 4.2.2 we describe the basic linear approach to estimating a tensor from 7 or more points (with several variations), and in Section 4.2.3 we mention some other algorithms for estimating the trifocal tensor and explain why we did not consider them in our evaluation.

4.2.1 Minimal Solution

It has been shown that any projective reconstruction algorithm that works on n views of $m+4$ points can be transformed into a dual algorithm for doing a projective reconstruction from

m views and $n+4$ points [Carlsson and Weinshall, 1998]. This observation is known as the Carlsson-Weinshall duality.

Thus, the relatively straightforward minimal reconstruction algorithm for 7 points in 2 views [Hartley and Zisserman, 2004, sec. 11.1.2] may be used to compute the dual problem of a minimal reconstruction from 6 points in 3 views [Quan, 1995; Carlsson and Weinshall, 1998; Hartley and Debunne, 1998; Hartley and Dano, 2000]. This is the basic idea behind the minimal approach, although some further tweaking is possible. The specific algorithm we use is given in [Hartley and Zisserman, 2004, alg 20.1], which we summarize below (with two minor improvements).

We denote the 3 unknown camera matrices as $\mathbf{P}_j$, $j = 1 \dots 3$, the 6 unknown structure points as $\mathbf{X}_i$, $i = 1 \dots 6$, and the image of the ith point in the jth view as $\mathbf{x}_i^j$. Therefore, the projection constraints are written as

$$\mathbf{x}_i^j \propto \mathbf{P}_j \mathbf{X}_i \quad \forall i, j. \tag{4.83}$$

In the dual algorithm, it will be necessary to use the image measurements as basis vectors, but the method only works if one chooses a set of 4 points, no 3 of which are collinear in any of the views.

Rather than simply verifying that the selection is not collinear (within a threshold), we take this a step beyond [Hartley and Zisserman, 2004, alg 20.1] by enumerating all 15 possible ways to pick the 4 points. For each way, we consider the 4 ways to pick a triangle out of the 4 points in each of the 3 views, and pick the set of 4 points so as to maximize the area of the triangle with minimal area in any view (our first improvement). Choosing the points in this way increases the stability of the remainder of the algorithm by ensuring that the points are as far from collinear as possible.

Using the fact that the area of a triangle is given by the determinant of the matrix constructed of homogeneous corner points as rows or columns, this maximization can be written as

$$\max_{\forall a,b,c,d} \left\{ \min_{\forall j} \left\{ \begin{array}{ll} \left|[\mathbf{x}_a^j | \mathbf{x}_b^j | \mathbf{x}_c^j]\right|, & \left|[\mathbf{x}_a^j | \mathbf{x}_b^j | \mathbf{x}_d^j]\right|, \\ \left|[\mathbf{x}_a^j | \mathbf{x}_d^j | \mathbf{x}_c^j]\right|, & \left|[\mathbf{x}_d^j | \mathbf{x}_b^j | \mathbf{x}_c^j]\right| \end{array} \right\} \right\}, \tag{4.84}$$

where $\{a, b, c, d\}$ is some combination of indices selected from $\{1, \dots, 6\}$. Alternatively, one could maximize the residual error to the least squares line (the result would be much the same). In the remaining steps, for notational convenience we assume that the points are ordered such that the selected 4 come first.

The second step is to find projective transforms $\mathbf{T}_j$ for each view $j = 1 \ldots 3$ that transform the first 4 points in that view to a canonical basis for the projective space $\mathbb{P}^2$. In other words,

$$\mathbf{T}_j \mathbf{x}_i^j = \mathbf{e}_i, \quad i = 1 \ldots 4, \tag{4.85}$$

where $\mathbf{e}_i$ for $i = 1 \ldots 3$ are the standard basis vectors of $\mathbb{R}^3$ and $\mathbf{e}_4 = (1, 1, 1)^\mathsf{T}$. These $\mathbf{T}_j$ matrices can be calculated in closed form, as shown in [Quan, 1995]. Then, by the Carlsson-Weinshall duality [Carlsson and Weinshall, 1998], correspondences in the dual problem are given by

$$\hat{\mathbf{x}}_j \leftrightarrow \hat{\mathbf{x}}'_j, \quad j = 1 \ldots 3, \tag{4.86}$$

where

$$\hat{\mathbf{x}}_j = \mathbf{T}_j \mathbf{x}_5^j \tag{4.87}$$

$$\hat{\mathbf{x}}'_j = \mathbf{T}_j \mathbf{x}_6^j. \tag{4.88}$$

In the dual problem, there are 4 implicit correspondences given by $\mathbf{e}_i \leftrightarrow \mathbf{e}_i$ for $i = 1 \ldots 4$. The constraints $\mathbf{e}_i^\mathsf{T} \hat{\mathbf{F}} \mathbf{e}_i = 0$ for $i = 1 \ldots 3$ imply that the diagonal elements of $\hat{\mathbf{F}}$ are zero, and the constraint $\mathbf{e}_4^\mathsf{T} \hat{\mathbf{F}} \mathbf{e}_4 = 0$ means that the sum of the elements of $\hat{\mathbf{F}}$ is zero. Thus, the dual fundamental matrix can be parameterized as

$$\hat{\mathbf{F}} = \begin{bmatrix} 0 & p & q \\ r & 0 & s \\ t & -(p+q+r+s+t) & 0 \end{bmatrix}. \tag{4.89}$$

From the additional dual correspondences in (4.86), 3 linear constraints are imposed on the entries p, q, r, s, t using $\hat{\mathbf{x}}_j^\mathsf{T} \hat{\mathbf{F}} \hat{\mathbf{x}}'_j = 0$. This leaves a 2-dimensional basis for the null space, but due to the overall scale ambiguity, there is just 1 degree of freedom remaining. Thus, we can write

$$\hat{\mathbf{F}} = \lambda \hat{\mathbf{F}}_1 + \hat{\mathbf{F}}_2, \tag{4.90}$$

where $\hat{\mathbf{F}}_1$ and $\hat{\mathbf{F}}_2$ are solutions corresponding to the null space basis vectors. The free parameter

λ is then determined using the the internal constraint that $\det \hat{\mathbf{F}} = 0$, a cubic equation for which there are 1 or 3 real solutions (complex/imaginary solutions can be ignored).

The next step is to retrieve a pair of reduced camera matrices compatible with the dual fundamental matrix. It is not known how these cameras might be formed directly from (4.89), but there is an alternative parameterization for the reduced fundamental matrix for which the answer is known. Specifically, if the reduced fundamental matrix is given by

$$\hat{\mathbf{F}} = \begin{bmatrix} 0 & b(d-c) & -c(d-b) \\ -a(d-c) & 0 & c(d-a) \\ a(d-b) & -b(d-a) & 0 \end{bmatrix}, \tag{4.91}$$

then a corresponding pair of reduced camera matrices is given by

$$\mathbf{P} = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix}, \qquad \mathbf{P}' = \begin{bmatrix} a & 0 & 0 & d \\ 0 & b & 0 & d \\ 0 & 0 & c & d \end{bmatrix}. \tag{4.92}$$

The question is then how to determine a, b, c, d in (4.91) from p, q, r, s, t in (4.89). It turns out that this can be solved linearly. Three linearly independent constraints are provided by

$$\begin{bmatrix} p & r & 0 \\ q & 0 & t \\ 0 & s & l \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} = \mathbf{0}, \tag{4.93}$$

and 3 more constraints (2 of which are linearly independent) are provided by

$$(d-a, d-b, d-c) \begin{bmatrix} 0 & p & q \\ r & 0 & s \\ t & -(p+q+r+s+t) & 0 \end{bmatrix} = \mathbf{0}. \tag{4.94}$$

These 6 constraints admit a least squares solution for a, b, c, d to be computed from p, q, r, s, t. Finally, back in the original measurement domain, the structure of the reconstruction is given by the dual of the dual reconstruction,

$$\mathbf{X}_i = \mathbf{E}_i, \quad i = 1 \dots 4 \tag{4.95}$$

$$\mathbf{X}_5 = (1, 1, 1, 1)^\mathsf{T} \tag{4.96}$$

$$\mathbf{X}_6 = (a, b, c, d)^\mathsf{T}, \tag{4.97}$$

where $\mathbf{E}_i$ are the standard basis vectors of $\mathbb{R}^4$. The camera matrices $\mathbf{P}_j$ can be computed in the original measurement domain by resectioning [Hartley and Zisserman, 2004, sec. 7.1], using the original measurements $\mathbf{x}_i^j$ and the reconstructed structure $\mathbf{X}_i$.

Because there may be up to 3 real solutions to (4.90) (only one of which is correct), it is recommended in [Hartley and Zisserman, 2004, alg 16.4] to make the function for computing the tensor output 3 possible results, all of which must then be tested within the RANSAC framework. We find this solution undesirable because it not only results in three times as much wasted computation, but also makes the output of the function 'messy.'

We have noticed that the initial search for triplet correspondences typically involves computing the fundamental matrix between the first two views, $\mathbf{F}_{21}$, to rule out bad matches that do not need to be searched for in the 3rd view. Therefore, we pass $\mathbf{F}_{21}$ into the minimal triplet routine and use it to select the correct solution which has the same fundamental matrix when there are 3 unique solutions (our second improvement).

4.2.2 Linear Algorithm

The following algorithm is from Hartley [1995], using the principles first developed in Shashua and Werman [1995]. A correspondence of three points $\mathbf{x} \leftrightarrow \mathbf{x}' \leftrightarrow \mathbf{x}''$ that are the images of one structure point $\mathbf{X}$ in each of the respective views gives rise to 9 linear constraints on $\mathcal{T}$. These constraints are not easily written in matrix notation, but can be expressed in tensor notation as

$$x^i(x'^j \epsilon_{jpr})(x''^k \epsilon_{kqs}) \mathcal{T}_i^{pq} = 0_{rs}, \tag{4.98}$$

where ϵ is the Levi-Civita symbol, x^i are the elements of $\mathbf{x}$, and a similar notation is used to denote the elements of $\mathbf{x}'$ and $\mathbf{x}''$. Only 4 of the 9 equations represented by (4.98) are linearly independent, so it is not necessary to use all of them. One choice of 4 linearly independent equations is given, after simplification, by

$$x^k(x'^i x''^l \mathcal{T}_k^{33} - x''^l \mathcal{T}_k^{i3} - x'^i \mathcal{T}_k^{3l} + \mathcal{T}_k^{il}) = 0, \quad \forall i, l \in \{1, 2\}. \tag{4.99}$$

These equations can be arranged into a homogeneous linear system,

$$\mathbf{A}\mathbf{t} = 0, \tag{4.100}$$

where $\mathbf{t}$ is a vector containing the elements of $\mathcal{T}$, and $\mathbf{A}$ is a constraint matrix containing 27 or more linearly independent rows. A least squares solution is obtained by minimizing $||\mathbf{A}\mathbf{t}||$ subject to $||\mathbf{t}|| = 1$, which can be accomplished using SVD [Hartley and Zisserman, 2004, alg A5.4].

There are primarily two limitations of this direct solution. First, none of the 8 internal constraints are enforced, so the tensor is not a consistent representation of any geometrical configuration. Second, the algebraic error that is minimized by SVD has no particular geometric meaning.

Because the error minimized by the linear solution has no particular geometric meaning, it is not surprising that the solution is not invariant to a scaling or translation of the image points. It has been noticed that normalizing the correspondence data generally leads to improved estimation accuracy [Hartley, 1998a; Hartley and Dano, 2000].

That is, instead of estimating $\mathbf{P}_i$ directly in $\mathbf{x} = \mathbf{P}_i\mathbf{X}$, it is recommended to replace $\mathbf{x}$ by $\tilde{\mathbf{x}} = \mathbf{H}_i\mathbf{x}$, where $\mathbf{H}_i$ is a 3×3 translation-scaling matrix constructed such that the distribution of points $\tilde{\mathbf{x}}$ in the ith image is centered around $(0, 0)$ and has a standard deviation of $\sqrt{2}$. Thus, one actually estimates $\tilde{\mathbf{P}}_i = \mathbf{H}_i\mathbf{P}_i$, and then maps the result back to $\mathbf{P}_i = \mathbf{H}_i^{-1}\tilde{\mathbf{P}}_i$.

4.2.2.1 Choosing Equations

In constructing the constraint matrix $\mathbf{A}$, there are a few different approaches that could be used. One option is to select only 4 of the 9 equations which are linearly independent, as in (4.99), for improved performance. However, it has been suggested that using all 9 constraints in (4.98) might give better results [Hartley, 1995]. A theoretical argument for using all 9 constraints was given in [Hartley and Zisserman, 2004, sec 17.7], where it was noted that the condition of the full set of equations is better, and therefore using all equations might help to avoid difficulties in near singular situations.

A third option is to translate the point-point-point correspondences into point-line-line correspondences [Hartley and Zisserman, 2004, sec. 17.7]. Given a correspondence between a point $\mathbf{x}$ in the first view, which is known to lie on a line $\mathbf{l}' = (l'_1, l'_2, l'_3)^\mathsf{T}$ in the second view and $\mathbf{l}'' = (l''_1, l''_2, l''_3)^\mathsf{T}$ in the third view, then there is one constraint on the tensor given by

$$x^i l'_q l''_r \mathcal{T}_i^{qr} = 0. \tag{4.101}$$

For each point-point-point correspondence, it is easy to generate 4 linearly independent point-line-line correspondences in the following manner: let $\mathbf{l}'^1$ and $\mathbf{l}'^2$ be two lines passing through $\mathbf{x}'$, and $\mathbf{l}''^1$ and $\mathbf{l}''^2$ be two lines passing through $\mathbf{x}''$. Then, the 4 constraints are given by

$$x^i l'^j_q l''^k_r \mathcal{T}_i^{qr} = 0, \quad \forall j, k \in \{1, 2\}. \tag{4.102}$$

If $\mathbf{l}'^1$ and $\mathbf{l}'^2$ are orthonormal, and $\mathbf{l}''^1$ and $\mathbf{l}''^2$ are orthonormal, then the resulting constraint matrix will have the same SVD as if all 9 point-point-point constraints had been used, and therefore give the same solution for lower computational cost.

It was suggested to find these orthonormal lines using Householder matrices, but we note that it is simpler to just use the horizontal and vertical lines passing through the point. Given a point $(x, y, 1)^\mathsf{T}$ in the image, the vectors representing these lines are given by

$$\mathbf{l}_h = \frac{(1, 0, -x)^\mathsf{T}}{\sqrt{1 + x^2}} \qquad \mathbf{l}_v = \frac{(0, 1, -y)^\mathsf{T}}{\sqrt{1 + y^2}}. \tag{4.103}$$

4.2.2.2 Enforcing Internal Constraints

A reconstruction from projection constraints alone is, at best, ambiguous up to an arbitrary projective transform having 15 degrees of freedom (dof) in homogeneous space. Each projection matrix has 11 dof, so there are $11m - 15$ dof to the projective geometry representing any configuration of m views [Hartley and Zisserman, 2004, sec. 17.5]. Thus, the projective geometry of 3 views has 18 dof.

The tensor is a homogeneous entity with 27 elements, so it has 26 dof, and this means that a geometrically consistent trifocal tensor must satisfy $26 - 18 = 8$ independent algebraic constraints. These constraints are implicitly enforced by the minimal estimation algorithm (Section 4.2.1), but cannot be directly enforced in the linear method (4.100). However, when mapping the tensor into projection matrices for general use with the algorithm of Section 4.1.1, one naturally obtains a geometrically consistent representation because projection matrices have no internal constraints. The problem with this passive approach to constraint enforcement is that the estimation is adjusted to satisfy internal consistency without regard to the image correspondence constraints, and this could potentially result in a very large increase in reprojection error.

A better solution is to reestimate a consistent tensor in a second linear step by holding some aspects from the original estimation fixed. We refer to these as quasi-linear methods. The first such method was given in Hartley [1995], where it was pointed out that if $\mathbf{a}_4$ and $\mathbf{b}_4$ from (4.13)

are known, then the tensor may be expressed linearly in terms of the remaining elements of the projection matrices. Specifically, one can write

$$\mathbf{t} = \mathbf{E}\mathbf{a}, \tag{4.104}$$

where $\mathbf{a}$ contains all the elements of $\mathbf{a}_i, \mathbf{b}_i \; \forall i \in \{1, 2, 3\}$, and $\mathbf{E}$ is a constraint matrix based on the known $\mathbf{a}_4$ and $\mathbf{b}_4$. From (4.26-4.34) it can be seen that, without loss of generality, one can choose $\mathbf{a}_4 = \mathbf{e}'$ and $\mathbf{b}_4 = \mathbf{e}''$, which can be extracted from the initial linear estimate using (4.19-4.20). Plugging (4.104) into (4.100), one obtains

$$\mathbf{A}\mathbf{E}\mathbf{a} = 0. \tag{4.105}$$

Thus, the initial problem (4.100) of minimizing $||\mathbf{A}\mathbf{t}||$ subject to $||\mathbf{t}|| = 1$ is analogous to minimizing $||\mathbf{A}\mathbf{E}\mathbf{a}||$ subject to $||\mathbf{E}\mathbf{a}|| = 1$, but the latter guarantees a geometrically consistent result.

Because this is not in the traditional form that can be easily solved by taking the right singular vector of $\mathbf{AE}$, and the $||\mathbf{E}\mathbf{a}|| = 1$ constraint has no geometrical significance beyond preventing the trivial solution of $\mathbf{a} = 0$, one might instead minimize $||\mathbf{A}\mathbf{E}\mathbf{a}||$ subject to $||\mathbf{a}|| = 1$. However, because $\mathbf{E}$ is not full rank, the solution vector would not be uniquely determined. In order to ensure a unique solution, it was suggested to use additional constraints of

$$\mathbf{a}_i \cdot \mathbf{a}_4 = 0 \quad i \in \{1, 2, 3\}, \tag{4.106}$$

which it was shown can be imposed without loss of generality. These additional constraints can be written as a system of linear equations by constructing an appropriate matrix $\mathbf{C}$ in

$$\mathbf{C}\mathbf{a} = 0, \tag{4.107}$$

and the minimization of $||\mathbf{A}\mathbf{E}\mathbf{a}||$ subject to $||\mathbf{a}|| = 1$ and $\mathbf{C}\mathbf{a} = 0$ can be performed linearly using [Hartley and Zisserman, 2004, alg A5.5].

It is not obvious if the addition of these latter constraints would actually be beneficial because the non-unique solutions would still be equivalent under the projective ambiguity, and the potential downside is that the degree to which the real trilinear constraints are violated must be increased in order to reduce the error on these artificial constraints.

Shortly thereafter in Hartley [1998a], it was shown that the problem of minimizing $||\mathbf{AEa}||$ subject to $||\mathbf{Ea}|| = 1$ could be solved directly ([Hartley and Zisserman, 2004, alg A5.6]), and this has become Hartley's recommended method in Hartley and Zisserman [2004, alg 16.2].

To summarize, there are three interesting quasi-linear methods that may have similar performance, and we put all three variations to the test in our empirical comparison:

$$\min ||\mathbf{AEa}|| \text{ subject to } ||\mathbf{a}|| = 1 \tag{4.108}$$

$$\min ||\mathbf{AEa}|| \text{ subject to } ||\mathbf{a}|| = 1 \text{ and } \mathbf{Ca} = 0 \tag{4.109}$$

$$\min ||\mathbf{AEa}|| \text{ subject to } ||\mathbf{Ea}|| = 1. \tag{4.110}$$

4.2.3 Algorithms not Considered

There exist a number of iterative algorithms for estimating the trifocal tensor, such as using the Sampson approximation [Hartley and Zisserman, 2004, sec 16.4.3] (first used for conic fitting by Sampson in [Sampson, 1982]), iterative adjustment of the epipoles [Hartley and Zisserman, 2004, sec 16.3], iterative adjustment of the image points [Torr and Zisserman, 1997], the nonlinear algorithm from [Faugeras and Keriven, 1998], or nonlinear enforcement of internal constraints [Hartley, 1998a]. We do not consider these nonlinear algorithms because they all require an initial estimate (e.g., found by the linear method), and once an initial geometrically valid tensor has been found, it can be converted into camera matrices without loss of information and then bundle adjustment is the maximum likelihood nonlinear improvement. Therefore, we concentrate our search only on finding the best geometrically consistent initialization.

We do not consider parameterizations of the linear algorithm using reduced affine coordinates such as [Heyden, 1998] because none of the error is distributed onto the estimation in the first view, and this will necessarily yield inferior results in comparison to a solution that evenly distributes the error across all views.

We do not consider the linear Factorization method [Tomasi and Kanade, 1992] (or its variations), because it assumes orthographic projection which is a crude approximation to perspective projection. Although there exist nonlinear methods to correct for perspective effects, such as in [Christy and Horaud, 1996], the initial orthographic solution might not be in the basin of attraction of the perspective correct solution. Therefore, it is an inferior approach to the linear algorithm which properly accounts for perspective in the initial solution.

Finally, we do not consider globally optimal approaches to estimate the trifocal tensor using branch and bound [Hartley and Kahl, 2009] because the exponential time complexity of this approach admittedly makes it impractical for general use, much less integration into a framework requiring many repeated evaluations such as RANSAC.

4.3 Robust Estimation with RANSAC

In practice, a correspondence set will usually contain some mismatches (outliers) that would be inconsistent with the true reconstruction. A robust procedure for dealing with outliers in any model fitting problem is RANSAC [Fischler and Bolles, 1981], and is often applied to computation of the trifocal tensor, as in Torr and Zisserman [1997]. There have been many improvements to the original RANSAC algorithm (see Raguram et al. [2008] for a more modern survey) but we mention only the basic algorithm here for simplicity.

The objective of RANSAC is to find the largest sample consensus; i.e., to find the model that is consistent with the largest subset of the data. This is achieved by picking many random subsets, creating an initial reconstruction from each subset, classifying inliers according to a threshold, and storing the model with the largest set of inliers that was found.

The usual way to choose the number of trials needed is by a probabilistic argument [Fischler and Bolles, 1981]: if the size of each random subset is s and the percent of inliers is p, then the probability of picking a subset of all inliers is p^s. If, after n trials, no trial subset has contained all inliers, then the overall result is failure. Thus, the probability of failure f is given by

$$f = (1 - p^s)^n. \tag{4.111}$$

Rearranging, one can solve for the minimum number of trials needed to meet any given probability of failure,

$$f = (1 - p^s)^n \tag{4.112}$$

$$n = \log_{(1-p^s)} f = \frac{\log f}{\log(1 - p^s)}. \tag{4.113}$$

Although p is not usually known in advance, it can be increased adaptively whenever a new larger sample consensus is found until the termination condition is exceeded [Torr and Zisserman, 1997].

4.4 Experimental Results

We start by trying to find the best variation of the linear algorithm by comparing the different ways to enforce internal constraints and represent the trilinear constraints (Section 4.4.1). Once we have identified the best linear variation, we compare the minimal 6 point algorithm to the best linear variation with 7 points in terms of accuracy and runtime performance (Section

4.4.2). Lastly, we investigate the effect of the number of points used (either 6 for the minimal algorithm ,or 7+ with the linear method) on the overall performance on RANSAC (Section 4.4.3).

In most of our tests we have used synthetic data where the levels of noise can be precisely controlled to more accurately investigate the dependence on noise. We generate synthetic correspondences from uniformly distributed 3D structure points in a $[-50, 50]^3$ volume imaged by camera views on a circle having random radius uniformly distributed in the range $(200, 1000)$. Each camera has a 45° field of view with principal point in the center of the image, and the separation between each camera on the circle is uniformly distributed in the range of $(0.01, 5)$ degrees. Correspondences are generated by projecting the structure points into each image plane and adding uniformly distributed random noise in the range $(-\varepsilon, \varepsilon)$ pixels, for a noise level of ε.

4.4.1 Best Linear Variation

We start by comparing the three methods of quasi-linear reestimation to enforce internal constraints. We first plotted the median of the mean reprojection error from 1000 repetitions as a function of noise (Fig. 4.2a). Note that we have only considered $\varepsilon < 1$, because generally the precision of a correspondence finder is limited by the image discretization. However, we also note that effective noise is relative to object distance.

Our results indicate that (4.108) actually increased the error in comparison to the passive method (Section 4.1.1), whereas minimizing either (4.109) or (4.110) reduced the error by roughly 50%, with no noticeable difference between the two. However, we observed some sensitivities when minimizing (4.110), and a plot of reconstruction error as a function of the SVD precision tolerance (Fig. 4.2b), with the noise level fixed at $\varepsilon = 0.5$, shows that in fact (4.110) is a much less stable algorithm. Specifically, (4.110) demands a precision of at least 1×10^{-15} to give good results, whereas (4.109) was insensitive to the SVD precision, requiring fewer iterations for the same accuracy. We therefore conclude that (4.109) is the superior way to enforce internal constraints, despite it being an older and lesser known method.

Next, we considered the various methods for representing trilinear constraints described in Section 4.2.2.1. These methods were tested on configurations of 100 structure points using the 7 point linear method. We measured the reprojection error for the 7 fitted points (Fig. 4.3, left), as well as the remaining 93 points (Fig. 4.3, middle) by looking at the residual errors from maximum likelihood triangulation. As before, we plotted the median results over 1000 trials for each level of noise.

From previous theoretical arguments (Section 4.2.2.1), one would expect to see equivalent results using either all 9 point-point-point constraints (`9ppp`) or the 4 point-line-line constraints

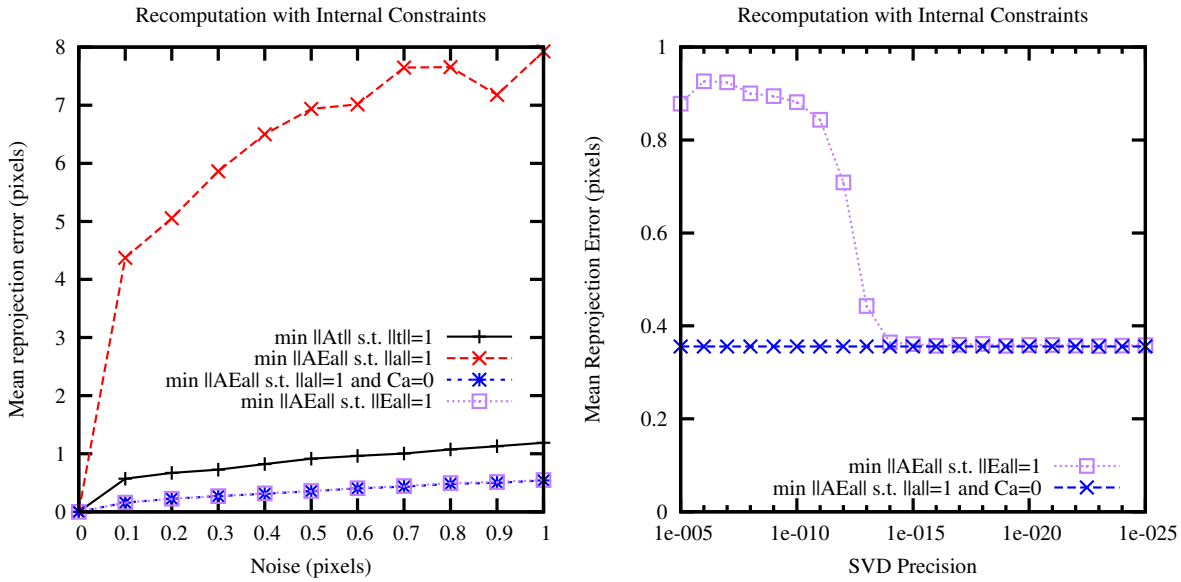


Figure 4.2. Comparison of methods for enforcing internal constraints in the linear algorithm by quasi-linear reestimation. The minimization of $||\mathbf{At}||$ s.t. $||\mathbf{t}|| = 1$ is the basic linear algorithm, and constraint enforcement is done passively when mapping back to projection matrices (Section 4.1.1); the minimization of $||\mathbf{AEa}||$ s.t. $||\mathbf{a}|| = 1$ and $\mathbf{Ca} = 0$ ((4.109)) is the quasi-linear re-estimation method from Hartley [1995]; the minimization of $||\mathbf{AEa}||$ s.t. $||\mathbf{a}|| = 1$ ((4.108)) investigates the necessity of the $\mathbf{Ca} = 0$ constraint; the minimization of $||\mathbf{AEa}||$ s.t. $||\mathbf{Ea}|| = 1$ ((4.110)) is the method from Hartley [1998a]. (a) the mean reprojection error for each estimation method is shown as a function of correspondence noise, with the median over 1000 trials is plotted. (b) although the difference between (4.109) and (4.110) is imperceptible from (a), the error as a function of the SVD precision tolerance, with $\varepsilon = 0.5$, shows that (4.110) is much less stable and requires more iterations for a reliable result. This plot is also the median over 1000 trials.

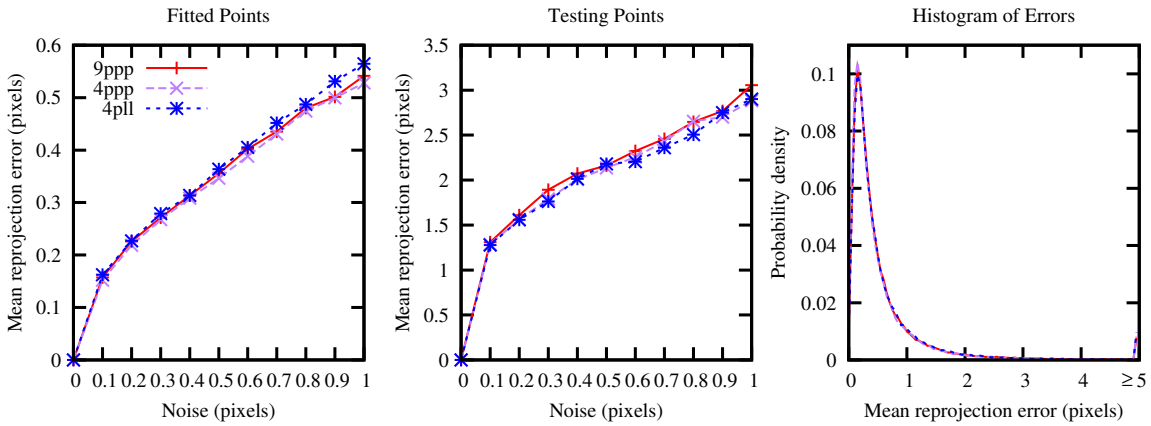


Figure 4.3. Effect of choosing different linear constraints in using the 7 point linear algorithm. Data sets were generated from 100 points. Left: mean reprojection error for the fitted data (first 7 points). Plot shows the median over 1000 trials. Middle: mean reprojection error for the testing data set (remaining 93 points), determined by triangulation minimizing the L_2-norm of reprojection errors. Plot shows the median over 1000 trials. Right: comparison between empirical PDF of mean reprojection error on the fitted data for each method, determined from 100,000 trials. Correspondence noise was set to $\varepsilon = 0.5$.

(`4pll`), and slightly worse results using just the 4 linearly independent point-point-point constraints (`4ppp`). However, our results showed no significant difference in median performance. In order to see if there was a difference in worst-case performance, we also analyzed the histogram of performance from 100,000 random configurations (Fig. 4.3, right). Surprisingly, the distribution of performance appears exactly identical. Therefore, we conclude that there is no justification for using the more computationally expensive `9ppp` method, and there is also no need to complexify the implementation by translating the point-point-point constraints into point-line-line constraints; in other words, we conclude that it is best to simply use the four linearly independent point-point-point constraints (`4ppp`).

4.4.2 Minimal vs. Linear

Having identified the best linear variation, we are now prepared to compare the performance of the linear method to the minimal 6 point method. This was done by generating configurations of 100 points and then reconstructing from a subset of 6 points using the minimal method or 7 points using the linear method.

We first plot the median of the mean reprojection error on the fitted data (Fig. 4.4, left) as an indicator of precision. Because the minimal 6 point method is an exact solution it is expected to have zero error, and this is confirmed in the plot. We measured actual error on the order of 1×10^{-12} which is due only to limited numerical precision. The linear algorithm has non-zero error that increases with noise because it is over-determined, and the fact that the reconstruction error remains proportional to and slightly less than the noise indicates that it is

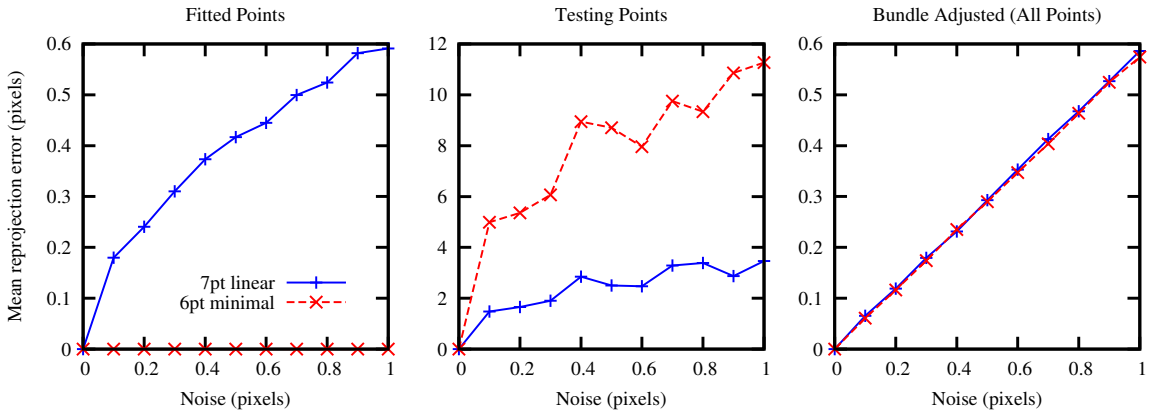


Figure 4.4. Comparison between minimal 6 point algorithm and best-performing variation of the linear 7 point algorithm. The left panel shows errors on the fitted data, indicating the level of precision. The middle panel shows reprojection errors after triangulation on the 100 testing correspondences, indicating the accuracy of the reconstruction. The right panel shows the result of finalizing with bundle adjustment on all available data.

capable of fitting the data well.

A second graph showing the median of the mean reprojection error for additional testing correspondences after maximum likelihood triangulation (Fig. 4.4, middle) shows how accurate the reconstruction actually was; here we see that the minimal 6 point algorithm, while precise, is much less accurate because it does not fit the testing points nearly as well as the linear algorithm for any non-zero level of noise. This is in contrast to previous results that did not use the quasi-linear enhancements [Torr, 1995; Torr and Zisserman, 1997], where the minimal algorithm was found to be superior.

A third graph shows the median of the mean error on all available data after bundle adjustment, which shows that even though the minimal initialization was worse, both methods are usually in the basin of attraction of the global minimum.

Runtime performance between the minimal 6 point method and the linear with 7 or more points was compared with a plot of the mean reconstruction runtime for 1000 random configurations (Fig. 4.5). We observed that the linear method exhibits $O(n)$ performance, at least when n (the number of points) was less than 80, despite the fact that the computational cost of SVD is $O(n^3)$. A linear regression gives shows that the performance of our implementation of the n-point linear method is about $0.096125n + 0.503315$ microseconds on the testing machine (Intel Core i7 920), compared to 0.124317 microseconds for the 6 point minimal method. In other words, the minimal algorithm is significantly faster, but the linear algorithm is still quite fast for practical purposes.

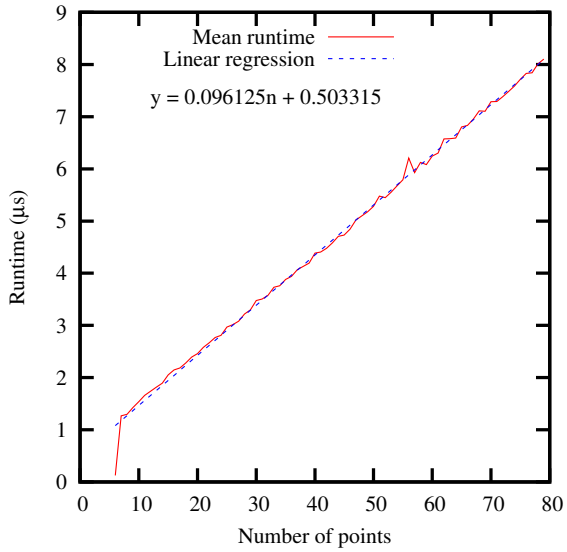


Figure 4.5. Mean reconstruction runtime as a function of the number of points used. The minimal algorithm is used for 6 points and the linear algorithm is used for 7 or more points.

It has been assumed that the minimal method will require the fewest iterations for RANSAC convergence, and in addition we have shown that the minimal algorithm by itself is significantly faster than the linear algorithm. However, we have speculated that the robustness to noise gained by the over-determined nature of the linear algorithm may actually lead to superior performance. We first investigated this by analyzing the size of the largest consensus size as a function of RANSAC trials using each of the minimal, 7 point linear, and 15 point linear algorithms on a random configuration (Fig. 4.6). The configuration consisted of 100 structure points with 80% inliers. The experiment was repeated at three noise levels for $\varepsilon = \{0, 0.5, 1\}$, and the RANSAC inlier threshold was fixed at $\tau = 1.75$. The most interesting observation from these results is that, in the presence of noise, using a larger number of points allowed RANSAC to converge to a larger final consensus size.

4.4.3 Subset size in RANSAC

As observed in Chum et al. [2003], the termination condition for RANSAC is based on the assumption that an estimate from an uncontaminated sample will correctly classify all inliers, which is not true in the presence of noise. Using non-minimal subsets to estimate the model parameters improves robustness to noise and may therefore allow convergence to a larger final consensus size in fewer iterations with greater reliability. One way to realize these gains is by using the LO-RANSAC algorithm [Chum et al., 2003], and another alternative is to simply run RANSAC with non-minimal subsets.

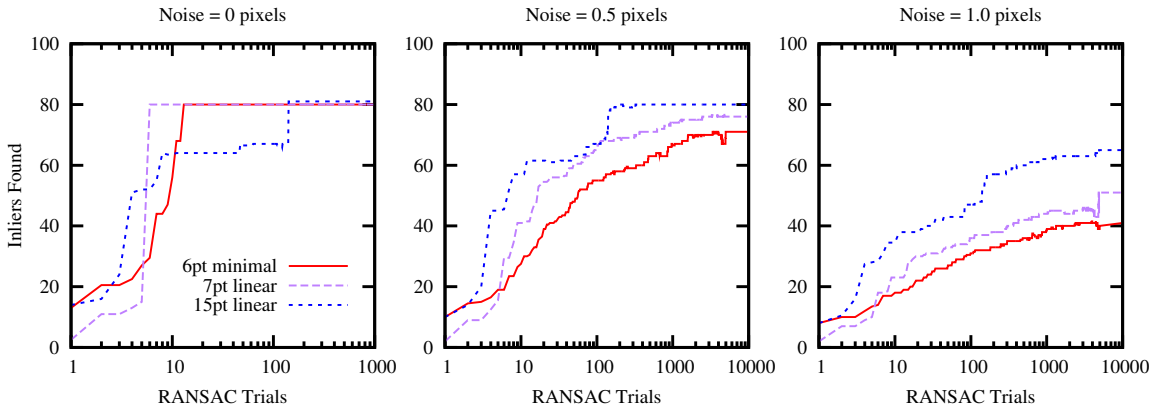


Figure 4.6. Comparison of RANSAC convergence using the 6 point algorithm, and the best linear variation from 7 and 15 points. The data set contained 100 points, of which 20 were outliers($p = 0.8$). The experiment is repeated using correspondence noise levels of $\varepsilon \in \{0, 0.5, 1\}$. The inlier threshold was fixed at $\tau = 1.75$ pixels. The median size of the largest consensus set over 100 random data sets is plotted as a function of RANSAC iterations.

The following experiments were designed to examine in detail the effect of varying subset size in standard RANSAC. First, we looked at the accuracy of the linear method as a function of the number of points n, for $n = 7, \ldots, 80$, to see how many points are necessary before one reaches diminishing returns in the accuracy of the reconstructed tensor. We fixed the correspondence error level at $\varepsilon = 0.5$ and generate correspondences from configurations of 100 points.

Looking at the mean reprojection error on just the fitted data before and after bundle adjustment (Fig. 4.7, left) indicates how close the linear estimate is to the maximum likelihood estimate. We see that the maximum likelihood estimate is significantly better for $n < 10$ points, but with about 15 or more points, the linear estimate is almost as good as a maximum likelihood estimate. We also looked at how well the remaining data points fit with this model before and after bundle adjusting using all data (Fig. 4.7, right). Here, we see also that a linear estimate from 10-15 points is typically capable of fitting all the remaining points very well. Using more points in the initial linear estimate causes an asymptotic convergence to the true configuration, but the returns are diminishing.

The ideal number of points to use in RANSAC depends upon both the inlier fraction and the noise distribution. To demonstrate this we define the RANSAC Rerformance Ratio (PR) to be the total number of inliers divided by the total runtime. We then calculated the subset sizes that empirically optimize the performance ratio over 100 courses of running RANSAC at various combinations of noise and inlier percentages (Fig. 4.8). Inliers were corrupted with normally distributed noise having σ at the specified level whereas outliers were corrupted with noise having $\sigma = 50$ pixels. The inlier threshold was set automatically at $\max(0.5, 3\sigma)$ and

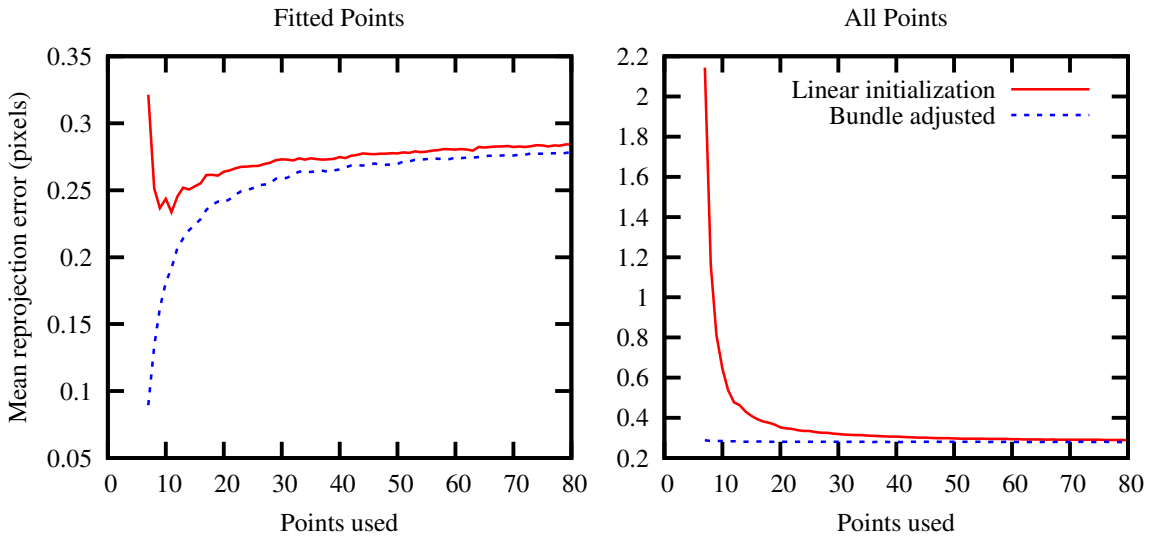


Figure 4.7. Dependence of linear reconstruction quality on the number of points used (median over 200 trials). The left panel shows reprojection errors on the fitted data, before and after bundle adjustment. The right panel shows reprojection errors on all available data (from 100 points), where additional points are initialized by triangulation, before and after bundle adjustment.

RANSAC was run until at least 95% of the inliers were found.

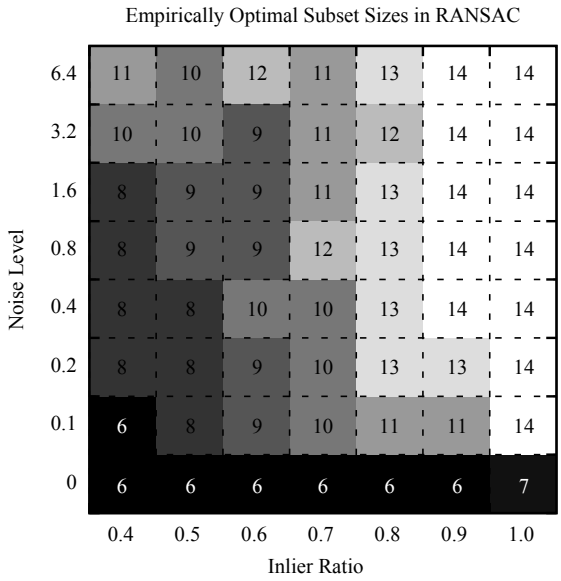


Figure 4.8. Empirically optimal subset sizes that maximize the summed performance ratio of final consensus sizes divided by total runtimes after running RANSAC 100 times. The minimal 6 point algorithm has a better performance ratio when there is zero noise, but a linear algorithm using more points gives superior performance when noise is introduced.

In these synthetic tests, we see clearly that the minimal 6 point algorithm maximizes the performance ratio only for zero noise. Even with an unrealistically low level of noise ($\sigma 0.1$ pixels), the linear method has a better performance ratio. In general, as either the noise level or the inlier fraction is increased, the benefits of using a larger subset size are increased. However, we note that the noise is relative to scene configuration, and therefore this table should not be used as a reference for choosing the subset size of real configurations based on the noise level and inlier fraction.

In order to see what size performs best on real image data we generated correspondences by automatically matching Harris [Harris and Stephens, 1988] corner points using the Normalized Cross Correlation. For the *Bookshelves* scene (Fig. 4.9), we found a total of 1369 triplet correspondences. We plotted the overall consensus size and runtime of RANSAC as a function of the number of points (Fig. 4.9a), with the corresponding performance ratio plotted in (Fig. 4.9b). The best performance was found using the 8 point linear method. We obtained a final consensus size of 1172/1369 points, and after bundle adjustment, the mean squared reprojection error was reduced to 0.159661 pixels (relative to the 1148×764 images).

Our results on another real scene, the *Desk* scene, (Fig. 4.10), shows similar results; this time we found a total of 1340 triplet correspondences. The overall consensus size and runtime of RANSAC as a function of the number of points is plotted in Fig. 4.10a, with the corresponding performance ratio plotted in (Fig. 4.10b). The best performance was again found using the 8 point linear method. We obtained a final consensus size of 1003/1340 points, and after bundle adjustment, the mean squared reprojection error was reduced to 0.192335 pixels (relative to the 1024×768 images).

4.5 Conclusions

We have introduced two small improvements to the minimal 6 point algorithm, one being a method for selecting points that form a stable basis in order to ensure precise results, and the other being a method for selecting the correct solution when there are 3 possible solutions.

We have also examined several variations of the linear algorithm in order to the determine the most accurate and efficient variation. We have shown that an older, lesser used, method of quasi-linear enforcement of the internal constraints actually performs best, and that there appears to be no difference in performance between the various methods of trilinear constraint representation, which leads us to believe that it is best to stick with the simplest and fastest method.

Contrary to previous results, we show that the best variation of the 7-point linear method is consistently more accurate than the minimal 6-point algorithm, and the linear estimate is nearly a maximum likelihood estimate when estimated from more than 10 points. We also

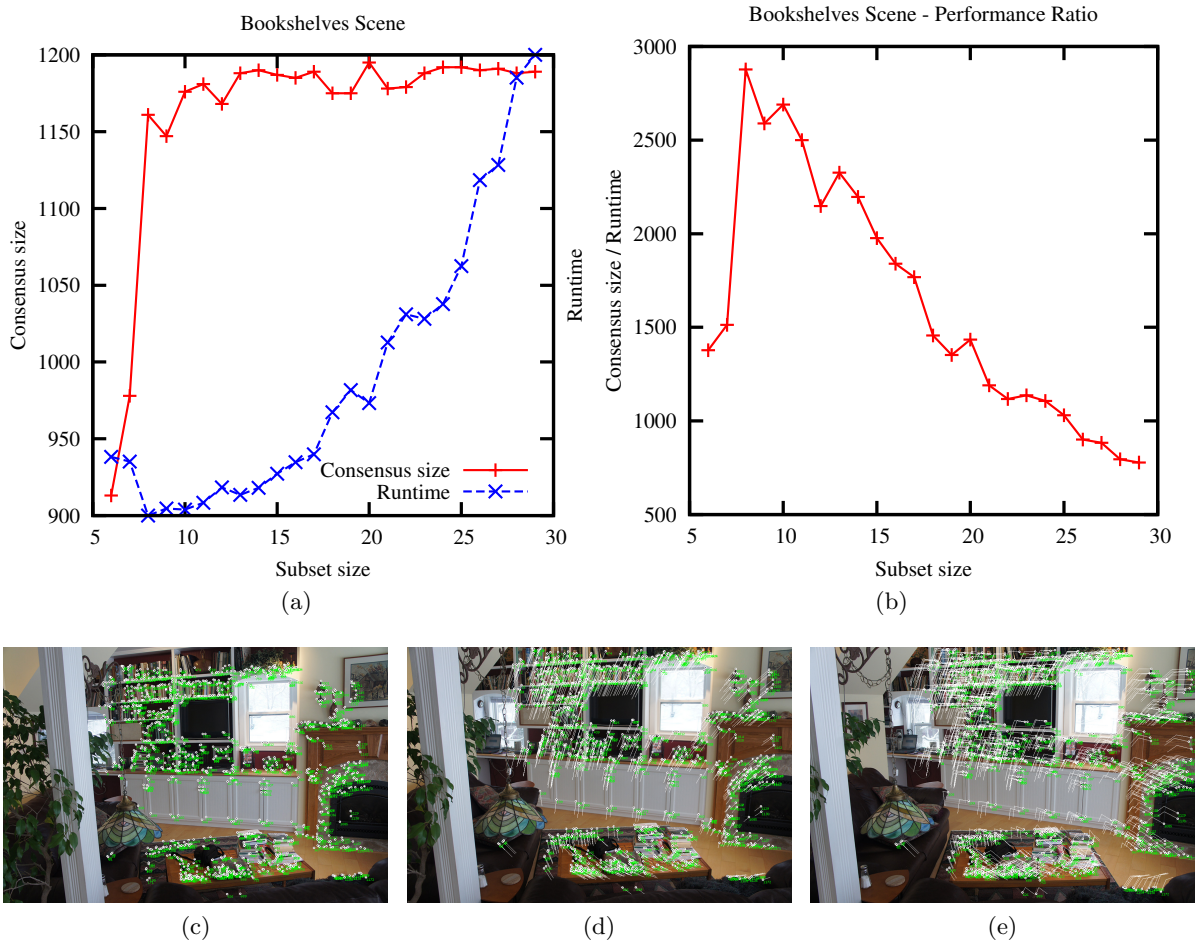


Figure 4.9. Example reconstruction using the trifocal tensor. The inlier threshold was automatically determined at 1.01605 pixels, and 1172 out of 1369 triplet correspondences were found as inliers. The mean squared reprojection error is 0.159661 pixels (in comparison, the image size is 1148×764 pixels).

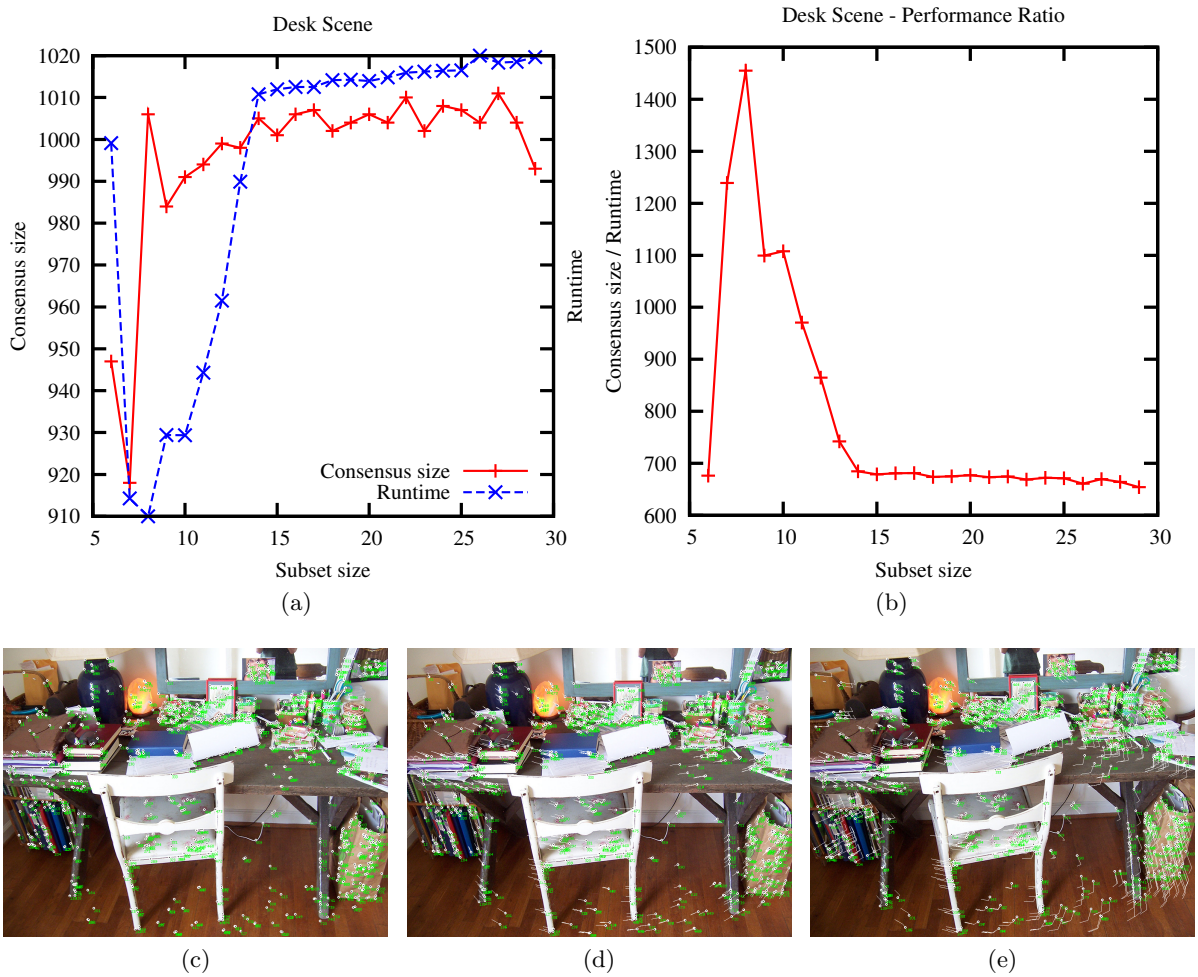


Figure 4.10. Example reconstruction using the trifocal tensor. The inlier threshold was automatically determined at 1.09729 pixels, and 1003 out of 1340 triplet correspondences were found as inliers. The mean squared reprojection error is 0.192335 pixels (in comparison, the image size is 1024×768 pixels).

show that using larger subset size in RANSAC with the linear method allows a larger final consensus size to be reached, and in a shorter overall runtime, despite the fact that runtime for the minimal method by itself is substantially faster. Further research will be needed to compare the efficiency tradeoffs with the LO-RANSAC approach [Chum et al., 2003].

Chapter 5

Projective Merging

In the structure-from-motion (SfM) problem, the objective is to simultaneously compute a reconstruction of 3D structure points and camera parameters from the motion parallax information encoded in a set of measured image correspondences. With uncalibrated cameras, projective reconstruction is usually the first step, followed by autocalibration to yield a metric reconstruction [Hartley and Zisserman, 2004, p.265].

A projective reconstruction satisfies projection constraints but makes no assumptions about the camera intrinsic parameters, and is ambiguous up to a 4×4 projective ambiguity. This ambiguity can theoretically be resolved by autocalibration, which imposes the prior knowledge that real cameras do not produce skewed or stretched images [Pollefeys et al., 1998]. However, autocalibration is not a well-posed problem; in general, there will not exist any 4×4 matrix that causes the camera intrinsic constraints to be satisfied exactly. Most autocalibration algorithms are very sensitive to reconstruction quality and will fail ungracefully when the initial projective reconstruction is not sufficiently accurate. Thus, autocalibration should be delayed until the projective reconstruction is as accurate as possible.

A commonly used technique for making a reconstruction that spans an arbitrary number of views is to compute many smaller independent reconstructions and then merge them together in order to obtain a larger reconstruction [Fitzgibbon and Zisserman, 1998; Nister, 2001a; Farenzena et al., 2009; Frahm et al., 2010]. Most previous merging approaches have derived merging constraints by using correspondences between structure points [Fitzgibbon and Zisserman, 1998; Repko and Pollefeys, 2005; Farenzena et al., 2009; Frahm et al., 2010]. However, because Euclidean distance is not preserved under the projective ambiguity, this has required autocalibration to be performed on each partial reconstruction prior to merging, which is not only computationally expensive but also increases the risk of system failure due to the instabilities of autocalibration.

In this chapter, we revisit a linear approach to merging that measures distance in image

space, thereby avoiding the need for premature autocalibration and also reducing the sensitivity to uncertainty in the structure points (Section 5.2.1). We show that although this approach usually produces good results, it can be unstable for certain camera configurations, but using the method symmetrically overcomes this problem (Section 5.3). Next, we propose a maximum likelihood nonlinear improvement of the merging homography that is completely invariant to the uncertainty in structure points (Section 5.4). We show how to robustly deal with outliers using this approach (Section 5.5), as well as how to efficiently merge inter-frame correspondences in order to strengthen the projective constraints for larger reconstructions while avoiding the systematic accumulation of errors (Section 5.6).

5.1 Merging Homography

The perspective projection of a homogeneous structure point $\mathbf{X} \in \mathbb{P}^3$, as viewed by a camera with 3×4 projection matrix $\mathbf{P}$, is a homogeneous image point $\mathbf{x} \in \mathbb{P}^2$, given by

$$\mathbf{x} \propto \mathbf{P}\mathbf{X}. \tag{5.1}$$

Let the estimate of the projection matrix for the jth view be denoted by $\widehat{\mathbf{P}}_j$ in the *left* reconstruction when it exists, and by $\widehat{\mathbf{P}}'_j$ in the *right* reconstruction when it exists. Similarly, the estimate of the ith structure point in the left reconstruction will be denoted by $\widehat{\mathbf{X}}_i$, and by $\widehat{\mathbf{X}}'_i$ in the right reconstruction.

Because both the left and right reconstructions are approximately related to some ground truth configuration by a 4×4 homography, there will also exist a 4×4 homography $\mathbf{H}$ that approximately relates the right reconstruction to the left reconstruction,

$$\widehat{\mathbf{P}}_j \propto \widehat{\mathbf{P}}'_j \mathbf{H} \quad \forall j \tag{5.2}$$

$$\widehat{\mathbf{X}}_i \propto \mathbf{H}^{-1} \widehat{\mathbf{X}}'_i \quad \forall i. \tag{5.3}$$

The goal of projective merging is to find the best possible estimate of $\mathbf{H}$. Once $\mathbf{H}$ is known, all projection matrices and structure points in the right reconstruction can be placed into the same projective reference frame as the left reconstruction using (5.2) and (5.3).

5.1.1 View Constraints

The homography $\mathbf{H}$ has 16 elements but just 15 degrees of freedom (dof) because it is a homogeneous entity with arbitrary scale; similarly, each 3×4 projection matrix has 11 dof. Any view j for which $\widehat{\mathbf{P}}_j$ exists in the left reconstruction and $\widehat{\mathbf{P}}'_j$ exists in the right reconstruction

is an *overlapping view*, and hence by (5.2), $\mathbf{H}$ is over-determined and can be estimated using linear least squares from two or more overlapping views. However, we avoid this approach for the following reasons:

1. Because there can only be one estimate for each projection matrix, any overlapping views in the right reconstruction will be discarded when merging the right reconstruction into the left reconstruction (see Fig. 5.1). However, $\widehat{\mathbf{P}}_j$ will not be exactly equal to $\widehat{\mathbf{P}}'_j\mathbf{H}$, and there is no guarantee on how similar they will be. Even a small change in one element of a projection matrix can result in an *arbitrarily large* increase in the reprojection error of a structure point, depending on where that point is in 3D space. Thus, the results could be very unstable.

2. A least squares approach to merging using constraints from overlapping projection matrices would align them by minimizing the Frobenius norm. However, this is not a meaningful quantification of error because it does not consider the location of structure points that were used to estimate the projection matrix. Thus, the transformation that minimizes the Frobenius norm does not even approximately attempt to minimize reprojection error and this would magnify the effect of errors from (1).

3. Because overlapping views are discarded during a merge, it is computationally wasteful to use more overlapping views than necessary. For example, the result of merging two triplets with two overlapping views is only a net increase of one view in the merged reconstruction.

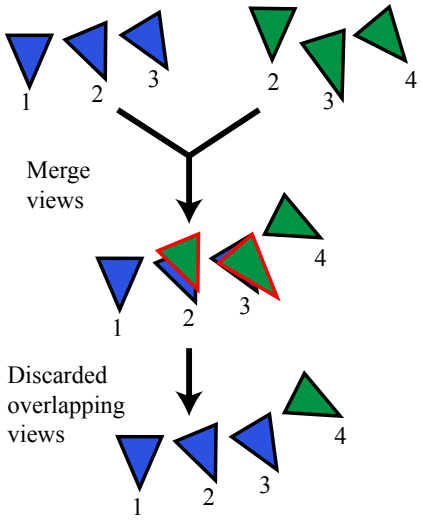


Figure 5.1. Example of projective merging with two views of overlap. The left reconstruction (blue) uses views $\{1, 2, 3\}$ and the right reconstruction (green) uses views $\{2, 3, 4\}$. In the first step, the right reconstruction is merged into the projective frame of the left reconstruction using only view constraints. Notice that neither of the projection matrices perfectly align. The two overlapping views (identified with red border) are discarded, causing the relative pose between views $\{2, 3, 4\}$ to be altered in a way that, depending on the location of structure points, may result in an unbounded increase of reprojection errors.

If there are no overlapping views then merging is still possible using (5.3), if corresponding structure points can somehow be identified. However, in order for a correspondence to exist between structure points there must have been a point visible in at least two views of each reconstruction (so that it could be triangulated), and if there are no overlapping views then this means that the point must have been imaged in at least four views. This is undesirable because it becomes exponentially more difficult to find reliable correspondences across more views.

If there is a single view of overlap, then this leaves $15 - 11 = 4$ dof remaining in the estimation of $\mathbf{H}$; thus, it is always possible to align the reconstructions so that *one* overlapping view is exactly equal, and then there will be no additional increase in reprojection error when this identical view is discarded during the merge. Furthermore, a correspondence of structure points requires an image feature to be identified in just three views, as noted in Laveau [1996]. Thus, we consider single-view overlap to be the superior choice.

When there is a single view of overlap $\mathbf{H}$ can be parameterized so as to enforce (5.2) exactly [Fitzgibbon and Zisserman, 1998]. Multiplying (5.2) by the pseudo-inverse, one obtains an equation for $\mathbf{H}$,

$$\widehat{\mathbf{P}}_j'^{+}\widehat{\mathbf{P}}_j \propto \mathbf{H}. \tag{5.4}$$

However, there is actually a 4-parameter family of solutions defined by the choice of $\mathbf{v}$ in

$$\mathbf{H}(\mathbf{v}) = \widehat{\mathbf{P}}_j'^{+}\widehat{\mathbf{P}}_j + \widehat{\mathbf{C}}'\mathbf{v}^{\mathsf{T}}, \tag{5.5}$$

where $\widehat{\mathbf{C}}'$ is the null space (eg, center of projection) of $\widehat{\mathbf{P}}_j'$. This can be easily verified by left-multiplying again by $\widehat{\mathbf{P}}_j'$ to yield

$$\widehat{\mathbf{P}}_j'\mathbf{H}(\mathbf{v}) = \widehat{\mathbf{P}}_j + \widehat{\mathbf{P}}_j'\widehat{\mathbf{C}}'\mathbf{v}^{\mathsf{T}}, \tag{5.6}$$

which is identical to (5.2) because $\widehat{\mathbf{P}}_j'\widehat{\mathbf{C}}' = \mathbf{0}$ by definition.

5.2 Merging with Single-View Overlap

Given a set of corresponding structure points $\widehat{\mathbf{X}}_i \leftrightarrow \widehat{\mathbf{X}}_i'$, the most obvious way to constrain $\mathbf{H}(\mathbf{v})$ would be to use (5.3) directly by minimizing

$$\sum_i D_E(\mathbf{H}(\mathbf{v})\widehat{\mathbf{X}}_i, \widehat{\mathbf{X}}_i')^2, \tag{5.7}$$

where $D_E(\mathbf{a}, \mathbf{b})$ is the *inhomogeneous* Euclidean distance. However, Euclidean distance is not preserved under the projective ambiguity, and attempting to minimize Euclidean distance in a projective space would be truly meaningless [Fitzgibbon and Zisserman, 1998].

As an example, two points might be measured as having a distance of 1 (with some arbitrary units) in the projective space when in fact the actual distance between those points after metric rectification should be ∞. Thus, in order to make the distance measurement in (5.7) meaningful, it would be necessary to first autocalibrate the right reconstruction (or the left reconstruction if $\mathbf{H}^{-1}$ is estimated instead).

A second complication is that the structure points in (5.7) are homogeneous, but the Euclidean distance is measured between inhomogeneous points which means that (5.7) will require a nonlinear minimization. However, this is not a serious complication because a good initialization can be obtained linearly by minimizing algebraic (rather than Euclidean) distance, as discussed in both Laveau [1996] and Fitzgibbon and Zisserman [1998].

If both left and right reconstructions have been autocalibrated, then $\mathbf{H}$ should theoretically be a similarity transform with 7 dof. The optimal estimation of a similarity transform between two corresponding point sets, called the absolute orientation problem, can be performed linearly [Horn et al., 1988; Umeyama, 1991; Matei and Meer, 1999], and this is perhaps the most commonly used approach to merging (see Repko and Pollefeys [2005]; Farenzena et al. [2009]; Frahm et al. [2010]).

However, because autocalibration is not a very well posed problem that cannot be solved perfectly (especially for smaller reconstructions), there is always some projective ambiguity remaining which means that the alignment between two autocalibrated reconstructions is not truly a similarity transformation.

Another problem that applies to merging in either projective or metric spaces is that structure points generally have a large degree of uncertainty, and this causes the constraints of (5.3) to be poorly satisfied even for the best choice of $\mathbf{H}$. The approach of Matei and Meer [1999] partially deals with this problem by taking into account the approximate uncertainty of structure points in the absolute orientation problem; however, we prefer to overcome the root of this problem.

5.2.1 Nister's Linear Method

A more attractive solution is to measure distance in image space, because image space is already a metric space. In other words, if $\tilde{\mathbf{x}}_i^j$ is the measured observation of $\mathbf{X}_i$ in view j, then one could instead minimize

$$\sum_{i,j} D_E(\widehat{\mathbf{P}}'_j \mathbf{H}(\mathbf{v}) \widehat{\mathbf{X}}_i, \tilde{\mathbf{x}}_i^j)^2. \tag{5.8}$$

Not only does (5.8) bypass the issue of measuring error in projective spaces, but for most configurations it is fairly robust to structure points that have a large degree of uncertainty in their depth, because re-projecting the structure point to measure distance on the image plane largely cancels out the uncertainty as long as the views in the right reconstruction are relatively close to the views in the left reconstruction.

Although there is no linear solution to (5.8), an algorithm was given in Nister [2001a, p. 65] that minimizes a *very similar* problem linearly. Although it is not explicitly mentioned there, that problem is

$$\sum_{i,j} D_A(\widehat{\mathbf{P}}'_j \mathbf{H}(\mathbf{v}) \widehat{\mathbf{X}}_i, \hat{\mathbf{x}}_i^j)^2, \tag{5.9}$$

where $D_A(\mathbf{a}, \mathbf{b})$ is the *inhomogeneous* algebraic distance, and $\hat{\mathbf{x}}_i^j$ is the closest point to $\tilde{\mathbf{x}}_i^j$ that can be found by varying $\mathbf{v}$. The details of this method are given below.

Let the index of the overlapping view be denoted by o. Then the epipolar line $\mathbf{l}_i^j$ in the jth view that contains the image of the ith point and the epipole of the oth view is given by

$$\mathbf{l}_i^j = \widehat{\mathbf{P}}'_j {\widehat{\mathbf{P}}'_o}^+ \widehat{\mathbf{P}}_o \widehat{\mathbf{X}}_i \times \widehat{\mathbf{P}}'_j \widehat{\mathbf{C}}'_o. \tag{5.10}$$

This may be verified as follows. To the right of the cross product, $\widehat{\mathbf{C}}'_o$ is the camera center of the oth view in the right reconstruction, and projecting by $\widehat{\mathbf{P}}'_j$ gives its epipole in the jth view. On the left hand side, we start with $\widehat{\mathbf{X}}_i$, the ith point in the left reconstruction. We multiply by $\widehat{\mathbf{P}}_o$ to get the image of this point in the overlap view, then by ${\widehat{\mathbf{P}}'_o}^+$ to back-project this to a structure point in the right reconstruction, and finally by $\widehat{\mathbf{P}}'_j$ to obtain an image in the jth view. The cross product of two points gives the line joining those points, so $\mathbf{l}_i^j$ is the desired epipolar line.

The closest point on this epipolar line to the measurement $\tilde{\mathbf{x}}_i^j$ is then given by

$$\hat{\mathbf{x}}_i^j = [\mathbf{l}_i^j]_\times [\tilde{\mathbf{x}}_i^j]_\times \Omega_\infty^* \mathbf{l}_i^j, \tag{5.11}$$

where $\Omega_\infty^* = \mathrm{diag}(1, 1, 0)$ is the absolute dual conic in a metric frame, and $[\mathbf{x}]_\times$ is the 3×3 skew-symmetric cross product matrix of $\mathbf{x}$.

Taking $\widehat{\mathbf{X}}_i$, the ith point in the left reconstruction, multiplying by $\mathbf{H}(\mathbf{v})$ should transform it into the right reconstruction, and then multiplying by $\widehat{\mathbf{P}}'_j$ gives its image in the jth view, which should be $\hat{\mathbf{x}}_i^j$. This is a homogeneous equivalence constraint that implies a zero cross product,

$$\begin{aligned}
\mathbf{0} &= \hat{\mathbf{x}}_i^j \times \widehat{\mathbf{P}}'_j \mathbf{H}(\mathbf{v}) \widehat{\mathbf{X}}_i && (5.12)\\
&= [\hat{\mathbf{x}}_i^j]_\times \widehat{\mathbf{P}}'_j ({\widehat{\mathbf{P}}'_o}^+ \widehat{\mathbf{P}}_o + \widehat{\mathbf{C}}'_o \mathbf{v}^\mathsf{T}) \widehat{\mathbf{X}}_i && (5.13)\\
&= [\hat{\mathbf{x}}_i^j]_\times \widehat{\mathbf{P}}'_j {\widehat{\mathbf{P}}'_o}^+ \widehat{\mathbf{P}}_o \widehat{\mathbf{X}}_i + [\hat{\mathbf{x}}_i^j]_\times \widehat{\mathbf{P}}'_j \widehat{\mathbf{C}}'_o \widehat{\mathbf{X}}_i^\mathsf{T} \mathbf{v}. && (5.14)
\end{aligned}$$

Rearranging and left-multiplying by $([\hat{\mathbf{x}}_i^j]_\times \widehat{\mathbf{P}}'_j \widehat{\mathbf{C}}'_o)^\mathsf{T}$, a single linear constraint on $\mathbf{v}$ is obtained,

$$[\hat{\mathbf{x}}_i^j]_\times \widehat{\mathbf{P}}'_j \widehat{\mathbf{C}}'_o \widehat{\mathbf{X}}_i^\mathsf{T} \mathbf{v} = -[\hat{\mathbf{x}}_i^j]_\times \widehat{\mathbf{P}}'_j \widehat{\mathbf{P}}'^{+}_o \widehat{\mathbf{P}}_o \widehat{\mathbf{X}}_i \tag{5.15}$$

$$\widehat{\mathbf{X}}_i^\mathsf{T} \mathbf{v} = \frac{-\widehat{\mathbf{C}}'^{\mathsf{T}}_o \widehat{\mathbf{P}}'^{\mathsf{T}}_j [\hat{\mathbf{x}}_i^j]_\times^2 \widehat{\mathbf{P}}'_j \widehat{\mathbf{P}}'^{+}_o \widehat{\mathbf{P}}_o \widehat{\mathbf{X}}_i}{\left|\left| [\hat{\mathbf{x}}_i^j]_\times \widehat{\mathbf{P}}'_j \widehat{\mathbf{C}}'_o \right|\right|^2}. \tag{5.16}$$

Thus, each correspondence between a structure point in the left reconstruction and an image of that point in the right reconstruction in any view other than the overlapping view provides a linear constraint. A total of at least four such constraints are needed to constrain $\mathbf{H}(\mathbf{v})$.

5.3 Symmetric Linear Merging

In most cases, Nister's linear algorithm will work well. However, if the baselines between *all* views in the left reconstruction are relatively small, then the structure points that are chosen from the left reconstruction will have a large degree of uncertainty in their depth. If in addition, the baselines between views in the right reconstruction are *not all* small, then this large uncertainty in depth may cause the projection of those points into the right reconstruction to be very inaccurate (see Fig. 5.2), and this can result in a failure to properly merge the reconstructions.

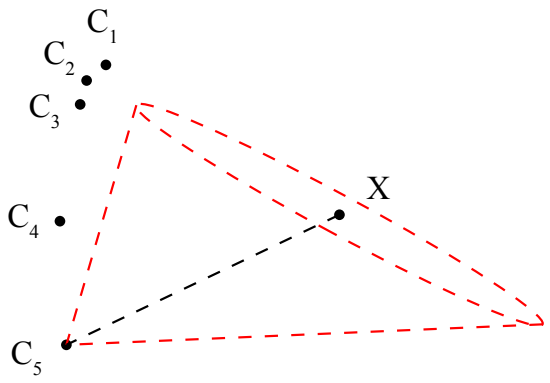


Figure 5.2. Example of a configuration that may result in unstable merge using Nister's linear algorithm. The true location of the five cameras are indicated by $\widehat{\mathbf{C}}_i$, $i = 1 \ldots 5$. The true location of a structure point $\mathbf{X}$ is also marked. The left reconstruction consists of views $\{1, 2, 3\}$ and the right reconstruction consists of views $\{3, 4, 5\}$. Because all views in the left reconstruction are relatively close together, there is a large degree of uncertainty in the triangulation of any structure point $\mathbf{X}$, indicated by the dotted red ellipse. As a result, the projection of this triangulated point into the 5th view may be far from the measured image point, causing the merging constraint to be bad and preventing the algorithm from identifying a good merging homography.

In this case, one notices that structure points in the right reconstruction would have much less uncertainty in their depth (because the baselines are not all small), and hence these points could be merged into the left reconstruction and projected onto the image plane where they would correctly match up with the measured image points.

In practice, a method of keyframe selection (see, for example Torr [2002]; Pollefeys et al. [2002a]; Repko and Pollefeys [2005]; Beder and Steffen [2006]) should be used to ensure that there is some sufficient baseline between each view. However, the fact remains that between any two reconstructions that one wishes to merge, one of them will have wider baseline than the other, and because Nister's linear algorithm is asymmetric, the structure points that are used for merging constraints should be chosen from the side that has wider baseline in order to achieve the greatest accuracy.

It is possible (albeit messy) to derive constraints on $\mathbf{v}$ from structure points in the right reconstruction corresponding to image points in the left reconstruction, and combining these with the constraints from (5.16) would allow $\mathbf{v}$ to be estimated linearly from a set of symmetric constraints. However, this type of symmetry would not actually be a good thing because the constraints from one direction always have higher error, so a single symmetric estimation would only be as good as the constraints of the lesser direction.

Therefore, our solution is to always apply the algorithm in the forward direction (as described in Section 5.2.1) as well as the reverse direction, and take the solution that results in lower reprojection error. In the reverse direction we use structure points from the right reconstruction corresponding to image points in the left reconstruction, and solve for $\mathbf{H}^{-1}$ parameterized by $\mathbf{v}'$,

$$\mathbf{H}^{-1}(\mathbf{v}') = \widehat{\mathbf{P}}_j^{+}\widehat{\mathbf{P}}'_j + \widehat{\mathbf{C}}\mathbf{v}'^{\mathsf{T}}, \tag{5.17}$$

where $\widehat{\mathbf{C}}$ is the null space of $\widehat{\mathbf{P}}_j$. Using the overlap view, and substituting (5.5) into (5.17), we can then compute $\mathbf{v}$ from $\mathbf{v}'$,

$$\widehat{\mathbf{P}}'^{+}_o\widehat{\mathbf{P}}_o + \widehat{\mathbf{C}}'_o\mathbf{v}^{\mathsf{T}} = \left(\widehat{\mathbf{P}}_o^{+}\widehat{\mathbf{P}}'_o + \widehat{\mathbf{C}}_o\mathbf{v}'^{\mathsf{T}}\right)^{-1} \tag{5.18}$$

$$\widehat{\mathbf{C}}'_o\mathbf{v}^{\mathsf{T}} = \left(\widehat{\mathbf{P}}_o^{+}\widehat{\mathbf{P}}'_o + \widehat{\mathbf{C}}_o\mathbf{v}'^{\mathsf{T}}\right)^{-1} - \widehat{\mathbf{P}}'^{+}_o\widehat{\mathbf{P}}_o \tag{5.19}$$

$$\mathbf{v}^{\mathsf{T}} = \frac{\widehat{\mathbf{C}}'^{\mathsf{T}}_o}{||\widehat{\mathbf{C}}'_o||^2}\left(\left(\widehat{\mathbf{P}}_o^{+}\widehat{\mathbf{P}}'_o + \widehat{\mathbf{C}}_o\mathbf{v}'^{\mathsf{T}}\right)^{-1} - \widehat{\mathbf{P}}'^{+}_o\widehat{\mathbf{P}}_o\right). \tag{5.20}$$

This still allows us to merge the right reconstruction into the left reconstruction even when using constraints from the reverse direction.

5.4 Structure Invariant Maximum Likelihood Merging

Once the merging homography has been found, structure points can be retriangulated from all views to obtain a point that is more accurate than the corresponding points previously existing in the left or right partial reconstructions. Thus, the ideal merging homography should seek to maximize the accuracy of the cameras without reference to the previously triangulated structure points.

Although the symmetric modification improves the reliability of Nister's linear method to obtain a better initial estimate of $\mathbf{H}$, it is still less than ideal; in particular, it attempts to minimize distance between the projection of a structure point and an artificial point $\hat{\mathbf{x}}_i^j$ rather than the actual image measurement $\tilde{\mathbf{x}}_i^j$, it minimizes an algebraic rather than Euclidean distance, and most importantly it is not completely invariant to the uncertainty in previously triangulated structure points.

Therefore, we seek the homography that would maximize the likelihood of the overall reconstruction after retriangulating all structure points from the merged camera matrices using a maximum likelihood method. Assuming measurement noise is Gaussian, it is well known that maximizing likelihood is equivalent to minimizing reprojection error [Hartley and Zisserman, 2004, p.102], and therefore the solution is given by

$$\widehat{\mathbf{v}}_{ML} = \underset{\mathbf{v}}{\operatorname{argmin}} \sum_{i,j} D_E(\tilde{\mathbf{x}}_i^j, \overline{\mathbf{P}}_j \widehat{\mathbf{X}}_{MLi})^2, \tag{5.21}$$

where $\overline{\mathbf{P}}_j(\mathbf{v})$ are the merged projection matrices as a function of $\mathbf{v}$,

$$\overline{\mathbf{P}}_j(\mathbf{v}) = \begin{cases} \widehat{\mathbf{P}}_j, & j \in \mathcal{L}, \\ \widehat{\mathbf{P}}'_j \mathbf{H}(\mathbf{v}), & j \notin \mathcal{L}, \end{cases} \tag{5.22}$$

and $\widehat{\mathbf{X}}_{MLi}$ is the maximum likelihood triangulation of the ith structure point from all available image measurements $\tilde{x}_i^j$ with respect to the merged cameras, $\overline{\mathbf{P}}_j$.

For points visible in just two views, we compute the maximum likelihood triangulation in closed form as in Hartley and Sturm [1997]; for more than two views, we compute the maximum likelihood triangulation by nonlinearly minimizing the sum of squared reprojection errors using Levenberg-Marquardt [Marquardt, 1963] from the homogeneous linear initialization [Hartley and Zisserman, 2004, p. 313]. In order to minimize (5.21) with respect to $\mathbf{v}$, we initialize using our symmetric linear correction to Nister's method and then use Levenberg-Marquardt with numerical differentiation.

It should be noted that even though (5.21) provides a maximum likelihood estimate of the merging homography, no estimate of the merging homography will produce a maximum likelihood reconstruction. Therefore, we always follow up merging with bundle adjustment [Triggs

et al., 2000; Lourakis and Argyros, 2004], the maximum likelihood nonlinear improvement of a projective reconstruction.

Of course, one could skip (5.21) and proceed directly to bundle adjustment after using the linear initialization. However, for a system of n points and m views projective bundle adjustment has $12m + 3n$ parameters, and in a typical problem there may be hundreds of thousands of free parameters, making bundle adjustment not only computationally expensive but very susceptible to falling into local minima. Projection matrices in bundle adjustment are almost always parameterized using an absolute coordinate system, and as a result a very small error in the merging homography $\mathbf{H}$ could necessitate rather significant changes to half of the views during the subsequent bundle adjustment. In contrast, $\mathbf{H}$ has only 15 dof, so it will be more efficient and reliable to optimize $\mathbf{H}$ as much as possible prior to bundle adjustment.

5.5 Robustness to Outliers

Because the measurements $\tilde{x}_i^j$ are obtained using a correspondence finding algorithm on images, there are likely to be some mismatches that result in outliers with very large error. These outliers violate the assumed Gaussian noise model, and it is therefore important to detect and ignore these measurements in order to make a robust estimate of $\mathbf{H}$ using (5.21).

We use the RANSAC [Fischler and Bolles, 1981] paradigm to handle outliers, specifically MSAC [Torr and Zisserman, 2000]. From the set of structure point correspondences $\widehat{\mathbf{X}}_i \leftrightarrow \widehat{\mathbf{X}}'_i$, our objective is to find the largest sample consensus of correspondences that agree upon a homography which can merge the partial reconstructions while keeping the reprojection error of all retriangulated structure points $\overline{\mathbf{X}}_i$ in (5.21) below some threshold τ.

This is done by picking random subsets from the set of correspondences and then minimizing (5.21) (initialized with the symmetric linear method) using *only* the selected subset of correspondences. From this estimate of $\mathbf{v}$ we then enlarge the subset to include all inliers and repeat within the RANSAC framework to find the largest sample consensus. Finally, we iteratively re-minimize (5.21) and re-classify inliers until convergence.

In general, the minimum number of correspondences that must be used in a random sampling is data dependent because the number of constraints that are provided by each correspondence depends on the number of images that a structure point is viewed in. However, we do not aim to use minimal subsets because a greater robustness to noise is achieved by using larger subsets. When using triplet correspondences to span the overlap view, we use a subset size of 10 and this results in 10 constraints on $\mathbf{v}$, which we find provides a good balance between speed of convergence and robustness to noise.

5.6 Merging Correspondences

After merging two partial reconstructions into a larger reconstruction with more views it may be possible to triangulate structure points using the additional views for greater accuracy, if the image measurements exist. For example, if the left reconstruction contains views $\{1, 2, 3\}$ and the right reconstruction contains views $\{3, 4, 5\}$, and one has good measurements for $\tilde{\mathbf{x}}_i^j$, $j = 1 \ldots 5$, then after merging an estimate of the point $\mathbf{X}_i$ can be made using all five views that will be more accurate than the estimate of that point in the original left or right reconstructions.

It is typical to detect correspondences in a separate module, either using feature tracking [Zach et al., 2008; Hwangbo et al., 2009; Kim et al., 2009] or inter-frame matching, that produces as output an increasing list containing the coordinates of an observed feature point in a series of views. However, despite attempts to remove outliers [Shi and Tomasi, 1994; Tommasini et al., 1998; Fusiello et al., 1999], some of these measurements will still be erroneous.

In this example, suppose that $\tilde{\mathbf{x}}_i^4$ is a bad measurement. Thus, it is likely that the point $\widehat{\mathbf{X}}_i$ will exist in the left reconstruction, but $\widehat{\mathbf{X}}_i'$ will not exist in the right reconstruction. After merging the two reconstructions together and attempting to triangulate a new point using all the images of this point, this will also fail and hence the ith structure point will be lost from the merged reconstruction even though $\widehat{\mathbf{X}}_i$ was formerly a well-triangulated point.

As feature tracks are increased in length, the probability of an outlier match increases, so it is important to have a method of preventing all the good points from eventually being thrown out when merging together partial reconstructions. Similar to Thormahlen et al. [2008], our solution to this problem is to associate measurements independently with each partial reconstruction, and then attempt to merge these correspondences when the partial reconstructions are merged.

In our implementation we have not used feature tracking to find the initial correspondences but rather we have used wide-baseline matching of Harris and Stephens [1988] corner points. To compute the initial triplet reconstructions we search for correspondences of triplets, and to merge them together we use a separate set of triplet correspondences. After merging the two projective reconstructions we search for structure points between the left and right reconstruction that can be merged.

In order to identify potential structure points for merging we project all of the structure points from the right reconstruction onto the image plane of the overlapping view. These image points are stored in a uniform grid structure [Bentley, 1975] that allows all points in a fixed radius to be found rapidly. For each structure point in the left reconstruction, we search for potential matches around its projected image point in the overlap view, and for each potential match we compute the maximum likelihood triangulation from the set of merged correspondences associated with those points. If the triangulated point reprojects back to all of the original measurements within some small threshold, then the merge is adopted.

5.7 Results

We compare the various merging approaches on synthetic data with controlled levels of noise so that the statistical differences between algorithms is made clearly apparent. In our synthetic tests, five cameras are generated on a circle of radius 100 units looking approximately toward the origin (± 20 units). The angular separation between successive cameras is uniformly distributed in the range of $0.1^\circ - 10^\circ$, and camera focal length is uniformly distributed in the range of $600 - 800$ units.

For scene structure, 100 points are generated on the surface of a cube of width 100 units that is shifted some distance from the origin in the direction of the average camera principal ray. A top down view of a generic synthetic configuration is shown in Fig. 5.3. Correspondences are generated by projecting the true structure points onto the image plane and adding normally distributed noise to simulate measurement error in the correspondence finder.

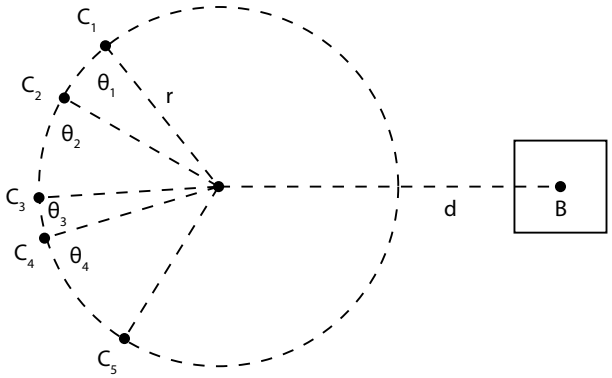


Figure 5.3. Top down view of a synthetic configuration. Cameras centers $\mathbf{C}_1, \ldots, \mathbf{C}_5$ are located on a circle of radius r with random angular separations of $\theta_1, \ldots, \theta_4$. The structure points are generated on the surface of a cube centered at $\mathbf{B}$, a distance d from the origin.

From these noisy correspondences we compute a robust estimate of the trifocal tensor for the first three views and the last three views (so that there is one view of overlap). We then attempt to merge these two reconstructions into a single reconstruction covering all five views using: (a) the optimal absolute orientation method of [Umeyama, 1991], after autocalibrating both partial reconstructions using the recent method of [Gherardi and Fusiello, 2010]; (b) Nister's forward linear method (Section 5.2.1); (c) our symmetric variation on Nister's method (Section 5.3); (d) the proposed Structure Invariant Maximum Likelihood (SIML) method (Section 5.4). We evaluate merging success for any particular trial by using the mean reprojection error of the merged result prior to bundle adjustment, because it is well known that the maximum

likelihood projective reconstruction should minimize reprojection error.

In our first experiment we examined sensitivity to noise. This was done by generating noisy correspondences from 100 random configurations at each level of noise and then looking at the median of the mean reprojection error (see Fig. 5.4). We observe that all methods have zero median error in the absence of noise, but the absolute orientation method is extremely sensitive and produces high median errors even under low noise conditions. In contrast, the image-spaced approaches exhibit reconstruction error that is almost proportional to the measurement error. Our symmetric linear method has lower median error than Nister's method, and the proposed SIML improvement has lower median error still, although these reductions to median error are relatively marginal.

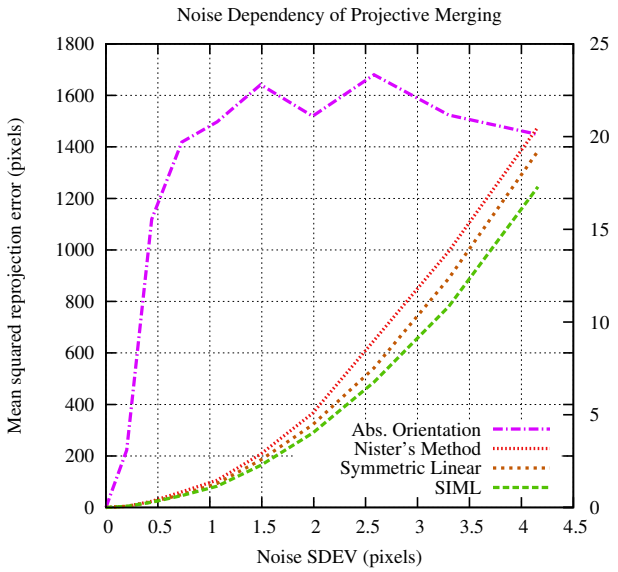


Figure 5.4. Comparison of reconstruction quality as a function of measurement noise. Scene distance is fixed at 500 units. Errors for the absolute orientation method are shown on the primary axis and errors for the other methods are shown on the secondary axis. The plotted curves are the median of 100 trials. All methods have zero median error in the absence of noise, but the absolute orientation method is extremely sensitive and produces high median errors even under low noise.

In our second experiment, we fix the noise level at $\sigma = 1$ pixels and examine the merged reconstruction quality as a function of scene distance (Fig. 5.5). Curiously, we observe that the reprojection error of the merged reconstruction is not a monotonic function of scene distance. This effect, while initially perplexing, can be attributed to two conflicting forces. On the one hand, the absolute (3D) error is increased for more distant reconstructions because measurement noise (which is added in image space) becomes relatively greater. On the other hand,

more distant scenes tend to have lower reprojection error because the projection of the entire scene bounding box occupies a smaller portion of the image. Thus, as distance of the scene is increased, the merged reprojection error will gradually increase until it becomes no better than random guessing, at which point the reprojection error will gradually reduce and asymptote at some small value, although the absolute errors continue to increase.

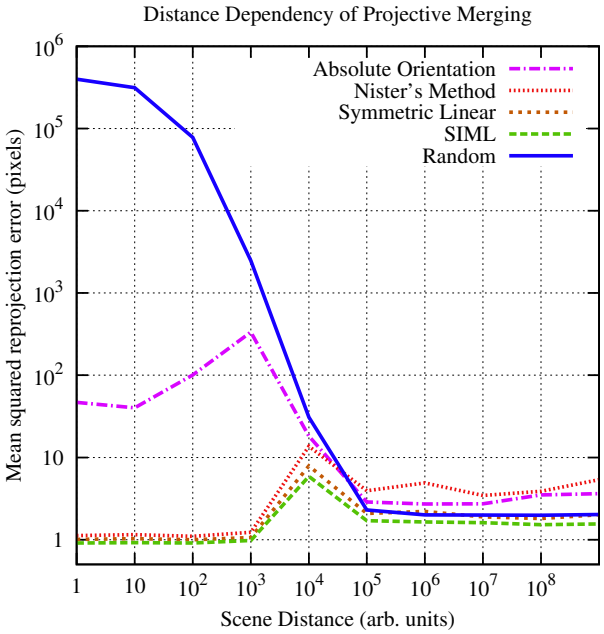


Figure 5.5. Comparison of reconstruction quality as a function of scene distance. Measurement noise is fixed at $\sigma = 1$ pixels. Interestingly, the reprojection error is not a monotonic function of scene distance. The plotted curves are the median of 100 trials. For this distribution of configurations, we see that the median absolute orientation method performs better than random when the scene distance is within 10^3, whereas the image-space methods perform better than random when the scene distance is less than 10^4, but they only provide accurate results out to 10^3. The absolute orientation method does not provide accurate results for any distance at this level of noise.

In order to approximate this overall downward trend in expected reprojection error for more distant geometry, we have plotted the median of the mean reprojection error obtained by randomizing the order of structure points in the true configuration and measuring the distance to the (incorrect) image projections after projecting those structure points by the true projection matrices. In other words, this shows the expected reprojection error if structure points were to be randomly chosen within the scene volume rather than being precisely triangulated.

We see from the graph that the median performance of the absolute orientation method is better than random only when the scene distance is within 10^3, whereas the image-space methods perform better than random when the scene distance is less than 10^4, but they only

provide accurate results out to 10^3. Between these methods, the proposed SIML method has the lowest median error, although the improvements to median performance are still only marginal. The absolute orientation method does not provide accurate results for any distance at this level of noise.

For extremely far distances (10^5 and beyond in this case), the random curve actually performs slightly better than the proposed methods. This is because at this extreme distance, the prior knowledge of the true scene bounding box that is assumed by the random method becomes more informative than the image measurements, because the large relative noise causes the uncertainty ellipsoid of a triangulated structure point to become larger than the true scene bounding box.

It should be noted that there is nothing magic about the number 10^3, it is simply the point at which the signal to noise ratio becomes too small for accurate reconstruction, and this is dependent on the specific camera configuration (particularly the distance between cameras) as well as the correspondence measurement noise.

We compare the Empirical Cumulative Distribution Functions (ECDFs) of the mean squared reprojection error (MSE) for each method of merging using from a set of 1000 random configurations with noise fixed at $\sigma = 2$ pixels and scene 'distance' set to zero (i.e., the scene being centered at the origin) in Fig. 5.6.

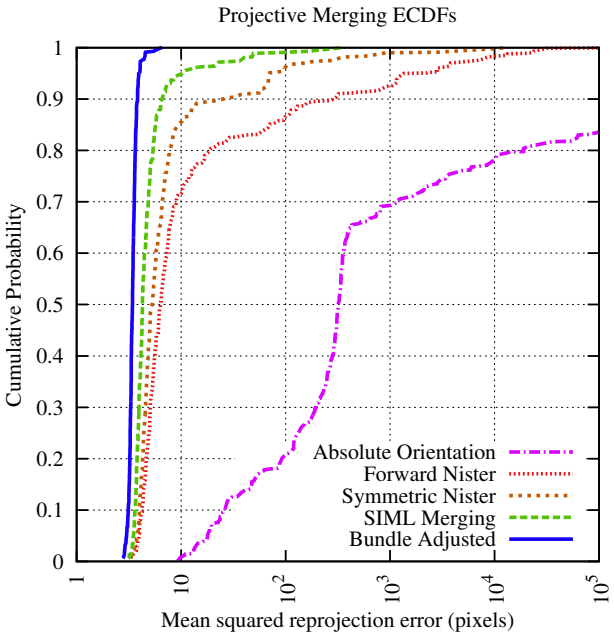


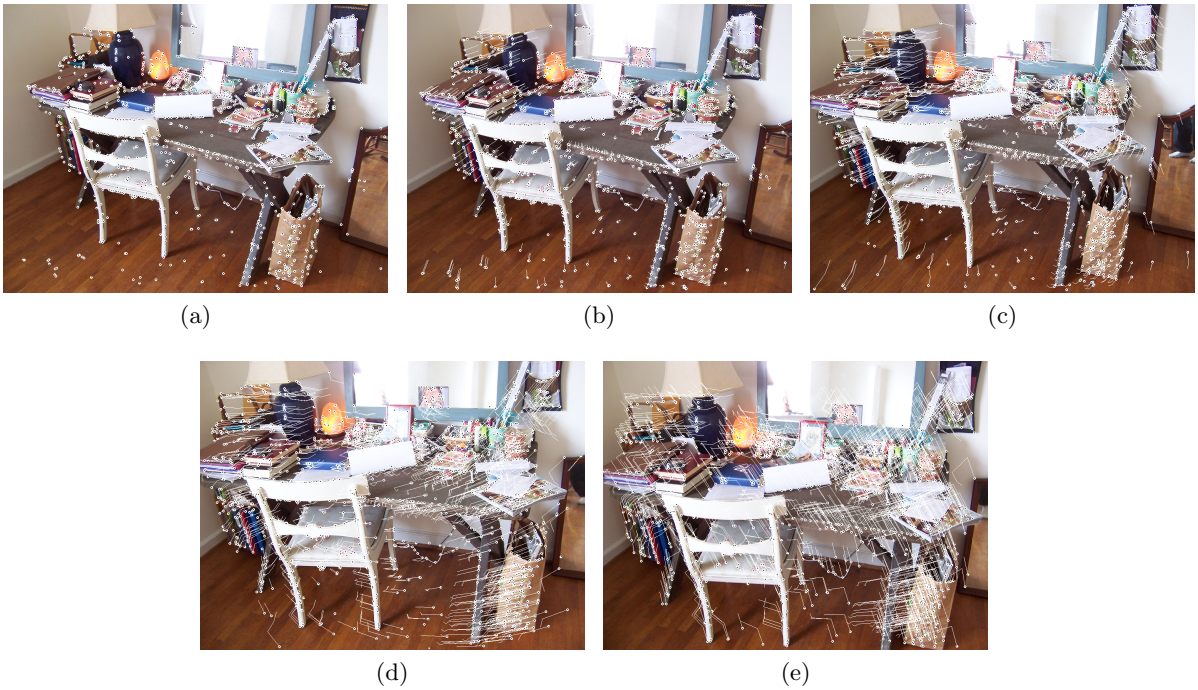
Figure 5.6. Comparison of the empirical cumulative distribution of mean squared reprojection error of the merged reconstructions using various merging methods, based on merging from 1000 randomly generated configurations.

By looking at the complete performance distribution, we finally see that the improvements offered by our symmetric and SIML improvements are significant when it comes to tail performance. This is expected, because the improvements are primarily designed to increase robustness under conditions of small baseline, but the majority of configurations that we randomly generate will not have the problem of small baseline.

Specifically, using Nister's original method the 90th percentile of mean squared reprojection error was 300 pixels, whereas the 90th percentile was reduced to 29 pixels after our symmetric modification, and further reduced down to just 6.5 pixels using the SIML method. Finally, after bundle adjusting the result, the 90th percentile error was reduced to 3.8 pixels, and was never worse than 6.6 pixels. Although this may still seem like a large amount, it is actually very good considering that this was the worst case error over all merging trials, and a total of 200,000 noise values were generated with $\sigma = 2$ pixels, so the input noise is expected to have exceeded 6.6 pixels approximately 387 times.

Finally, we demonstrate an example of the robust SIML merging method on some measurements gathered from real data. We started with a series of five sequential snapshots of a desk and then proceeded to find correspondences by matching corners. We computed two estimates of the trifocal tensor robustly and then merged them together using the proposed robust maximum likelihood method. The correspondences were merged as described in Section 5.6 and then bundle adjustment was used to nonlinearly improved the merged result.

In this reconstruction we found a total of 2,661 structure points with a mean squared reprojection error of 0.55 pixels, all of which were below the threshold of 2 pixels used within the RANSAC framework. We show a selection of three views from the merged result of five views in Fig. 5.7, where we have drawn the reprojected structure points in comparison to the original corner points to demonstrate the low reprojection error visually. To reduce visual clutter, we only draw the reprojections of points that were merged so that they have an image in all five views. Some views of the reconstructed point cloud are shown in Fig. 5.8.



(a) (b) (c)

(d) (e)

Figure 5.7. A reconstruction of five views that was formed by merging two triplets that overlap by one view using the proposed approach. The merged reconstruction consists of 3,367 structure points with a mean squared reprojection error of 0.51 pixels. The image width is 1000 pixels. The white tracks show image measurements and the black points are the reprojected structure points.

(a) (b)

Figure 5.8. Two views of the structure points in the merged desk reconstruction. (a) from a side perspective, and (b) from a front perspective. The reconstructed cameras are shown as pyramids.

5.8 Conclusions

Merging of partial reconstructions requires the computation of the transformation matrix that aligns their respective spaces. This is commonly solved as an absolute orientation problem in metric space after autocalibration. However, autocalibration is an inherently sensitive procedure that we feel is best delayed until the reconstruction becomes larger and more precise in order to avoid instabilities. Moreover, the absolute orientation approach is very sensitive to the accuracy of structure points that have been already triangulated in the partial reconstructions.

By using Nister's method, errors can be measured in metric image space, thereby avoiding the need to perform premature autocalibration, which also has the advantage of canceling out the majority of structure point uncertainty, thereby reducing sensitivity to noise. However, this uncertainty is never fully canceled out. To solve this problem, we have proposed a way to apply Nister's method symmetrically and thereby provide a more reliable initialization. Most importantly, we have also proposed a maximum likelihood method that is completely invariant to the uncertainty in structure points, and shown how this can be used within a RANSAC framework to obtain truly robust results.

Thus, this new merging method can be used to increase the accuracy and reliability of any projective structure from motion system that relies on a merging operation.

Chapter 6

Autocalibration

Autocalibration is the process of determining the homography that rectifies a projective reconstruction to metric, bringing it within a similarity transformation of the true configuration. Because metric constraints cannot usually be enforced in the initial uncalibrated reconstruction, it is necessary to compute a projective reconstruction first and then autocalibrate it. Although autocalibration is a well-studied problem, previous approaches have relied upon heuristic objective functions that are sensitive to noise. We propose a maximum likelihood objective and show that it can be implemented robustly and efficiently, and often provides substantially greater accuracy especially when there are fewer views or greater noise.

A metric reconstruction is one that differs from the true configuration only by the choice of coordinate system; in other words, there is some unknown rotation, translation and scale [Hartley and Zisserman, 2004]. It is well known that metric reconstruction is not possible from projective constraints alone [Faugeras, 1992; Hartley, 1992; Hartley and Zisserman, 2004] because the solution is ambiguous up to multiplication by some arbitrary homography. Thus, a reconstruction obtained from projection constraints alone is referred to as a projective reconstruction.

Given any additional constraints on the intrinsic camera parameters (e.g., that the images are not skewed, that pixel aspect ratio is known, that the center of projection is in the center of the image, or that multiple images were produced by the same physical camera), the ambiguity can be resolved. However, incorporating these constraints directly into an initial estimate is difficult: an efficient solution is only possible in the simplest minimal case of two views with fully calibrated cameras [Nistér, 2004]. A solution for two uncalibrated cameras was recently proposed in Hartley and Kahl [2009], although it was admittedly computationally impractical.

In contrast, techniques for computing a projective reconstruction are much more efficient, so the usual approach is to compute an initial projective reconstruction minimally or linearly, which can then be refined to a maximum likelihood (ML) projective reconstruction using bundle

adjustment [Triggs et al., 2000; Hartley and Zisserman, 2004; Lourakis and Argyros, 2004], and finally rectified by multiplying the reconstruction by a homography that causes metric constraints to be approximately satisfied. The estimation of this rectifying homography is known as autocalibration (a.k.a. self-calibration).

In projective bundle adjustment, the cameras are parameterized by projection matrices and the projection equation is simple, with the only nonlinearity being due to the perspective division. The basin of attraction for projective bundle adjustment is relatively large, and convergence is fast and reliable. In contrast, the presence of rotation matrices significantly complicates the projection equation in metric bundle adjustment, making the linearized update approximations less accurate. As a result, we observe a much smaller basin of attraction with less reliable convergence. Thus, even when a metric estimate is available one might still prefer to do bundle adjustment in projective space, although this would necessitate the use of autocalibration to map the result back into a metric space.

A plethora of approaches to autocalibration have been presented in the literature (see Section 6.1), but autocalibration has a reputation for instability, and obtaining robust results in the presence of estimation error can often be difficult. This has motivated a recent trend towards approaches that use global optimization methods. However, the objectives that are optimized by these global approaches are still heuristic and sensitive to noise.

In this paper, we formulate a maximum likelihood objective for autocalibration and show that it can be optimized efficiently and robustly. Using the maximum likelihood method avoids the sensitivity to noise and can give considerably more accurate results, especially for small numbers of views or high levels of relative noise (equivalent to more distant point clouds) where the autocalibration problem is more difficult.

We begin by summarizing past work in autocalibration (Section 6.1), and then derive our maximum likelihood objective (Section 6.2). We show how this concise formulation properly accounts for all previously recognized constraints (Section 6.2.2), and explain the fundamental instability and limitations behind the heuristic maximum *a priori* objectives (Section 6.3).

A method for efficient optimization is described (Section 6.4), along with a novel resectioning step to ensure that metric constraints are enforced exactly. We identify a set of representative autocalibration algorithms to compare against (Section 6.5), and devise a framework for objective experimental comparison (Section 6.6). Finally, we present the results of our experiments (Section 6.7), demonstrating the efficiency, robustness and quality of the proposed ML method, and offer some concluding remarks (Section 6.8).

6.1 Background

The first known method of autocalibration was based on the Kruppa equations [Kruppa, 1913; Faugeras et al., 1992; Vieville and Lingrand, 1996; Hartley, 1997a], now understood to be an algebraic representation of the correspondence of epipolar lines tangent to the dual image of a conic (DIAC).

It was shown in Huang and Faugeras [1989] that an equivalent constraint to the Kruppa equations is that the essential matrix between any view pair must have two equal non-zero singular values, called the rigidity constraint. This is the fundamental principle behind several autocalibration approaches that theoretically work for two views when focal length is the only unknown [Hartley, 1992; Mendonça and Cipolla, 1999; Bougnoux, 1998; Lao et al., 2004; Gao and Radha, 2004], although they are highly sensitive to noise.

When more than two views are considered, autocalibration via the Kruppa equations requires finding the simultaneous solutions to many quadratic equations, which has not been regarded as a promising approach [Hartley and Zisserman, 2004], but has been attempted using homotopy continuation [Luong, 1992], nonlinear methods [Zeller, 1996; Luong and Faugeras, 1997], and more recently using globally convergent interval analysis [Fusiello et al., 2004]. Because the Kruppa equations do not enforce all of the calibration constraints that are now understood, such as the common support plane for the plane at infinity, these methods are subject to singularities that can lead to instabilities.

In Pollefeys et al. [1996], the modulus constraint on the plane at infinity was introduced, which is complementary to the Kruppa equations because it enforces constraints on the common plane at infinity without enforcing constraints on the DIAC. A unifying framework for these entities was presented with the absolute dual quadric (ADQ) [Triggs, 1997], a fixed entity in space that encodes for both the plane at infinity and absolute dual conic (ADC) and projects to the DIACs. The ADQ is useful because all autocalibration constraints can be translated onto it.

The ADQ can be estimated using linear and nonlinear least squares [Triggs, 1997; Pollefeys et al., 1998; Hartley and Zisserman, 2004; Oliensis, 1999; Ponce, 2001], sometimes weighted according to prior assumptions as in Pollefeys et al. [2002b]. Unfortunately, both of these variations are often unstable in practice [Bougnoux, 1998]. It has been commented [Bocquillon et al., 2007] that the main reason for instability of the linear method is that the rank and positive-semidefinite constraints of the ADQ are not enforced. However, we believe that the greater issue with the linear method is that the constraint equations do not directly correspond to the parameters they are intended to constrain in the presence of noise.

The nonlinear method has no singularities and enforces all known constraints, but still does not have any geometric meaning [Hartley and Zisserman, 2004, p. 467] and frequently produces

unstable results in practice. We speculate from the recent trend towards more global approaches that minimize essentially the same cost function that the instability of the nonlinear method has been largely attributed to difficulties in obtaining a good initialization.

For example, in Hartley's stratified approach [Hartley et al., 1999], chirality constraints [Hartley, 1998b] are used to solve for a finite bounding volume for the plane at infinity and then this space is explored with a brute force search. From each candidate location, the infinite homography constraint is used to linearly estimate the ADC from any desired calibration constraints, the best plane is taken as the one that minimizes the least squares residual, and finally the result is improved nonlinearly. Unfortunately, this brute force search can be slow, and we have observed that the minimum is often so pointlike that the basin of attraction is not reliably found using any reasonably spaced discretization. Additionally, it has been pointed out [Nister, 2001b] that a single outlier can cause the chirality constraints to have no solution, or to not contain the correct solution.

More recently, the issue of discretization has been addressed by globally convergent methods. For example, interval analysis (IA) with branch and bound was used to minimize a heuristic based on the essential matrix constraint in Fusiello et al. [2004]. Unfortunately, the method was not very efficient, having computation times of about 1.5 hours for a problem with 40 views. IA was used again in Bocquillon et al. [2007], but the parameterization that was used only works for constant focal length and does not evenly distribute error. Computation times were improved in this latter method, but were still on the order of a minute for 20 views, which is too slow for many applications.

Under the constraint of zero skew (which can always be assumed in practice) and known principal point (which can be guessed but is often not known exactly), semidefinite programming was used to globally minimize a heuristic cost function in Agrawal [2004], which was extended with a brute force search for principal point in Agrawal [2007]. These methods enforced the internal ADQ constraints, but neglected the constraints on aspect ratio and always assumed that principal point is constant, which makes them applicable to video but not photo collections.

Convex relaxation was used with branch and bound to identify the plane at infinity that globally minimizes a heuristic cost associated with the modulus constraint in the recent stratified approach of Chandraker et al. [2007b, 2010], but the heuristic is not ideal because it does not consider constraints on the DIAC in the search for the plane at infinity. Similar techniques were used to estimate the ADQ directly using all known constraints in Chandraker et al. [2007a], which makes it perhaps the most generally applicable global approach.

In general, the globally convergent approaches are very difficult to implement and not very efficient. As an alternative, the dual stratified approach of estimating the plane at infinity from known calibration matrix, first proposed in Bougnoux [1998], has recently been revived with a closed form solution from a view pair in Gherardi and Fusiello [2010]. The advantage of the

dual stratified approach is that prior knowledge may be used to restrict the search into a very narrow plausible region, rather than exhaustively searching through all of parameter space for the plane at infinity. This leads to an algorithm that is simple, fast and robust. However, it lacks in precision, and still minimizes a heuristic objective that is not geometrically meaningful. As a result, attempting to further minimize the heuristic using nonlinear methods can result in divergence.

The fundamental limitation of all previous algorithms is that the objective being minimized is a heuristic with no particular geometric meaning, and these heuristics do not always work as well as one would hope. This becomes especially apparent for projective reconstructions with greater noise (or equivalently, more distant geometry) and for short reconstructions which are commonly on the verge of a Critical Motion Sequence (CMS) [Bocquillon et al., 2007] for which the problem is ill-posed.

Currently, the most meaningful heuristic objectives minimize a weighted sum of squared errors between the rectified intrinsic parameters and an assumed mean of zero. If calibration parameters are all independent and normally distributed, then the homography that minimizes this error is a maximum *a priori* rectifying homography [Nister, 2001b], where the weights (chosen heuristically) implicitly correspond to inverse variance of some assumed prior distribution. However, the tenability of this prior model has never been justified, and it has been primarily adopted out of convenience as a substitute for likelihood. In the next section, we show that in fact likelihood can be used as the objective, and we will present an algorithm to maximize likelihood robustly and efficiently.

6.2 Maximum Likelihood Autocalibration

Consider a set of n homogeneous structure points $\bar{\mathbf{X}}_i, i = 1 \dots n$ in the projective space $\mathbb{P}^3$, viewed by a set of m cameras having 3×4 projection matrices $\bar{\mathbf{P}}^j = \bar{\mathbf{K}}^j[\bar{\mathbf{R}}^j|\bar{\mathbf{t}}^j]$, $j = 1 \dots m$, where $\bar{\mathbf{R}}^j$ is a rotation matrix and $\bar{\mathbf{K}}^j$ is a non-singular upper triangular calibration matrix with positive diagonal elements. We refer to the combined set of this information as the *true configuration*, denoted by

$$\bar{\Theta} = \{\ \bar{\mathbf{X}}_i, \bar{\mathbf{P}}^j | \forall i, j\ \}, \tag{6.1}$$

and any estimate $\widehat{\Theta}$ of the configuration from some measurements as a *reconstruction* of the configuration.

The perspective projection of a homogeneous structure point $\mathbf{X} \in \mathbb{P}^3$ as viewed by a camera with projection matrix $\mathbf{P}$ is accomplished by multiplication, yielding a homogeneous image

point $\mathbf{x} \in \mathbb{P}^2$,

$$\mathbf{x} \propto \mathbf{P}\mathbf{X}. \tag{6.2}$$

Let the measured coordinates of the image of the ith structure point in the jth image be denoted by $\tilde{\mathbf{x}}_i^j$. If we assume, as is commonly done [Hartley and Sturm, 1997], that measurement error is normally distributed with standard deviation σ, then the probability (or likelihood) of a measurement is

$$P(\tilde{\mathbf{x}}_i^j|\bar{\Theta}) = \frac{1}{2\pi\sigma^2} \exp\left(-d(\tilde{\mathbf{x}}_i^j, \bar{\mathbf{x}}_i^j)^2/(2\sigma^2)\right), \tag{6.3}$$

where $\bar{\mathbf{x}}_i^j$ is the true image of $\bar{\mathbf{X}}_i$ in the jth view, and $d(\mathbf{a}, \mathbf{b})$ is the Euclidean distance between the inhomogeneous points represented by homogeneous points $\mathbf{a}$ and $\mathbf{b}$. The log-probability of a measurement is

$$\log P(\tilde{\mathbf{x}}_i^j|\bar{\Theta}) = -\frac{1}{2\sigma^2} d(\tilde{\mathbf{x}}_i^j, \bar{\mathbf{x}}_i^j)^2 + \underbrace{\log\left(1/(2\pi\sigma^2)\right)}_{constant}, \tag{6.4}$$

and therefore the maximum likelihood (ML) projective reconstruction $\widehat{\Theta}_{ML}$ from measurements $\{\tilde{\mathbf{x}}_i^j\}$ is given by

$$\widehat{\Theta}_{ML} = \underset{\Theta}{\operatorname{argmax}} \prod_{i,j} P(\tilde{\mathbf{x}}_i^j|\Theta) \tag{6.5}$$

$$= \underset{\Theta}{\operatorname{argmax}} -\frac{1}{2\sigma^2} \sum_{i,j} d(\tilde{\mathbf{x}}_i^j, \mathbf{x}_i^j)^2 \tag{6.6}$$

$$= \underset{\Theta}{\operatorname{argmin}} \sum_{i,j} d(\tilde{\mathbf{x}}_i^j, \mathbf{x}_i^j)^2. \tag{6.7}$$

The distance $d(\tilde{\mathbf{x}}_i^j, \mathbf{x}_i^j)$ is known as *reprojection error*, so the ML reconstruction is the one that minimizes the sum of squared reprojection errors. This nonlinear minimization is known as *bundle adjustment* [Triggs et al., 2000; Hartley and Zisserman, 2004; Lourakis and Argyros, 2004]. Because multiplying a projective reconstruction by a homography does not change reprojection error, the result of projective bundle adjustment is ambiguous up to a homography.

Autocalibration is an attempt to resolve this ambiguity. Previous algorithms have done so by assuming that certain intrinsic parameters (discussed in Section 6.2.1) should be distributed according to a known prior model in the metric frame, and have therefore sought the rectifying homography that maximizes prior probability according to the prior model. We refer the interested reader to 6.3, where we explain the fundamental instability and limitations of this

heuristic maximum *a priori* approach.

However, as discussed in Section 6.2.1, a reconstruction that does not satisfy metric constraints is not really a plausible reconstruction, and if one projects the rectified solution into the space of valid metric reconstructions by imposing metric constraints then reprojection errors will be increased. Thus, it is possible to seek the maximum likelihood rectifying homography by minimizing these reprojection errors. Specifically, the maximum likelihood rectifying homography is given by

$$\widehat{\mathbf{H}}_{ML} = \underset{\mathbf{H}}{\operatorname{argmin}} \sum_{i,j} d(\tilde{\mathbf{x}}_i^j, \mathbf{P}_{\mathsf{C}}^j \mathbf{H}^{-1} \mathbf{X}_i)^2, \tag{6.8}$$

where $\{\mathbf{P}_{\mathsf{C}}^j | \forall j\}$ are the closest projection matrices to $\{\mathbf{P}^j\mathbf{H} | \forall j\}$ that exactly satisfy the metric constraints. They can be found by decomposing each $\mathbf{PH} \Rightarrow \mathbf{K}[\mathbf{R}|\mathbf{t}]$, forming the augmented calibration matrix $\mathbf{K}'$ by enforcing metric constraints on $\mathbf{K}$, and then reforming $\mathbf{P}_{\mathsf{C}} \Leftarrow \mathbf{K}'[\mathbf{R}|\mathbf{t}]$.

6.2.1 Metric Constraints

Any 3×4 projection matrix $\mathbf{P}$ can be uniquely factored [Golub and Van Loan, 1996b, p.230] into a rotation matrix $\mathbf{R}$, translation $\mathbf{t}$, and right triangular calibration matrix $\mathbf{K}$ as

$$\mathbf{P} \propto \mathbf{K}[\mathbf{R}|\mathbf{t}]. \tag{6.9}$$

Following Hartley and Zisserman [2004], we denote the elements of the calibration matrix by

$$\mathbf{K} = \begin{bmatrix} \alpha_x & s & u \\ & \alpha_y & v \\ & & 1 \end{bmatrix}, \tag{6.10}$$

where $s \in (-\infty, \infty)$ is the skew parameter, α_x and α_y are the horizontal and vertical image scaling factors (α_x is focal length), and (u, v) are the coordinates of the principal point. These are the intrinsic camera parameters.

Because the images taken by any real camera will be unskewed and have a known (usually 1:1) pixel aspect ratio, this means that in the true configuration $s = 0$ and $\alpha_x = \alpha_y$. Furthermore, when it is known that the principal point is in the center of the image, the image coordinate system can be chosen so that $u = v = 0$. Additionally, for any two projection

matrices that are known to correspond to the same camera, the calibration matrices will be equal.

The restriction that cameras can only image what is in front of them results in an inequality constraint for each measured image point. These inequality constraints, known as *chirality constraints* [Hartley, 1998b], can be used to restrict the location of the plane at infinity π_∞ (which partially defines $\mathbf{H}$) to a convex polytope [Hartley et al., 1999], but do not actually assist in pinpointing an exact solution.

With the exception of chirality constraints, all previous constraints that have been used for autocalibration can be derived from the above constraints on $\mathbf{K}$ (see Section 6.2.2). This includes the rank (e.g., common plane at infinity π_∞) and positive-semidefinite constraints of the absolute dual quadric $\mathbf{Q}^*_\infty$ Triggs [1997], the linear and nonlinear constraints on $\mathbf{Q}^*_\infty$ [Pollefeys et al., 1998], the infinite homography constraint on the absolute dual conic Ω^*_∞ [Hartley and Zisserman, 2004], the Kruppa equations [Kruppa, 1913], the essential matrix and rigidity constraints [Huang and Faugeras, 1989], and the modulus constraint Pollefeys and Van Gool [1999]. In other words, although (6.8) is not specifically formulated in terms of these previously used constraints, it does not neglect any of them.

We propose an additional inequality constraint based on the fact that a camera field of view must be in the range $(0, \pi)$, and practically speaking a much less conservative range can usually be assumed which we denote $(\theta_{min}, \theta_{max})$. Thus, focal length must be in the range

$$f \in \left(\frac{S_w}{2\tan(\frac{\theta_{max}}{2})}, \frac{S_w}{2\tan(\frac{\theta_{min}}{2})} \right). \tag{6.11}$$

For a standard 35mm camera using an 18-55mm lens, this means that focal length is approximately in the range of 1-3 screen widths (in pixels).

6.2.2 Relationship to Previous Autocalibration Constraints

Recall that any 3×4 projection matrix $\mathbf{P}$ can be uniquely factored into a rotation matrix $\mathbf{R}$, translation $\mathbf{t}$, and calibration matrix $\mathbf{K}$ as

$$\mathbf{P} \propto \mathbf{K}[\mathbf{R}|\mathbf{t}]. \tag{6.12}$$

Define $\tilde{\mathtt{I}} = \mathrm{diag}(1, 1, 1, 0)$. Then, observe that the rotation and translation components can

be cancelled out by

$$\begin{aligned}\mathbf{P}\tilde{\mathtt{I}}\mathbf{P}^{\mathsf{T}} &\propto \mathbf{K}[\mathbf{R}|\mathbf{t}]\tilde{\mathtt{I}}[\mathbf{R}|\mathbf{t}]^{\mathsf{T}}\mathbf{K}^{\mathsf{T}} \\ &\propto \mathbf{K}[\mathbf{R}|\mathbf{0}]\begin{bmatrix}\mathbf{R}^{\mathsf{T}} \\ \mathbf{0}\end{bmatrix}\mathbf{K}^{\mathsf{T}} \\ &\propto \mathbf{K}\mathbf{K}^{\mathsf{T}}.\end{aligned} \tag{6.13}$$

Any projective camera $\mathbf{P}^j$ in the reconstruction is related to a metric camera (denoted by subscript $\mathtt{M}$) via the rectifying homography $\mathbf{H}_{\mathtt{M}}$,

$$\mathbf{P}^j_{\mathtt{M}} \propto \mathbf{P}^j\mathbf{H}_{\mathtt{M}} \quad \forall j. \tag{6.14}$$

Substituting (6.14) into (6.13), we obtain

$$\mathbf{P}^j\mathbf{H}_{\mathtt{M}}\tilde{\mathtt{I}}\mathbf{H}_{\mathtt{M}}^{\mathsf{T}}{\mathbf{P}^j}^{\mathsf{T}} \propto \mathbf{K}^j{\mathbf{K}^j}^{\mathsf{T}} \quad \forall j. \tag{6.15}$$

Using (6.15), prior constraints on the calibration matrices can be translated into nonlinear constraints on $\mathbf{H}_{\mathtt{M}}$. This makes it the most fundamental equation of autocalibration, from which all other constraints that are based on calibration matrices can be derived. Making the substitutions of $\mathbf{Q}^*_\infty = \mathbf{H}_{\mathtt{M}}\tilde{\mathtt{I}}\mathbf{H}_{\mathtt{M}}^{\mathsf{T}}$ and $\omega^{*j} = \mathbf{K}^j{\mathbf{K}^j}^{\mathsf{T}}$, equation (6.15) is usually written as a constraint on $\mathbf{Q}^*_\infty$ and ω^{*j},

$$\mathbf{P}^j\mathbf{Q}^*_\infty{\mathbf{P}^j}^{\mathsf{T}} \propto \omega^{*j} \quad \forall j. \tag{6.16}$$

In the literature $\mathbf{Q}^*_\infty$ is known as the *absolute dual quadric* (ADQ), and ω^{*j} is a *dual image of the absolute conic* (DIAC). The relationship between these entities is depicted graphically in Fig. 6.1.

The advantage of using $\mathbf{Q}^*_\infty$ is that some constraints can be translated into linear constraints on $\mathbf{Q}^*_\infty$, allowing linear least squares method to be used as an initialization. However, there are additional internal constraints on $\mathbf{Q}^*_\infty$ that cannot be enforced by a linear solution. By construction, $\mathbf{Q}^*_\infty$ must be symmetric, rank 3 and positive-semidefinite (Theorem 2) and all ω^{*j} must be symmetric and positive-semidefinite (Theorem 3).

Theorem 2. *Let* $\mathbf{H}$ *be any real matrix. Then* $\mathbf{H}\mathbf{H}^{\mathsf{T}}$ *is positive-semidefinite.*

Proof. A square matrix $\mathbf{M}$ is positive-semidefinite if and only if $\mathbf{z}^\mathsf{T}\mathbf{M}\mathbf{z} \geq 0$, for any non-zero $\mathbf{z}$. It holds that $\mathbf{z}^\mathsf{T}\mathbf{H}\mathbf{H}^\mathsf{T}\mathbf{z} = ||\mathbf{H}^\mathsf{T}\mathbf{z}||^2 \geq 0$. □

Theorem 3. *Let* $\mathbf{Q}$ *be an* $n \times n$ *positive-semidefinite matrix, and* $\mathbf{P}$ *be any* $m \times n$ *matrix. Then* $\mathbf{P}\mathbf{Q}\mathbf{P}^\mathsf{T}$ *is also positive-semidefinite.*

Proof. The factorization $\mathbf{Q} = \mathbf{L}\mathbf{L}^\mathsf{T}$ must exist because $\mathbf{Q}$ is positive-semidefinite. Let $\mathbf{H} = \mathbf{P}\mathbf{L}$. Then $\mathbf{H}\mathbf{H}^\mathsf{T} = \mathbf{P}\mathbf{L}\mathbf{L}^\mathsf{T}\mathbf{P}^\mathsf{T} = \mathbf{P}\mathbf{Q}\mathbf{P}^\mathsf{T}$ is positive-semidefinite by Theorem 2. □

Because $\mathbf{Q}^*_\infty$ is singular, it is a degenerate (dual) quadric, meaning that it actually represents a dual conic embedded in some plane. This dual conic, called the *absolute dual conic* (ADC) and denoted by Ω^*_∞, is encoded in the upper 3×3 portion of $\mathbf{Q}^*_\infty$. The plane it lives in is called the *plane at infinity*, denoted by π_∞ and encoded by the null space of $\mathbf{Q}^*_\infty$. Geometrically, (6.16) shows us that ω^{*j} is the projection of $\mathbf{Q}^*_\infty$ onto the image plane of $\mathbf{P}^j$. Therefore, Ω^*_∞ is also the projection of $\mathbf{Q}^*_\infty$ by the canonical projection matrix, $\mathbf{P} = [\mathtt{I}|0]$.

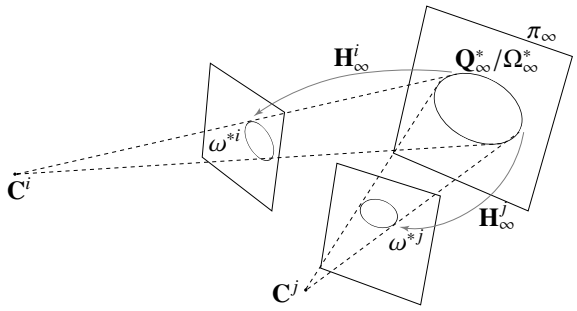


Figure 6.1. The absolute dual quadric $\mathbf{Q}^*_\infty$ encodes for the absolute dual conic Ω^*_∞, as well as the plane at infinity π_∞ that Ω^*_∞ is embedded in. A dual image of the absolute conic ω^{*j} is the image of Ω^*_∞ as seen by camera $\mathbf{P}^j$ having focal point at $\mathbf{C}^j$, and can be obtained either by projection of $\mathbf{Q}^*_\infty$, or by using the infinite homography $\mathbf{H}^j_\infty$ to map Ω^*_∞ from π_∞ to the image plane of $\mathbf{P}^j$.

There is some notational inconsistency in the literature, with ω^* sometimes being used to represent Ω^*_∞, and Ω^*_∞ sometimes being used to represent $\mathbf{Q}^*_\infty$. We do not prefer this alternative notation because, under the assumption of constant calibration matrices where $\omega^{*j} = \omega^{*i}\ \forall i, j$, it can lead to confusion between ω^* and ω^{*j}, which are equivalent only if $\mathbf{P}^j = [\mathtt{I}|\mathbf{0}]$.

In a metric reconstruction $\mathbf{H}_\mathtt{M} = \mathtt{I}$, in which case one observes that $\mathbf{Q}^*_{\infty\mathtt{M}} = \tilde{\mathtt{I}}$, $\Omega^*_{\infty\mathtt{M}} = \mathtt{I}$, and $\pi_{\infty\mathtt{M}} = (0, 0, 0, 1)^\mathsf{T}$, which does not correspond to any real plane equation. By definition, a point $\mathbf{X}$ lies on a plane π iff $\pi^\mathsf{T}\mathbf{X} = 0$, and therefore the only points lying on $\pi_{\infty\mathtt{M}}$ are homogeneous points of the form $(a, b, c, 0)^\mathsf{T}$. These are all points at infinity, hence why we call π_∞ the plane at infinity. Under the action of a homography $\mathbf{H}$ a plane π transforms to $\pi' = \mathbf{H}^{-\mathsf{T}}\pi$, and hence π_∞ could be any real plane in the projective reconstruction.

6.2.2.1 The Infinite Homography

As a consequence of the fact that $\mathbf{Q}_\infty^*$ is degenerate, its projection ω^{*j} can alternatively be computed via a planar homography transfer of Ω_∞^*. The homography that transfers from $\pi_\infty = (\mathbf{p}^\mathsf{T}, 1)^\mathsf{T}$ to the image plane of view $\mathbf{P}^j = [\mathbf{A}^j|\mathbf{a}^j]$ is called the *infinite homography*, denoted by $\mathbf{H}_\infty^j$, and given by

$$\mathbf{H}_\infty^j \propto \mathbf{A}^j - \mathbf{a}^j\mathbf{p}^\mathsf{T} \quad \forall j. \tag{6.17}$$

Under the action of a homography $\mathbf{H}$, a dual conic ω^* transforms to $\omega^{*\prime} = \mathbf{H}\omega^*\mathbf{H}^\mathsf{T}$. Thus, we obtain the *infinite homography constraint* on Ω_∞^*,

$$\mathbf{H}_\infty^j \Omega_\infty^* {\mathbf{H}_\infty^j}^\mathsf{T} \propto \omega^{*j} \quad \forall j. \tag{6.18}$$

The infinite homography constraint is a pairwise constraint between image planes (see Fig. 6.1), a special case of (6.15) that does not enforce the common support plane for π_∞.

6.2.2.2 The Kruppa Equations

Let $\mathbf{e}^{ij}$ denote the image of the ith camera center in view j (an epipole). There is a corresponding skew-symmetric matrix $[\mathbf{e}^{ij}]_\times$ [Hartley and Zisserman, 2004, p. 581], and multiplying both sides of (6.18) by $[\mathbf{e}^{ij}]_\times$ leads to

$$[\mathbf{e}^{ij}]_\times \omega^{*j} [\mathbf{e}^{ij}]_\times \propto \left([\mathbf{e}^{ij}]_\times \mathbf{H}_\infty^j\right) \Omega_\infty^* \left({\mathbf{H}_\infty^j}^\mathsf{T} [\mathbf{e}^{ij}]_\times\right) \tag{6.19}$$

$$\propto \mathbf{F}_{ij} \Omega_\infty^* \mathbf{F}_{ij}^\mathsf{T} \quad \forall ij, \tag{6.20}$$

where $\mathbf{F}_{ij}$ is the fundamental matrix between views i and j. The set of equations in (6.20) are equivalent to the original Kruppa equations [Kruppa, 1913], expressing constraints on Ω_∞^* in the form of corresponding epipolar lines tangent to ω^{*j}. Thus, the Kruppa equations are just a special case of the infinite homography constraint.

6.2.2.3 The Rigidity Constraint

It has been shown [Huang and Faugeras, 1989] that the Kruppa equations are also equivalent to a constraint that the essential matrix has two identical singular values and one zero singular value. The essential matrix is related to the fundamental matrix by

$$\mathbf{E}_{ij} = [\mathbf{t}]_\times \mathbf{R} = {\mathbf{K}^j}^\mathsf{T} \mathbf{F}_{ij} \mathbf{K}^i, \tag{6.21}$$

where $\mathbf{t}$ and $\mathbf{R}$ represent the translation and rotation between the camera pair. This constraint may also be stated as

$$\det(\mathbf{E}_{ij}) = 0 \ \wedge \ 2\,\mathrm{tr}((\mathbf{E}_{ij}\mathbf{E}_{ij}^\mathsf{T})^2) - (\mathrm{tr}(\mathbf{E}_{ij}\mathbf{E}_{ij}^\mathsf{T}))^2 = 0, \tag{6.22}$$

and is called the *rigidity constraint* on $\mathbf{E}_{ij}$ because it is a result of the rigid motion between cameras.

6.2.2.4 The Modulus Constraint

Without loss of generality, we can align the projective reconstruction such that $\mathbf{P}^i = [\mathtt{I}|\mathbf{0}]$, and choose our metric reconstruction such that $\mathbf{P}^i_\mathtt{M} = \mathbf{K}^i[\mathtt{I}|0]$. Then, because $\mathbf{P}^i_\mathtt{M} = \mathbf{P}^i\mathbf{H}_\mathtt{M}$, it can be verified that $\mathbf{H}_\mathtt{M}$ is of the form

$$\mathbf{H}_\mathtt{M} = \begin{bmatrix} \mathbf{K}^i & 0 \\ \mathbf{v}^\mathsf{T} & 1 \end{bmatrix}. \tag{6.23}$$

In this case, $\Omega^*_\infty = \mathbf{K}^i{\mathbf{K}^i}^\mathsf{T}$, and $\mathbf{v} = -{\mathbf{K}^i}^\mathsf{T}\mathbf{p}$. Substituting (6.23) into (6.14), we see that

$$\mathbf{K}^j[\mathbf{R}^j|\mathbf{t}^j] \propto [(\mathbf{A}^j - \mathbf{a}^j\mathbf{p}^\mathsf{T})\mathbf{K}^i|\mathbf{a}^j] \tag{6.24}$$

$$\mathbf{K}^j\mathbf{R}^j \propto (\mathbf{A}^j - \mathbf{a}^j\mathbf{p}^\mathsf{T})\mathbf{K}^i \tag{6.25}$$

$$\mathbf{K}^j\mathbf{R}^j{\mathbf{K}^i}^{-1} \propto \mathbf{A}^j - \mathbf{a}^j\mathbf{p}^\mathsf{T}. \tag{6.26}$$

Thus, if $\mathbf{K}^j = \mathbf{K}^i$ then $\mathbf{H}^j_\infty = \mathbf{A}^j - \mathbf{a}^j\mathbf{p}^\mathsf{T}$ is similar (a.k.a. conjugate) to a rotation, and has eigenvalues proportional to $\{e^{i\theta}, e^{-i\theta}, 1\}$. In other words, the modulus of the first two eigenvalues are equal. This is known as the *modulus constraint* on $\mathbf{p}$. Note that the modulus constraint is usually enforced as a constraint on the coefficients of the characteristic polynomial of $\mathbf{H}^j_\infty$, as in Pollefeys and Van Gool [1999]; Chandraker et al. [2010].

Assuming all calibration matrices are equal, enforcing the modulus constraint between all pairs of views would ensure a common support plane for π_∞, and is therefore also a special case of (6.16).

6.3 Limitations of Maximum *a Priori* Autocalibration

We denote the metric-rectified calibration vector for the jth view by $\mathbf{k}^j = (\alpha_x^j - \alpha_y^j, s^j, u^j, v^j)^\mathsf{T}$. If one *assumes* that these calibration vectors are each distributed according to independent multivariate normal distributions, then the maximum *a priori* rectifying homography would be given by minimizing a sum of squared Mahalanobis distances,

$$\widehat{\mathbf{H}}_{MAP} = \underset{\mathbf{H}}{\operatorname{argmax}} \prod_{j=1}^{m} \frac{1}{\sqrt{(2\pi)^N |\Sigma_\mathbf{k}|}} \exp\left(-\frac{1}{2}(\mathbf{k}^j - \mu_\mathbf{k})^\mathsf{T} \Sigma_\mathbf{k}^{-1} (\mathbf{k}^j - \mu_\mathbf{k})\right) \tag{6.27}$$

$$= \underset{\mathbf{H}}{\operatorname{argmin}} \sum_{j=1}^{m} (\mathbf{k}^j - \mu_\mathbf{k})^\mathsf{T} \Sigma_\mathbf{k}^{-1} (\mathbf{k}^j - \mu_\mathbf{k}), \tag{6.28}$$

where $N = 4$ is the length of $\mathbf{k}^j$, and $\mu_\mathbf{k} = (0, 0, 0, 0)^\mathsf{T}$ and $\Sigma_\mathbf{k}$ are mean and covariance that define the prior distribution. If one further *assumes* that all parameters are independent, then (6.28) reduces to a weighted sum of squared errors,

$$\widehat{\mathbf{H}}_{MAP} = \underset{\mathbf{H}}{\operatorname{argmin}} \sum_{j} w_u (u^j)^2 + w_v (v^j)^2 + w_\alpha (\alpha_x^j - \alpha_y^j)^2 + w_s (s^j)^2. \tag{6.29}$$

Most autocalibration algorithms strive to optimize an objective of this form (or some close approximation), where the weighting coefficients w_u, w_v, w_α, w_s are determined using various guesses [Pollefeys et al., 2002b; Bocquillon et al., 2007; Gherardi and Fusiello, 2010] or more commonly just omitted [Pollefeys et al., 1998; Hartley et al., 1999; Hartley and Zisserman, 2004; Bocquillon et al., 2007], which is equivalent to assuming they are all equal. It has been thought that the ideal way to choose these weighting coefficients is by looking at the distribution of intrinsic parameters in real cameras [Nister, 2001b; Pollefeys et al., 2002b].

However, with the exception of focal length, the uncertainty of these parameters in real cameras is negligible because modern cameras are manufactured to not produce skewed or stretched images. In other words, the prior model for these parameters should essentially be a delta function, and the parameters of any rectified result will never fall within the high density region of such a distribution. It is well known that squared error is sensitive to outliers, and this explains the fundamental instability of methods that use (6.29) as an objective.

The dominant source of uncertainty in the rectified calibration vectors is not due to uncertainty in the prior model, but rather is propagated from uncertainty in the measured image correspondences. Because projection matrices are over-parameterized (a homogeneous projection matrix has 11 dof whereas a metric camera has only 7 dof for pose and focal length), metric

constraints will be happily violated by projective bundle adjustment in order to achieve a solution with lower reprojection error. The rectifying homography does not have enough dof to restore these metric constraints, and hence the error is propagated into the intrinsic parameters of the metric rectified result.

Thus, the distribution of calibration parameters is configuration dependent, and the assumptions of (6.29) that parameters are normally distributed and independent are not valid. If one were to use this objective anyway, the optimal choice of weighting coefficients w_u, w_v, w_α, w_s should ideally be determined as the inverse of propagated variance, rather than being hard-coded based on some prior model.

Because variance must be propagated through projective reconstruction, then through autocalibration, and finally through the RQ factorization to get calibration parameters, we have not had much success using analytical approximations. However, we have managed to obtain accurate estimations of the propagated variance by using Monte Carlo methods. Each trial consists of perturbing the original image point measurements with Gaussian noise and then repeating the entire process of projective reconstruction and autocalibration. An example of the sample covariance matrix containing all intrinsic parameters for a system of 6 views is shown in Fig. 6.2, calculated from 1000 random trials.

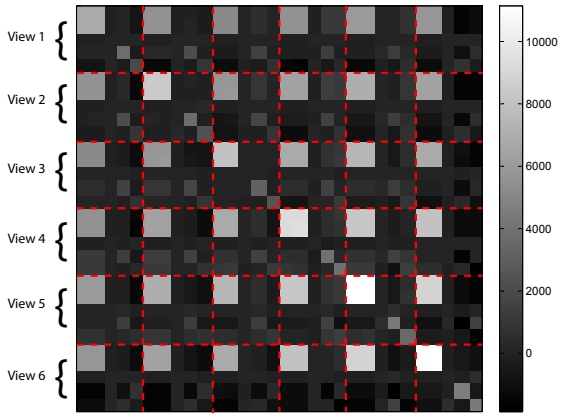


Figure 6.2. Covariance matrix of intrinsic parameters for a set of six views. The parameters are ordered as $(\alpha_x^1, \alpha_y^1, s^1, u^1, v^1, \ldots, \alpha_x^6, \alpha_y^6, s^6, u^6, v^6)$. The delineation between parameters of the same view is indicated by dotted grid lines.

We see from Fig. 6.2 that the greatest amount of uncertainty is propagated into focal length, and secondly into principal point. This confirms the already known facts that neither the estimation of principle point nor focal length are well-conditioned problems [Bougnoux, 1998]. The large covariance between horizontal and vertical focal lengths shows that the aspect

ratio constraint is relatively good. In some views, there is a noticeable positive or negative correlation between focal length and principal point. Additionally, we see that there is very strong dependence between the focal lengths of different views. In other words, the assumption of independence in (6.29) is incorrect.

In Fig. 6.3, we take a closer look at the covariance of constraints typically used in auto-calibration. We use both Hartleys and Faugeras parameterization. Of these quantities, we see that principal point has the greatest variation, and becomes increasingly uncertain for cameras farther from the origin.

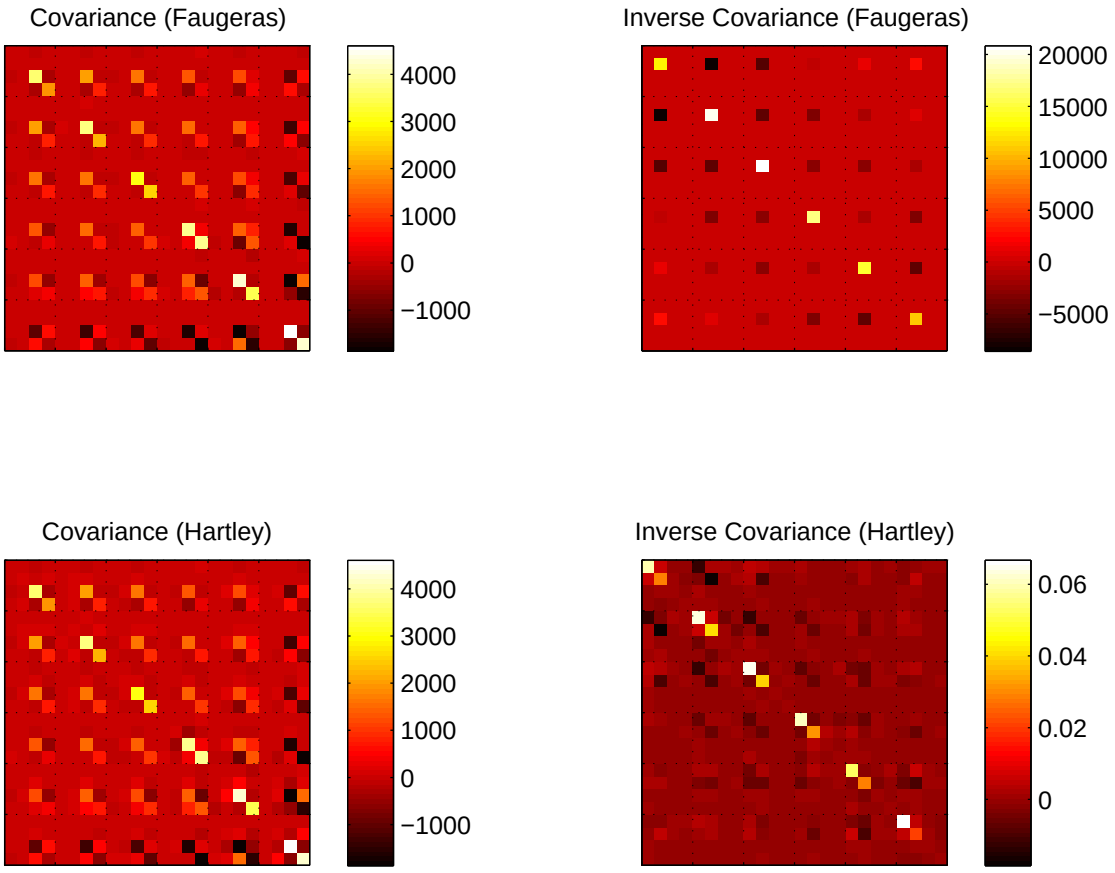


Figure 6.3. Covariance matrix of calibration vector for all six cameras (camera boundaries indicated by grid lines). In Hartley's parameterization, the calibration vector is $(\alpha_x - \alpha_y, s, u, v)$. In Faugeras' parameterization, the calibration vector is $(\alpha_u - \alpha_v, \theta, u, v)$. Each. The sample covariance is based on monte carlo propagation of covariance with 1000 perturbations of the correspondence measurements with $\sigma = 1$ pixel.

By looking at the inverse of these matrices, we see how to optimally weight the constraints in a linear least squares problem (i.e., so as to minimize Mahalanobis distance or maximize

the likelihood). In Hartley's parameterization, skew and aspect ratio are both highly weighted, whereas constraints on principal point are mostly useless.

The inverse variance for aspect ratio is consistently about 2-3 times higher than for the skew. We notice significant negative correlation between the skew parameters of spatially similar cameras. However, it is better to minimize the skew constraint on skew angle rather than skew parameter because minimizing the skew parameter would create a bias towards smaller focal lengths, and is also subject to greater variability because it is multiplied by focal length. We notice that the variance in the skew parameter is about 6 orders of magnitude greater than the variance in skew angle.

Interestingly, we see a strong negative correlation between the skew angle of spatially similar cameras. Additionally, there is less variance in skew for cameras that are in the middle of the chain. This is likely due to the spatial correlations between skew, which means that the more nearby cameras there are, the more restricted the skew angle becomes and the more powerful the skew constraint becomes for those cameras.

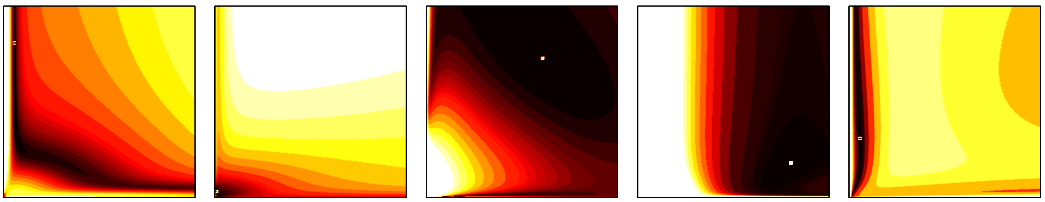
In order to get a more qualitative idea about how to best choose weighting coefficients, we calculated the constraint power (inverse variance) from 100 random configurations and show the results relative to skew, with 95% confidence intervals, in Table 6.1. These results indicate that in general, skew and aspect ratio constraints should be weighted approximately equally, whereas principal point should have a very negligible weight. In practice, we find that using any non-zero weight on principal point or focal length tends to magnify the effects of noise and provide a worse result.

Table 6.1. Power of calibration constraints relative to a constraint on skew. Calculated from 100 random configurations and shown with 95% confidence intervals.

Constraint	**Relative Power**
s	1
$\alpha_x - \alpha_y$	1.01577 ± 0.0170464
α_x	0.0169471 ± 0.00156958
α_y	0.0169465 ± 0.00157017
u	0.0217334 ± 0.00202811
v	0.0219337 ± 0.00200886

However, we stress that the optimal choice of weighting coefficients in (6.29) is still highly configuration dependent. In order to demonstrate this we have generated several random configurations and then autocalibrated using all possible relative weightings between skew and aspect

ratio. The objective measure of autocalibration accuracy is shown as a 2D surface for each configuration in Fig. 6.4. One sees that the location of the minima is different for each problem, which is a clear indication that the optimal choice of weights is configuration dependent.



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

Figure 6.4. Example cost surfaces demonstrating that the optimal choice of weights is configuration dependent. Each surface corresponds to a different configuration, and the intensity at each point on the surface indicates the objective reconstruction quality as a function of the relative weighting between skew (x-axis) and aspect ratio (y-axis) constraints. The weights corresponding to the most accurate reconstruction is marked, and change significantly with each configuration.

This configuration dependency means that one can only expect a marginal statistical advantage by using optimal weighting coefficients. To demonstrate this, we created a suite of 100 random configurations and then tried autocalibrating them while varying the weight on aspect ratio relative to skew angle. The results are shown in Fig. 6.5 and Fig. 6.6, and confirm our assumption that the best quality is obtained when using the weights determined by constraint power, as well as the large degree of variability.

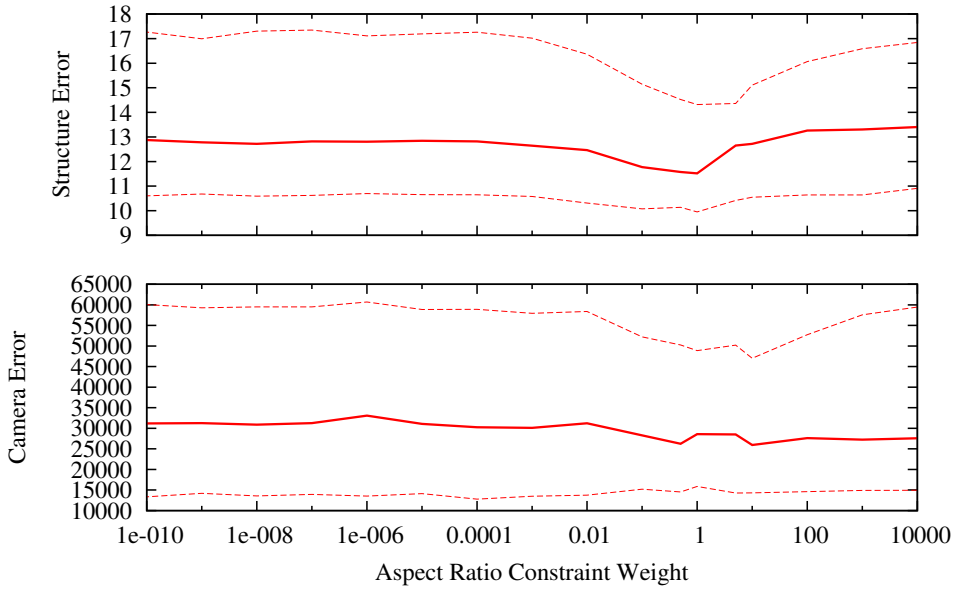


Figure 6.5. Objective reconstruction quality after autocalibration with varying weight on aspect ratio constraint using Hartley's parameterization, relative to a weight of 1 on the skew parameter constraint. We have plotted the median, first and third quartiles over 100 configurations.

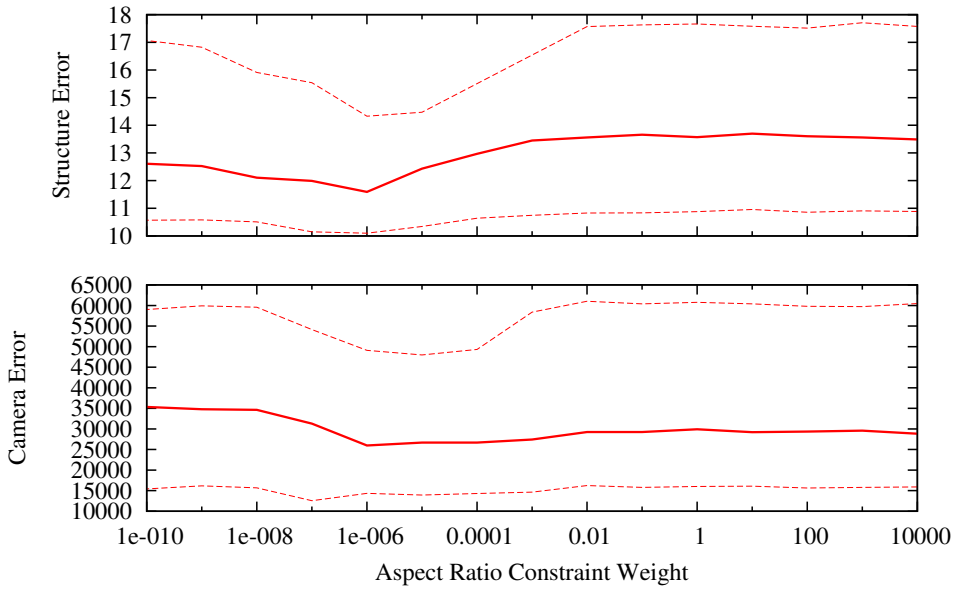


Figure 6.6. Objective reconstruction quality after autocalibration with varying weight on aspect ratio constraint using Faugeras' parameterization, relative to a weight of 1 on the skew angle constraint. We have plotted the median, first and third quartiles over 100 configurations.

Surprisingly, there does not seem to be any difference in the results when using Hartley or Faugeras' parameterization with the best weights. This is further confirmed by a plot of the empirical cumulative distribution of the objective measure, shown in Fig. 6.7.

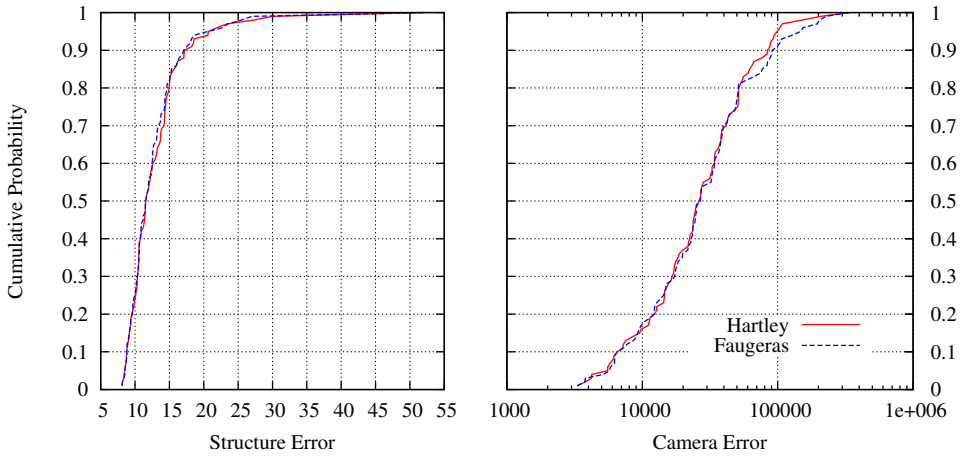


Figure 6.7. Empirical cumulative distribution of structural and camera reconstruction quality when using Hartley's vs Faugeras' parameterization with optimal weighting coefficients for skew and aspect ratio.

6.4 Implementation

The potential constraints that could be incorporated into (6.8) depend upon the type of problem (if it is a photo collection, video, etc). The most general constraints, which are applicable to all practical autocalibration problems, are that skew is zero and pixel aspect ratio is unity. On some problems, one may also assume that the principal point is constant or zero, or that focal length is constant.

However, we have found experimentally that even in synthetic problems where these latter constraints are known to be satisfied, including them at this stage actually worsens the results (see Section 6.3). This is because even a small amount of noise in the image points will cause the projective reconstruction to find a solution that cannot be autocalibrated without introducing a large amount of error into principal point and focal length. Therefore, the only calibration matrix constraints we incorporate into (6.8) are skew, aspect ratio, and the focal length inequality constraints. Note that we still make use of the remaining constraints later, as will be discussed in Section 6.4.2.

As previously mentioned, Nister [2001b] pointed out that a single outlier can cause the chirality constraints to have no solution, or for the solution polytope to not contain the true plane at infinity. Therefore, in order to remain robust to these potential circumstances, we do

not enforce chirality constraints as hard-constraints but rather add a large penalty (1000) to the error for each point that would violate chirality constraints.

From an initial estimate, it is straightforward to minimize (6.8) using Levenberg-Marquardt (LM) [Marquardt, 1963] with numerical differentiation. The only remaining challenge is in finding a good initial estimate of the rectifying homography.

6.4.1 Initialization

We start off by linearly estimating $\mathbf{Q}_\infty^*$ because it is efficient and precise in the absence of noise. Then we adapt the theory of the dual stratified approach (DS) of Gherardi and Fusiello [2010] to directly minimize (6.8), with some modifications to improve generality and performance. The advantage of the dual stratified approach is that it is simple to implement and delivers both efficiency and robustness by using importance sampling to guide the search. Moreover, it gives us the freedom to minimize the same objective function during the initialization that we later optimize nonlinearly.

Specifically, we use a Monte Carlo (MC) search and sample the calibration matrices for a random view pair from an assumed prior distribution. Thus, the inequality constraints on focal length can be imposed simply by not sampling outside of this range. We assume that both views in the pair have the same calibration matrices because it reduces the size of the search space from S^2 to S, and this generally improves performance and robustness. There is only a negligible loss of generality because even when calibration matrices are not all equal, in practice there is still enough inherent similarity that one can always assume a subset of two of them will be sufficiently similar in this initial step, and the calibration matrices can still be treated independently in all subsequent steps.

Next we calculate the rectifying homography based on the two views and keep track of the homography that minimizes reprojection error. We call this dual stratified Monte Carlo maximum likelihood (DS-MC-ML) initialization, and give pseudo-code in Algorithm 2. The 'Evaluate' function refers to (6.8) and 'AutocalibratePair' refers to the closed form solution presented in Gherardi and Fusiello [2010].

Algorithm 2 DS-MC-ML Initialization

Require: A projective *reconstruction* from ≥ 2 views, and a prior model for the distribution of $\mathbf{K}_i$, $i = 1 \ldots m$.

Ensure: $\widehat{\mathbf{H}}$ should be in the basin of attraction of $\widehat{\mathbf{H}}_{ML}$.

$\widehat{\mathbf{H}} \leftarrow$ LinearAutocalibrate(*reconstruction*)
$\epsilon_{min} \leftarrow$ Evaluate($\widehat{\mathbf{H}}$, *reconstruction*)
repeat
  $\{\mathbf{P}_1, \mathbf{P}_2\} \leftarrow$ select two projection matrices at random.
  $\mathbf{K} \leftarrow$ sample from prior distribution.
  $\mathbf{H} \leftarrow$ AutocalibratePair($\mathbf{P}_1, \mathbf{K}, \mathbf{P}_2, \mathbf{K}$)
  $\epsilon \leftarrow$ Evaluate($\mathbf{H}$, *reconstruction*)
  if $\epsilon < \epsilon_{min}$ **then**
    $\epsilon_{min} \leftarrow \epsilon$
    $\widehat{\mathbf{H}} \leftarrow \mathbf{H}$
    $count \leftarrow 0$
  else
    $count \leftarrow count + 1$
  end if
until $count \geq stopCount$

The number of iterations that are needed in order to find a solution that is in the basin of attraction of the global optimum is dependent on the structure of the epipolar geometry, the number of cameras, and the assumed prior distribution. If the epipolar geometry is near a Critical Motion Sequence (CMS) [Bocquillon et al., 2007], or if there is a large number of views, or if there is a large uncertainty in the assumed prior distribution then more samples will be needed.

Therefore, rather than choosing a fixed number of iterations up front, we prefer to use an adaptive strategy that terminates the sampling process after *stopCount* iterations have elapsed without further improvement. This allows the algorithm to remain efficient for simple problems, while naturally scaling up to use more iterations on more difficult problems.

Because we measure reprojection error, it would also be easy to pick a reasonable threshold for early termination based on the error value. This is something that is not possible to do using previous error heuristics where there is no clear relationship between the heuristic and a meaningful quantification of the actual amount of error.

The beauty of DS-MC-ML initialization is that the search space is inherently restricted to solutions that rectify with at least two cameras having calibration matrices well within the

high density region of the assumed probability distribution of calibration matrices. This is a guarantee that even the globally optimal approaches cannot make.

6.4.2 Resectioning to Enforce Constraints

Because metric constraints are not enforced during the initial projective reconstruction phase, there are essentially too many degrees of freedom in the initial solution, and hence multiplication by the rectifying homography will never cause the constraints to be satisfied exactly.

In order to obtain a rectified solution that exactly satisfies all metric constraints, we follow up metric rectification by resectioning (a.k.a. re-estimating from structure points) all of the camera matrices using the rectified structure points. This is done by choosing a metric parameterization for camera matrices that implicitly enforces the metric constraints and then minimizing reprojection errors using LM.

We use quaternion rotations for camera pose and assume principal point is known. Thus, for a system of m views taken by the same camera, we need only add one parameter for focal length giving $7m + 1$ parameters.

6.5 Algorithms Compared

Fundamentally, the rectifying homography $\mathbf{H}$ is encoded for by the absolute dual quadric $\mathbf{Q}_\infty^*$, which is defined by the plane at infinity π_∞ and absolute dual conic Ω_∞^*. These relationships (explained in detail in Appendix 6.2.2) define a natural categorization of autocalibration algorithms.

In linear methods, $\mathbf{Q}_\infty^*$ is estimated directly. In nonlinear methods $\mathbf{H}$ is improved nonlinearly. In stratified methods a brute force search is first used to identify π_∞ (after computing a finite bounding volume via chirality constraints) and then Ω_∞^* is estimated in a second phase. Finally, in dual stratified methods Ω_∞^* is first guessed based on prior knowledge and then π_∞ is estimated in a second phase.

Therefore, we compare against representative algorithms from each category. The specific algorithms we compare against are:

Linear Method (L). We use the linear method of Pollefeys et al. [1998]; Hartley and Zisserman [2004] to estimate $\mathbf{Q}_\infty^*$ using the symmetric parameterization of $\mathbf{Q}_\infty^*$ (Section 6.5.2). We have found that this is much more reliable than using the reduced parameterization. The constraints we use are based on the assumptions of zero skew, unit aspect ratio, and zero principal point, with experimentally determined weighting coefficients of $w_s = 1$, $w_r = 1$, $w_u = w_v = 0.2$.

Linear with Nonlinear Method (L+NL). A nonlinear improvement is also given in Pollefeys et al. [1998]; Hartley and Zisserman [2004] which can be parameterized by the rectifying homography to implicitly enforce the rank and positive-semidefinite constraints (Section 6.5.3). It is also necessary to parameterize $\omega^{*j} \forall j$ using some problem-dependent constraints. We found that this method has a tendency to become extremely unstable if the parameterization of ω^{*j} allows for nonzero principal point or varying focal length, so we use only a single parameter for constant focal length.

Stratified Method (S). We used Hartley's stratified approach [Hartley et al., 1999] with a brute force search for π_∞ (Section 6.5.4). We used the recommended discretization of $100 \times 100 \times 100$. Additionally, we augment the search space with one additional point representing the location of π_∞ that would be found by the linear algorithm. When solving for the absolute conic, we include constraints for zero skew, constant focal length, unit aspect ratio and constant principal point.

Stratified with Nonlinear Method (S+NL). Hartley [Hartley et al., 1999] calls for a nonlinear improvement after the initial stratified search so we follow by using the method of Pollefeys et al. [1998].

Dual Stratified Method (DS). We have used the method of Gherardi and Fusiello [2010] (Section 6.5.5), with the recommended parameters of 50 samples for focal length, $w_{sk} = 1/0.01$, $w_{ar} = 1/0.2$, $w_{u0} = 1/0.1$, $w_{v0} = 1/0.1$.

Dual Stratified with Nonlinear Method (DS+NL). It is recommended in Gherardi and Fusiello [2010] to follow up with a nonlinear improvement so we use the method of Pollefeys et al. [1998].

(NEW) Maximum Likelihood Method (ML). We initialize the rectifying homography using DS-MC-ML (Algorithm 2) and then follow up by nonlinearly minimizing (6.8) using Levenberg-Marquardt.

(NEW) ML with Resection Method (ML+R). We follow up the ML estimation of the rectifying homography by resectioning the cameras as described in Section 6.4.2. This ensures that metric constraints are satisfied exactly in the solution.

6.5.1 Parameterization

From the previous section it is clear that autocalibration is equivalent to determining either $\mathbf{H}_{\mathtt{M}}$, $\mathbf{Q}_\infty^*$, or $\{\Omega_\infty^*, \pi_\infty\}$. In this section we will discuss various ways to parameterize the solution in more detail.

The simplest and most general parameterization is to use the elements of $\mathbf{H}_{\mathtt{M}}$ directly. The advantage of parameterizing by $\mathbf{H}_{\mathtt{M}}$ is that any invertible $\mathbf{H}_{\mathtt{M}}$ represents a valid solution, making it the ideal choice for a nonlinear minimization.

It is not necessary to use all 16 dof to represent $\mathbf{H}_{\mathtt{M}}$, because the final column will be eliminated when multiplied by $\tilde{\mathtt{I}}$. Thus, a general parameterization is given by

$$\mathbf{H}_{\mathtt{M}} = \begin{bmatrix} h_1 & h_2 & h_3 & 0 \\ h_4 & h_5 & h_6 & 0 \\ h_7 & h_8 & h_9 & 0 \\ h_{10} & h_{11} & h_{12} & 1 \end{bmatrix}. \tag{6.30}$$

This can be further reduced by using a canonical reference frame. Without loss of generality, we can align the projective reconstruction such that $\mathbf{P}^1 = [\mathtt{I}|\mathbf{0}]$, and choose our metric reconstruction such that

$$\mathbf{P}^1_{\mathtt{M}} = \mathbf{K}^1[\mathtt{I}|0]. \tag{6.31}$$

Then, because

$$\mathbf{P}^1_{\mathtt{M}} = \mathbf{P}^1\mathbf{H}_{\mathtt{M}}, \tag{6.32}$$

it can be easily verified that $\mathbf{H}_{\mathtt{M}}$ must be of the form

$$\mathbf{H}_{\mathtt{M}} = \begin{bmatrix} \mathbf{K}^1 & 0 \\ \mathbf{v}^\mathsf{T} & 1 \end{bmatrix}. \tag{6.33}$$

In this case, $\mathbf{v}^\mathsf{T}$ is related to $\mathbf{K}^1$ and π_∞ by

$$\pi_\infty = \mathbf{H}_{\mathtt{M}}^{-\mathsf{T}} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} -(\mathbf{K}^1)^{-\mathsf{T}}\mathbf{v} \\ 1 \end{bmatrix}. \tag{6.34}$$

Another option is to parameterize directly by $\mathbf{Q}^*_\infty$. For example, this is useful in linear estimation because it is not possible to linearly estimate $\mathbf{H}_{\mathtt{M}}$. One possible parameterization

is simply to use the 10 unique symmetric elements of $\mathbf{Q}_\infty^*$. However, this obviously does not enforce the rank and semidefinite constraints.

Alternatively, if we transform into the canonical frame then $\Omega_\infty^* = \omega^{*1}$, and from (6.33-6.34) $\mathbf{Q}_\infty^*$ can be written explicitly in terms of $\pi_\infty = (\mathbf{p}^\mathsf{T}, 1)^\mathsf{T}$ and Ω_∞^* as

$$\begin{aligned}\mathbf{Q}_\infty^* &= \begin{bmatrix} \mathtt{I}|\mathbf{0} \\ -{\pi_\infty}^\mathsf{T} \end{bmatrix} \begin{bmatrix} \Omega_\infty^* & \mathbf{0} \\ \mathbf{0}^\mathsf{T} & 0 \end{bmatrix} \begin{bmatrix} \mathtt{I}|\mathbf{0} \\ -{\pi_\infty}^\mathsf{T} \end{bmatrix}^\mathsf{T} \\ &= \begin{bmatrix} \Omega_\infty^* & -\Omega_\infty^*\mathbf{p} \\ -\mathbf{p}^\mathsf{T}\Omega_\infty^* & \mathbf{p}^\mathsf{T}\Omega_\infty^*\mathbf{p} \end{bmatrix}.\end{aligned} \tag{6.35}$$

By using (6.33) or (6.35) it is clear that, if we make the choice of reference frames explicit, then 8 parameters are sufficient to define $\mathbf{H}_\mathtt{M}$. We refer to these as *reduced* parameterizations.

If one further assumes that there is zero skew, aspect ratio is unity, and that the principal point is known, then focal length is the only unknown element of $\mathbf{K}^1$ or Ω_∞^*, and hence we can reduce the parameterization to just 4 numbers. We refer to this as a *constrained* parameterization. A constrained parameterization is possible in the linear case as well [Pollefeys et al., 1998]. Multiplying out (6.35), we obtain

$$\mathbf{Q}_\infty^* = \begin{bmatrix} f_1^2 & 0 & 0 & f_1 v_x \\ 0 & f_1^2 & 0 & f_1 v_y \\ 0 & 0 & 1 & v_z \\ f_1 v_x & f_1 v_y & vz & v_x^2 + v_y^2 + v_z^2 \end{bmatrix}, \tag{6.36}$$

where $\mathbf{v} = (v_x, v_y, v_z)^\mathsf{T}$. Thus, the constrained linear parameters can be taken as

$$\mathbf{Q}_\infty^* = \begin{bmatrix} a & 0 & 0 & b \\ 0 & a & 0 & c \\ 0 & 0 & 1 & d \\ b & c & d & e \end{bmatrix}. \tag{6.37}$$

Note that $e = (b^2 + c^2)/a + d^2$ would not be strictly enforced by a linear estimate, but is required for consistency (rank 3 and semidefinite).

It should be noted that there is a significant cost to using the constrained parameterizations, because even when the true configuration satisfies the constraint assumptions, the projective reconstruction does not. Therefore, it is incorrect to assume that there will exist a rectifying homography that causes the constraints to be satisfied.

6.5.2 Linear Method

Given certain constraints on the calibration matrices, it is possible to form linear constraint equations on $\mathbf{Q}_\infty^*$. For example, one can typically assume that the image is not skewed or stretched and that the center of projection is in the center of the image. From these particular assumptions, (6.16) provides 4 linear constraints on $\mathbf{Q}_\infty^*$ for each view [Pollefeys et al., 1998] that are given by

$$\left(\mathbf{P}^j\mathbf{Q}_\infty^*\mathbf{P}^{j\mathsf{T}}\right)_{11} = \left(\mathbf{P}^j\mathbf{Q}_\infty^*\mathbf{P}^{j\mathsf{T}}\right)_{22} \tag{6.38}$$

$$\left(\mathbf{P}^j\mathbf{Q}_\infty^*\mathbf{P}^{j\mathsf{T}}\right)_{12} = 0 \tag{6.39}$$

$$\left(\mathbf{P}^j\mathbf{Q}_\infty^*\mathbf{P}^{j\mathsf{T}}\right)_{13} = 0 \tag{6.40}$$

$$\left(\mathbf{P}^j\mathbf{Q}_\infty^*\mathbf{P}^{j\mathsf{T}}\right)_{23} = 0. \tag{6.41}$$

A linear estimate is straight-forward to compute using the singular value decomposition of the constraint matrix, using either the general symmetric or constrained (6.37) parameterizations. The equations can optionally be weighted to enforce certain constraints more strongly. However, it should be noted that the weights do not exactly correspond to the parameters they are intended to constrain in the presence of noise.

We have noticed that the linear solution is not invariant, and in fact highly sensitive to the projective ambiguity. This motivates a simple extension whereby one perturbs the reconstruction by arbitrary homographies and re-attempts the linear autocalibration, keeping track of the solution with the lowest error. We have found that this adds a great deal of robustness to the algorithm when dealing with small numbers of views, and it is worthwhile to mention for its simplicity. We refer to this as a *linear distortion search*.

6.5.3 Nonlinear Method

The linear method has many weaknesses; the weights do not directly correspond to the parameters they are intended to, internal constraints (ie, rank, positive-semidefinite) cannot be enforced, and it only works when principle point is known exactly (and assumed zero), and it is quite sensitive to this assumption. The nonlinear method overcomes these issues by minimizing

$$\sum_{j=1}^{m}\left\|\frac{\omega^{*j}}{\|\omega^{*j}\|} - \frac{\mathbf{P}^j\mathbf{H}\tilde{\mathtt{I}}\mathbf{H}^{\mathsf{T}}\mathbf{P}^{j\mathsf{T}}}{\left\|\mathbf{P}^j\mathbf{H}\tilde{\mathtt{I}}\mathbf{H}^{\mathsf{T}}\mathbf{P}^{j\mathsf{T}}\right\|}\right\|^2, \tag{6.42}$$

where the Frobenius norm has been used to remove the scale ambiguity [Pollefeys et al., 1998;

Hartley and Zisserman, 2004] (this is more effective than dividing by the lower right element, which tends to create a bias towards lower focal lengths). The free parameters must parameterize $\mathbf{H}$ and all of the ω^{*i}'s. Note that by explicitly parameterizing by $\mathbf{H}$, all of the rank and positive-semidefinite constraints of $\mathbf{Q}_\infty^*$ are easily accounted for.

6.5.4 Stratified Method

The basic idea of the stratified approach [Pollefeys and Gool, 1997; Pollefeys and Van Gool, 1999; Hartley et al., 1999; Hartley and Zisserman, 2004] is to do a brute force search for $\pi_\infty \in \mathbb{R}^3$ first. It turns out that if π_∞ is exactly known, there is a linear algorithm for estimating ω^*, and these two entities combined define $\mathbf{H}_\mathtt{M}$. Given a candidate homography and a heuristic measure of error, the homography that minimizes this error can be selected.

The method for calculating Ω_∞^* from π_∞ is linear and relies upon the planar homography between Ω_∞^* and the ω^{*j} matrices. Once π_∞ is known one can calculate the infinite homographies in (6.17), and thus each camera can be used to derive constraints on Ω_∞^* according to (6.18).

In order for a brute force search for $\pi_\infty = (\mathbf{p}^\mathsf{T}, 1)^\mathsf{T}$ to be possible we must first obtain a finite bounding volume for $\mathbf{p} \in \mathbb{R}^3$. In a general projective reconstruction, π_∞ could be literally anywhere, and it usually can be found slicing its way through the reconstructed point cloud, causing the point cloud to be distributed from one end of infinity to the other.

The first step is upgrading the reconstruction to quasi-affine, which involves multiplying points and cameras by -1 until the projective depths of all points in all views is positive. In a quasi-affine reconstruction π_∞ is guaranteed to not cut through the point cloud. In an affine reconstruction, the plane at infinity is in its proper place $\pi_\infty = (0, 0, 0, 1)^\mathsf{T}$, and we call this a quasi-affine reconstruction because π_∞ is at least outside of the point cloud.

From a quasi-affine reconstruction, chirality constraints can be solved in order to obtain a "strong" quasi-affine reconstruction, in which all points are in front of all cameras. At this point it is important to normalize the principal components of the point cloud for numerical stability, otherwise the entire point cloud will appear to happily live embedded in a plane in $\mathbb{R}^3$. Finally, the chirality constraints can be solved to obtain bounds on each parameter of the plane at infinity.

Two alternative sets of constraints for estimating Ω_∞^* were suggested in [Hartley and Zisserman, 2004, alg 19.2], one based on assuming zero skew and the other based on assuming constant calibration matrices. One of the biggest advantages of the stratified approach is that one can linearly estimate ω (the inverse of Ω_∞^*) using

$$\omega^j \propto {\mathbf{H}_\infty^j}^{-\mathsf{T}} \omega {\mathbf{H}_\infty^j}^{-1} \tag{6.43}$$

based only on an assumption of zero skew, whereas the linear estimation of $\mathbf{Q}_\infty^*$ requires knowledge of the principal point. However, there is an inconsistency in the method based on constant calibration matrices, and we prefer to use more than just a zero skew constraint.

If one assumes constant calibration matrices (up to scale), then $\mathbf{K}^i \propto \mathbf{K}^j \quad \forall ij$, which implies that $\omega^{*i} \propto \omega^{*j} \quad \forall ij$. In other words, the assumption is that calibration matrices for each camera in the reconstruction should be equal when in the metric frame.

However, this does not imply that *all* projections of $\mathbf{Q}_\infty^*$ should be equal – only the projections of $\mathbf{Q}_\infty^*$ by the *metric rectified* cameras should be equal. In particular, $\Omega_\infty^* \neq \omega^{*j}$. This is obvious if one considers the metric frame, in which $\Omega_\infty^* = \mathtt{I}$ but $\omega^{*j} = \mathbf{K}^j \mathbf{K}^{j\mathsf{T}}$. Only when one of the estimated cameras in the current projective frame has the canonical form of $\mathbf{P}^j = [\mathtt{I}|0]$ can we say that $\Omega_\infty^* = \omega^{*j}$.

Because the algorithm in [Hartley and Zisserman, 2004, alg 19.2] calls for a normalizing transform after the quasi-affine upgrade (which we find is quite necessary for numerical stability), no camera can be assumed to have canonical form, yet the algorithm still incorrectly assumes $\Omega_\infty^* = \omega^{*j}$.

We correct for this inconsistency here and show how constancy constraints can still be used. Using Hartley's parameterization (6.10), ω^j expands to

$$\omega^j = \begin{bmatrix} \alpha_y^2 & -s\alpha_y & -u\alpha_y^2 + vs\alpha_y \\ & \alpha_x^2 + s^2 & \alpha_y su - \alpha_x^2 v - s^2 v \\ sym & & \alpha_x^2\alpha_y^2 + \alpha_x^2 v^2 + (\alpha_y u - sv)^2 \end{bmatrix}. \tag{6.44}$$

Because scale is irrelevent, the parameterization of ω may assume $\omega_{3,3} = 1$ so that only 5 numbers are needed (as opposed to the 6 used in [Hartley and Zisserman, 2004, alg 19.2]). We will also require 1 parameter for each constant value (c_1, c_2, ...).

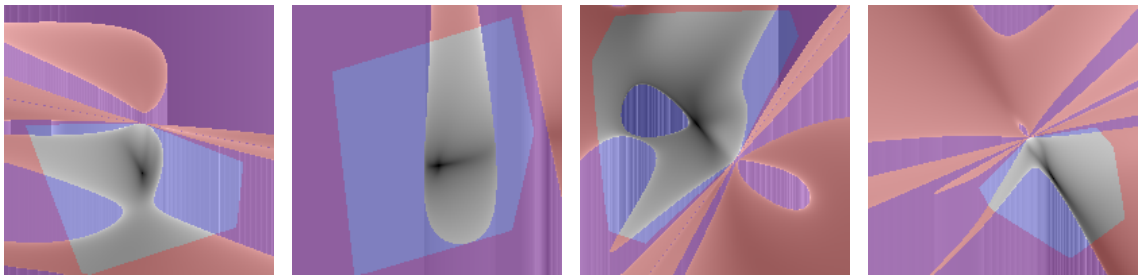
A good set of constraints to use when all views come from the same physical camera are: zero skew ($\omega_{1,2}^j = 0$), constant but unknown principal point ($\omega_{1,3}^j = c_2$, $\omega_{2,3}^j = c_3$), and constant but unknown focal length with unit aspect ratio ($\omega_{1,1}^j = c$, $\omega_{2,2}^j = c$). Thus, we have 7 parameters and $5m$ equations for m views making a solution possible from 2 or more views. If focal length is changing, then the latter two constraints can be replaced with a single constraint for unit aspect ratio ($\omega_{1,1}^j - \omega_{2,2}^j = 0$).

To summarize the stratified approach,

1. Multiply cameras and structure points by -1 as necessary to obtain a quasi-affine reconstruction.

2. Compute the convex hull of the set of points and camera centers.

3. Solve a LP problem to find a suitable location for π_∞ that satisfies all chirality constraints to obtain a strong quasi-affine reconstruction.



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

Figure 6.8. Example 2D slices through the 3D cost-volume associated with the location of π_∞. The red tinted area is rejected due to chirality constraints and the blue tinted area is rejected due to the positive-semidefinite constraint of Ω_∞^*.

4. Solve the 6 LP problems to find a bounding volume for π_∞ that contains all solutions satisfying the chirality constraints.

5. Discretize and exhaustively explore the space. For each candidate position of π_∞,

   (a) Evaluate the chirality constraints against the convex hull points, and reject if they are not all satisfied.

   (b) Linearly estimate Ω_∞^* (via ω) using the assumptions of zero skew, aspect ratio of unity, constant unknown principal point, and (optionally) constant focal length.

   (c) If Ω_∞^* is not positive-semidefinite, reject the solution.

   (d) Evaluate the desired objective (eg, prior probability, least squares residual, or constrained likelihood), and record the solution if it is minimal.

It is not necessary to compute the convex hull points, but this reduces the time complexity of the rejection test in the brute force search, which may be important if using an extremely fine discretization or if there are a large number of structure points.

Some example cost surfaces produced using the stratified approach with the least squares residual heuristic are shown in Fig. 6.8. These graphs show that the cost surface is actually rather smooth, meaning that a brute force stratified search is not really well justified. We do not know how to reconcile these results with the chaotic landscapes observed in [Hartley et al., 1999] for the same problem.

6.5.5 Dual Stratified Method

In the regular stratified approach, one searches for π_∞ and uses a linear algorithm to infer the remaining parameters – namely, $\mathbf{K}_1$. The trouble with this approach is that π_∞ could lie

anywhere in $\mathcal{P}^3$, and hence it is necessary to solve all those clumsy linear programming problems to find real bounds on the parameters and then brute force search within that parameter space. Even more troubling, however, is that the determination of the parameters of $\mathbf{K}$ are often ill-conditioned [Bougnoux, 1998], and hence it is not guaranteed that a linear estimation would find good values for $\mathbf{K}$ even from the perfect choice of π_∞.

In fact, a completely uninformed guess as to the parameters of $\mathbf{K}$ is sometimes more accurate than solving for those parameters. Skew can be assumed zero, the principal point in the center of the image. The precise focal length is not known, but the practical range of focal lengths is actually a tighter constraint than the uncertainty in estimation assuming the parameter is free using the classic linear/nonlinear methods. Thus, even when focal length is unknown, one can simply guess a value that is a reasonable estimate. However, metric structure cannot be recovered until π_∞ has also been found.

6.5.5.1 Linear Least Squares Solution

One naturally wonders if there is a direct algorithm for estimating $\pi_\infty = (\mathbf{p}^\mathsf{T}, 1)^\mathsf{T}$ from known $\mathbf{K}$. Indeed, there was described in [Bougnoux, 1998] a linear least squares algorithm using constraints from 2 or more views for estimating π_∞ based on the assumption (or approximation) that $\mathbf{K}^i$ are all equal, and using the reduced parameterization for $\mathbf{H}$ (6.35).

We re-derive that result here using more conventional notation. Let the parameters of any general projective camera be given by $\mathbf{P} = [\mathbf{M}|\mathbf{t}]$ with

$$\mathbf{M}^\mathsf{T} = [\mathbf{m}_1|\mathbf{m}_2|\mathbf{m}_3] \tag{6.45}$$

$$\mathbf{t} = (t_1, t_2, t_3)^\mathsf{T} \tag{6.46}$$

Assuming that $\mathbf{K}^i = \mathbf{K}^j \forall i, j$ and one of the cameras is canonical, this implies $\Omega^*_\infty = \omega^{*j}$, so we just drop the subscripts on all these, and say that $\Omega^*_\infty = \mathbf{K}^i \mathbf{K}^{i\mathsf{T}}$. Thus,

$$\Omega^*_\infty \propto \mathbf{P}\mathbf{Q}^*_\infty\mathbf{P}^\mathsf{T} \tag{6.47}$$

$$\propto \mathbf{P}\mathbf{H}\tilde{\mathtt{I}}(\mathbf{P}\mathbf{H})^\mathsf{T} \tag{6.48}$$

now using the reduced parameterization for $\mathbf{H}$, we make the scale factor explicit as α and expand (6.48) to

$$\alpha\Omega^*_\infty = \left(\mathbf{MK} + \mathbf{tv}^\mathsf{T}\right)\left(\mathbf{MK} + \mathbf{tv}^\mathsf{T}\right)^\mathsf{T} \tag{6.49}$$

$$= \mathbf{M}\Omega^*_\infty\mathbf{M}^\mathsf{T} + \mathbf{tv}^\mathsf{T}\mathbf{K}^\mathsf{T}\mathbf{M}^\mathsf{T} + \mathbf{MKvt}^\mathsf{T} + ||\mathbf{v}||^2\mathbf{tt}^\mathsf{T} \tag{6.50}$$

Using the fact that $\Omega^*_{\infty 33} = 1$, this allows us to write the scale factor explicitly as

$$\alpha = \left(\mathbf{M}\Omega^*_\infty\mathbf{M}^\mathsf{T} + \mathbf{tv}^\mathsf{T}\mathbf{K}^\mathsf{T}\mathbf{M}^\mathsf{T} + \mathbf{MKvt}^\mathsf{T} + ||\mathbf{v}||^2\mathbf{tt}^\mathsf{T}\right)_{33} \tag{6.51}$$

$$= \mathbf{m}_3^\mathsf{T}\Omega^*_\infty\mathbf{m}_3 + 2\mathbf{m}_3^\mathsf{T}\mathbf{Kv}t_3 + ||\mathbf{v}||^2 t_3^2 \tag{6.52}$$

Now substituting (6.52) back into (6.50), we obtain

$$\begin{aligned}(\mathbf{m}_3^\mathsf{T}\Omega^*_\infty\mathbf{m}_3 + 2\mathbf{m}_3^\mathsf{T}\mathbf{Kv}t_3 + ||\mathbf{v}||^2 t_3^2)\Omega^*_\infty = \\ \mathbf{M}\Omega^*_\infty\mathbf{M}^\mathsf{T} + \mathbf{tv}^\mathsf{T}\mathbf{K}^\mathsf{T}\mathbf{M}^\mathsf{T} + \mathbf{MKvt}^\mathsf{T} + ||\mathbf{v}||^2\mathbf{tt}^\mathsf{T}\end{aligned} \tag{6.53}$$

which can be rearranged to

$$\begin{aligned}||\mathbf{v}||^2(\mathbf{tt}^\mathsf{T} - t_3^2\Omega^*_\infty) = \mathbf{M}\Omega^*_\infty\mathbf{M}^\mathsf{T} + \mathbf{tv}^\mathsf{T}\mathbf{K}^\mathsf{T}\mathbf{M}^\mathsf{T} + \mathbf{MKvt}^\mathsf{T} \\ - \mathbf{m}_3^\mathsf{T}\Omega^*_\infty\mathbf{m}_3\Omega^*_\infty - 2\mathbf{m}_3^\mathsf{T}\mathbf{Kv}t_3\Omega^*_\infty\end{aligned} \tag{6.54}$$

Because (6.54) is a symmetric 3×3 matrix equation, it gives 5 equations for $||\mathbf{v}||^2$ (excluding the lower right equation). These equations may be enumerated by any choice of i, j such that $i \in \{1, 2\}$ and $j \in [i \ldots 3]$,

$$||\mathbf{v}||^2 = (\mathbf{M}\Omega^*_\infty\mathbf{M}^\mathsf{T} + \mathbf{tv}^\mathsf{T}\mathbf{K}^\mathsf{T}\mathbf{M}^\mathsf{T} + \mathbf{MKvt}^\mathsf{T} \tag{6.55}$$

$$- \mathbf{m}_3^\mathsf{T}\Omega^*_\infty\mathbf{m}_3\Omega^*_\infty - 2\mathbf{m}_3^\mathsf{T}\mathbf{Kv}t_3\Omega^*_\infty)_{ij} / \left(\mathbf{tt}^\mathsf{T} - t_3^2\Omega^*_\infty\right)_{ij} \tag{6.56}$$

$$= (\mathbf{m}_i^\mathsf{T}\Omega^*_\infty\mathbf{m}_j + t_i\mathbf{v}^\mathsf{T}\mathbf{K}^\mathsf{T}\mathbf{m}_j + \mathbf{m}_i^\mathsf{T}\mathbf{Kv}t_j \tag{6.57}$$

$$- \mathbf{m}_3^\mathsf{T}\Omega^*_\infty\mathbf{m}_3\Omega^*_{\infty ij} - 2\mathbf{m}_3^\mathsf{T}\mathbf{Kv}t_3\Omega^*_{\infty ij}) / \left(t_i t_j - t_3^2\Omega^*_{\infty ij}\right) \tag{6.58}$$

with some further rearrangements, we can isolate $\mathbf{Kv}$,

$$||\mathbf{v}||^2 = (\mathbf{m}_i^\mathsf{T}\Omega_\infty^*\mathbf{m}_j + t_i\mathbf{m}_j^\mathsf{T}\mathbf{K}\mathbf{v} + t_j\mathbf{m}_i^\mathsf{T}\mathbf{K}\mathbf{v} \tag{6.59}$$

$$- \mathbf{m}_3^\mathsf{T}\Omega_\infty^*\mathbf{m}_3\Omega_{\infty ij}^* - 2\Omega_{\infty ij}^* t_3\mathbf{m}_3^\mathsf{T}\mathbf{K}\mathbf{v})/\left(t_i t_j - t_3^2\Omega_{\infty ij}^*\right) \tag{6.60}$$

$$= (\mathbf{m}_i^\mathsf{T}\Omega_\infty^*\mathbf{m}_j - \mathbf{m}_3^\mathsf{T}\Omega_\infty^*\mathbf{m}_3\Omega_{\infty ij}^* + \tag{6.61}$$

$$(t_i\mathbf{m}_j^\mathsf{T} + t_j\mathbf{m}_i^\mathsf{T} - 2\Omega_{\infty ij}^* t_3\mathbf{m}_3^\mathsf{T})\mathbf{K}\mathbf{v})/\left(t_i t_j - t_3^2\Omega_{\infty ij}^*\right) \tag{6.62}$$

One of these 5 equations can be substituted back into the other 4. It is probably best to choose the equation which maximizes the divisor $|t_i t_j - t_3^2\Omega_{\infty ij}^*|$. Let the indices of this chosen equation be $\hat{i}, \hat{j}$. Then, the remaining 4 equations after substituting are given by varying i, j in

$$\begin{aligned}\left(t_{\hat{i}}t_{\hat{j}} - t_3^2\Omega_{\infty \hat{i}\hat{j}}^*\right)(\mathbf{m}_i^\mathsf{T}\Omega_\infty^*\mathbf{m}_j - \mathbf{m}_3^\mathsf{T}\Omega_\infty^*\mathbf{m}_3\Omega_{\infty ij}^* \\ +(t_i\mathbf{m}_j^\mathsf{T} + t_j\mathbf{m}_i^\mathsf{T} - 2\Omega_{\infty ij}^* t_3\mathbf{m}_3^\mathsf{T})\mathbf{K}\mathbf{v}) = \\ \left(t_3^2\Omega_{\infty ij}^* - t_i t_j\right)(\mathbf{m}_{\hat{i}}^\mathsf{T}\Omega_\infty^*\mathbf{m}_{\hat{j}} - \mathbf{m}_3^\mathsf{T}\Omega_\infty^*\mathbf{m}_3\Omega_{\infty \hat{i}\hat{j}}^* \\ +(t_{\hat{i}}\mathbf{m}_{\hat{j}}^\mathsf{T} + t_{\hat{j}}\mathbf{m}_{\hat{i}}^\mathsf{T} - 2\Omega_{\infty \hat{i}\hat{j}}^* t_3\mathbf{m}_3^\mathsf{T})\mathbf{K}\mathbf{v})\end{aligned} \tag{6.63}$$

Notice that these equations are linear in $\mathbf{v}$ and may therefore be solved using traditional methods of linear least squares, with 4 equations being provided for each view beyond the first (assuming the first view is canonical). If desired, π_∞ can then be recovered from $\mathbf{v}$ according to (6.34).

6.5.5.2 Closed Form Solution

It was later shown that, in the special case of 2 views, there is a unique closed form solution that does not assume the calibration matrices are equal [Gherardi and Fusiello, 2010]. We will derive this result below to correct for some ambiguities in the original publication.

Assume that the true metric calibration matrices $\mathbf{K}_1$ and $\mathbf{K}_2$ corresponding to the projective cameras $\mathbf{P}_1$ and $\mathbf{P_2}$. If all four of these entities are accurate, then there should exist a homography $\mathbf{H}_\mathtt{M}$ that satisfies

$$\mathbf{P}_1\mathbf{H}_\mathtt{M} \propto \mathbf{K}_1[\mathbf{R}_1|\mathbf{t}_1] \tag{6.64}$$

$$\mathbf{P}_2\mathbf{H}_\mathtt{M} \propto \mathbf{K}_2[\mathbf{R}_2|\mathbf{t}_2]. \tag{6.65}$$

The first step is to transform so that the first projection matrix is canonical. We can always find $\mathbf{H}$ such that

$$\mathbf{P}_1\mathbf{H} = \mathbf{P}'_1 = [\mathtt{I}|\mathbf{0}] \tag{6.66}$$

$$\mathbf{P}_2\mathbf{H} = \mathbf{P}'_2 = [\mathbf{Q}_2|\mathbf{q}_2]. \tag{6.67}$$

Specifically, a non-singular choice of $\mathbf{H}$ is given by

$$\mathbf{H} = \mathbf{P}_1^+[\mathtt{I}|\mathbf{0}] + \mathbf{C}(0,0,0,1)^\mathsf{T}, \tag{6.68}$$

where $\mathbf{P}_1\mathbf{C} = \mathbf{0}$. Using the reduced parameterization

$$\mathbf{H}'_{\mathtt{M}} = \begin{bmatrix} \mathbf{K}_1 & \mathbf{0} \\ \mathbf{v}'^\mathsf{T} & 1 \end{bmatrix}, \tag{6.69}$$

we may relate $\mathbf{P}'_1$ and $\mathbf{P}'_2$ to their metric counterparts as

$$\mathbf{P}'^{\mathtt{M}}_1 = \mathbf{P}'_1\mathbf{H}'_{\mathtt{M}} = \mathbf{K}_1[\mathtt{I}|\mathbf{0}] \tag{6.70}$$

$$\mathbf{P}'^{\mathtt{M}}_2 = \mathbf{P}'_2\mathbf{H}'_{\mathtt{M}} = \mathbf{K}_2[\mathbf{R}_2|\mathbf{t}_2] = \alpha[\mathbf{Q}_2\mathbf{K}_1 + \mathbf{q}_2\mathbf{v}'^\mathsf{T}|\mathbf{q}_2], \tag{6.71}$$

where α is an explicit scale factor.

It is important to note the presence of this scale factor because the scale of $\mathbf{K}_2$ is fixed and known, and this fixes the scale factor of the left hand side. The matrix $\mathbf{P}'_2$ is treated as a known value, but it's scale was chosen arbitrarily, so this fixes the scale of the right hand side. Thus, α is not a free parameter but rather an unknown constant value.

Continuing, (6.71) can be broken up into two equations,

$$\mathbf{K}_2\mathbf{R}_2 = \alpha(\mathbf{Q}_2\mathbf{K}_1 + \mathbf{q}_2\mathbf{v}'^\mathsf{T}) \tag{6.72}$$

$$\mathbf{K}_2\mathbf{t}_2 = \alpha\mathbf{q}_2. \tag{6.73}$$

Define the rotation $\mathbf{R}^*$ such that

$$\mathbf{R}^*\frac{\mathbf{K}_2^{-1}\mathbf{q}_2}{||\mathbf{K}_2^{-1}\mathbf{q}_2||} = (1,0,0)^\mathsf{T}. \tag{6.74}$$

Then we can rearrange and left multiply (6.71) by $\mathbf{R}^*$ to get

$$\mathbf{R}^*\mathbf{R}_2 = \alpha \underbrace{\mathbf{R}^*\mathbf{K}_2^{-1}\mathbf{Q}_2\mathbf{K}_1}_{\mathbf{W}} + \alpha(||\mathbf{K}_2^{-1}\mathbf{q}_2||, 0, 0)^\mathsf{T}\mathbf{v}'^\mathsf{T} \tag{6.75}$$

$$= \begin{bmatrix} \alpha\mathbf{w}_1^\mathsf{T} + \alpha||\mathbf{K}_2^{-1}\mathbf{q}_2||\mathbf{v}'^\mathsf{T} \\ \alpha\mathbf{w}_2^\mathsf{T} \\ \alpha\mathbf{w}_3^\mathsf{T} \end{bmatrix} \tag{6.76}$$

Although we don't know what α is, we do know that the left hand side is a rotation matrix, and therefore so is the right hand side. For a rotation matrix, each row or column should have unit magnitude. Thus, we are free to divide by the magnitude of the bottom row without changing the truth of the above equation, and this cancels out the pesky scale factor,

$$\mathbf{R}^*\mathbf{R}_2 = \frac{1}{||\mathbf{w}_3||} \begin{bmatrix} \mathbf{w}_1^\mathsf{T} + ||\mathbf{K}_2^{-1}\mathbf{q}_2||\mathbf{v}'^\mathsf{T} \\ \mathbf{w}_2^\mathsf{T} \\ \mathbf{w}_3^\mathsf{T} \end{bmatrix}. \tag{6.77}$$

For any rotation matrix $\mathbf{R}$, it holds that $\mathbf{r}_1^\mathsf{T} = \mathbf{r}_2^\mathsf{T} \times \mathbf{r}_3^\mathsf{T}$, where $\mathbf{r}_i$ is the ith row of $\mathbf{R}$. Thus, from the right hand side, we may obtain

$$\frac{\mathbf{w}_1 + ||\mathbf{K}_2^{-1}\mathbf{q}_2||\mathbf{v}'}{||\mathbf{w}_3||} = \frac{\mathbf{w}_2}{||\mathbf{w}_3||} \times \frac{\mathbf{w}_3}{||\mathbf{w}_3||} = \frac{\mathbf{w}_2 \times \mathbf{w}_3}{||\mathbf{w}_3||^2} \tag{6.78}$$

$$\mathbf{w}_1 + ||\mathbf{K}_2^{-1}\mathbf{q}_2||\mathbf{v}' = \frac{\mathbf{w}_2 \times \mathbf{w}_3}{||\mathbf{w}_3||}. \tag{6.79}$$

Thus,

$$\mathbf{v}' = \frac{\frac{\mathbf{w}_2 \times \mathbf{w}_3}{||\mathbf{w}_3||} - \mathbf{w}_1}{||\mathbf{K}_2^{-1}\mathbf{q}_2||}. \tag{6.80}$$

Of course, $\mathbf{v}'$ defines the remaining parameters of $\mathbf{H}'_{\mathtt{M}}$, the transformation which takes the transformed cameras $\mathbf{P}'_1$ and $\mathbf{P}'_2$ to metric. In the original reference frame, we must use

$$\mathbf{H}_{\mathtt{M}} = \mathbf{H}\mathbf{H}'_{\mathtt{M}}. \tag{6.81}$$

Plugging (6.80) back into (6.76), it can be verified that

$$\alpha = 1/||\mathbf{w}_3|| \tag{6.82}$$

$$\mathbf{t}_2 = \mathbf{K}_2^{-1}\mathbf{q}_2/||\mathbf{w}_3|| \tag{6.83}$$

$$\mathbf{R}_2 = \mathbf{K}_2^{-1}(\mathbf{Q}_2\mathbf{K}_1 + \mathbf{q}_2\mathbf{v}'^{\mathsf{T}})/||\mathbf{w}_3|| \tag{6.84}$$

6.6 Experimental Methods

We compare the proposed maximum likelihood method against several other representive autocalibration algorithms by generating synthetic configurations and then projecting structure points to obtain image points (a.k.a., image correspondences). The correspondences are corrupted by adding normally distributed noise, and then projective bundle adjustment is used to find the maximum likelihood projective reconstruction. Finally, we use each autocalibration algorithm to rectify the projective reconstruction and make an objective comparison to the true configuration.

6.6.1 Objective Evaluation

A reconstruction includes structure points, camera poses, and intrinsic camera parameters. Due to the large number of different types of parameters, identifying a good objective evaluation can be challenging.

It is common to see autocalibration algorithms compared purely on the basis of how accurately specific intrinsic camera parameters have been reconstructed. However, each camera has 5 intrinsic parameters, and the relative importance of each parameter is unclear, so there is no objective way to combine the various intrinsic parameters into a single measure of reconstruction quality. One could report the error for each parameter independently, but usually a reduction of the error in any one parameter forces an increase in the error of some other parameter. Thus, there would still be no clear objective way for the reader to assess which method was better.

A much more objective evaluation is to look at the structure of the reconstruction, because all of the structure points and camera centers in a perfect reconstruction should be within a similarity transformation of the true configuration. Therefore, when the true configuration is known one can simply factor out this similarity transformation to align the two point clouds and then measure the sum of squared distances between them. A linear solution for finding the similarity transformation that minimizes sum of squared distance is given in Umeyama [1991].

We have observed that all autocalibration algorithms tend to have much higher error in the reconstructed camera centers than in the reconstructed structure point cloud. Therefore, we

use only the structure point cloud to compute the alignment, and then use the mean squared error of reconstructed camera centers as the objective error. Any error in the intrinsic camera parameters necessitates shifting the camera center to compensate (e.g., a reduction in focal length moves the cameras closer), and thus by measuring error in camera center we obtain a truly objective measure of reconstructed camera accuracy that is independent of the specific objectives being optimized by the autocalibration algorithm.

6.6.2 Experiments

We generate random configurations with 2000 structure points distributed uniformly on the surface of a cube of width 100 centered at the origin. There are 10 cameras arranged on a circle of radius 1500 at 10° increments, with a positional jitter of ± 10, looking at a random point in a cube of width 40 centered at the origin. All cameras have a constant focal length in the range of 600-800 (relative to an image size of 640×480), zero skew, unit aspect ratio and zero principal point.

For each level of noise, we generate a set of 100 random configurations, project the image points and perturb with additive gaussian noise using $\sigma = 0$, $\sigma = 1$, or $\sigma = 3$ pixels. Then we run projective bundle adjustment to obtain the maximum likelihood projective reconstruction and attempt autocalibration.

The case of zero noise is not realistic, but validates that the algorithms have been implemented correctly. The case of $\sigma = 1$ represents a realistic level of noise for a typical subpixel matching algorithm after outliers have been removed using MLESAC [Torr and Zisserman, 2000] or some other variation of RANSAC [Fischler and Bolles, 1981]. The case of $\sigma = 3$ represents a larger level of noise that might be obtained by using less accurate multi-scale features, such as SIFT [Lowe, 1999] features.

For each algorithm and for each noise level, we computed the Empirical Cumulative Distribution Function (ECDF) of the objective error measure after autocalibrating all 100 configurations.

6.7 Results

For the unrealistic case of $\sigma = 0$ (i.e., the when there exists a homography that exactly rectifies the projective solution with no noise), we found that all methods performed extremely well, with the bulk of objective error (as described in Section 6.6.1) being approximately in the range of 10^{-27} to 10^{-15} for all algorithms. Because the stratified and dual stratified algorithms use discretized searches, their precision was worse before nonlinear improvement.

The ECDFs at a more realistic noise level of $\sigma = 1$ pixel are shown in Fig. 6.9. Here we see that many of the previous algorithms have difficulty achieving robust and accurate results. It

is clear that our ML+R and ML algorithms have superior precision and robustness to all of the other algorithms compared. The DS algorithm is a close runner up about 85% of the time, but gives unstable results the remaining 15% of the time. Surprisingly, the nonlinear improvement actually tends to worsen the DS algorithm at this level of noise, which we speculate is due to inherent ability of the DS algorithm to guarantee that at least two cameras have calibration matrices that meet our expectations, whereas the nonlinear method has the potential to diverge.

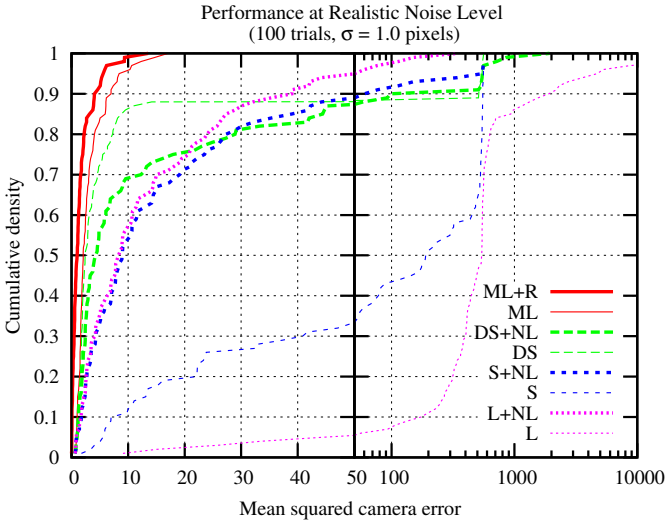


Figure 6.9. Empirical cumulative distribution of camera errors from 100 random configurations using each method of autocalibration. The initial projective reconstruction is obtained by projective bundle adjustment from image point measurements with normally distributed noise having $\sigma = 1.0$ pixels. The x-axis uses a log-scale for $x > 50$.

We show the ECDFs at a larger noise level of $\sigma = 3$ pixels in Fig. 6.10. At this high level of noise, the DS algorithm becomes very unreliable. Only our ML and ML+R algorithms continue to provide robust results, and the benefit of the final resectioning stage is more pronounced.

In Fig. 6.11, we evaluate autocalibration performance of the ML method at $\sigma = 1$ using 2,3,4,5,6 and 10 views. We have plotted the median error with interquartile range (IQR), and both asymptotically approach zero as the number of views increases, demonstrating stability for larger problems.

Although the ML+R method performs slightly better it is omitted from Fig. 6.11 for clarity because the median performance is almost the same. It should also be noted that lower error could be achieved with the same number of views by using a wider baseline, moving the points closer to the cameras, or reducing the added noise.

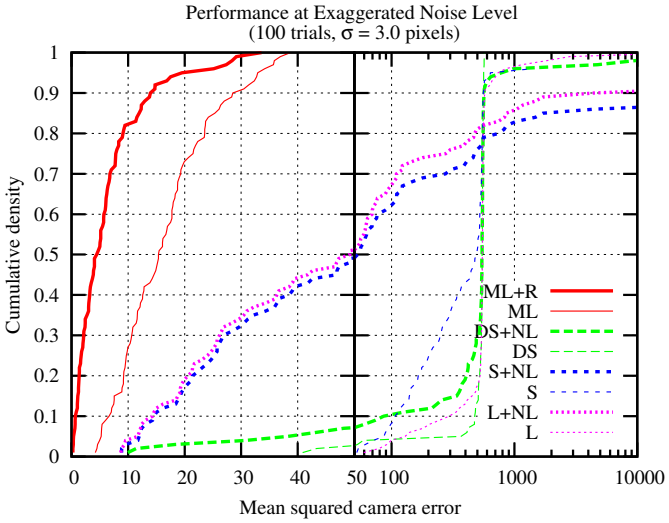


Figure 6.10. Empirical cumulative distribution of camera errors from 100 random configurations using each method of autocalibration. The initial projective reconstruction is obtained by projective bundle adjustment from image point measurements with normally distributed noise having $\sigma = 3.0$ pixels. The x-axis uses a log-scale for $x > 50$.

We demonstrate the runtime performance of our ML and ML+R autocalibration algorithms in Fig. 6.12, using constraints on aspect ratio, skew and principal point during the resectioning step. The timings are shown with 95% confidence intervals from 25 repetitions, and indicate that performance scales linearly with the number of views. Runtime is about 1.2 seconds for 1000 points in 64 views using the ML method, or 6 seconds when using the ML+R method, on a Core i7 920 processor.

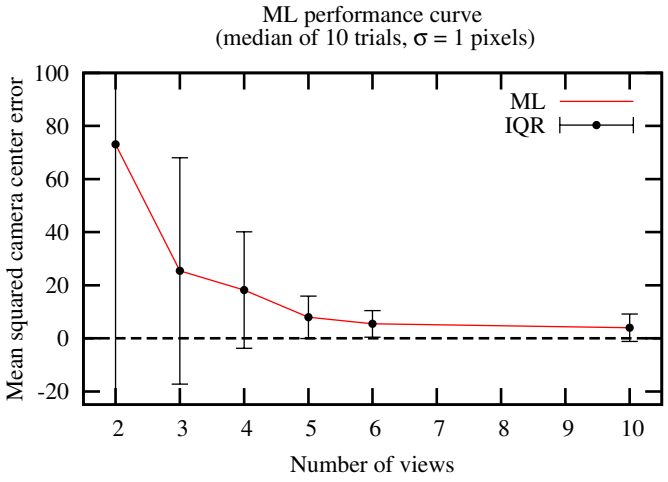


Figure 6.11. Objective autocalibration performance using 2,3,4,5,6 and 10 views at $\sigma = 1$. The median of 10 trials with interquartile range is plotted.

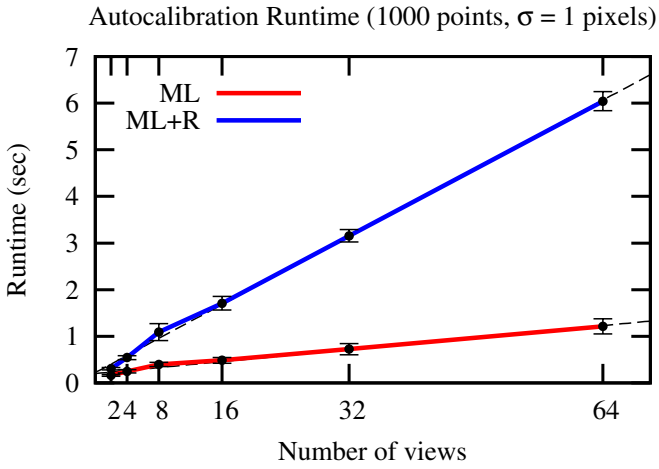


Figure 6.12. Runtime performance of ML and ML+R autocalibration routines, shown with 95% confidence intervals from 25 repetitions. Performance scales linearly with the number of views.

6.7.1 Examples on Real Data

In order to demonstrate our method on real data we obtained some reconstructions that were created using existing structure from motion systems. We will show that the method is general enough to be applied on any type of problem, such as a photo collection from different cameras, a photo collection from the same camera, and a video reconstruction. Our general approach

is to take an existing metric reconstruction and then apply projective bundle adjustment to corrupt it, giving us a ML projective reconstruction from which we attempt autocalibration. We have not used any metric bundle adjustment to improve the results.

It should be noted that this is a much more realistic test of autocalibration than simply multiplying an existing metric reconstruction by an arbitrary homography and then trying to recover this homography; as we have shown in the results section, all of the tested algorithms are capable of solving that problem extremely well. The difficulty in autocalibration is entirely due to the violation of metric constraints during projective bundle adjustment.

The success of an autocalibration algorithm can be assessed visually by looking at the reconstructed point cloud (Fig. 6.13). The point cloud of an arbitrary projective reconstruction (e.g., as obtained after projective bundle adjustment) is unbounded (Fig. 6.13b), and it is impossible to discern any meaningful structure. If π_∞ has been identified approximately correctly but Ω^*_∞ has not, then the convex hull will be bounded but the reconstruction will appear with a roughly affine skew as in Fig. 6.13c, and this is called a quasi-affine reconstruction. On the other hand, if Ω^*_∞ is identified correctly but π_∞ is not as in Fig. 6.13e, then the reconstruction will be unbounded and typically have a 'bow-tie' shape that spans from $-\infty$ to ∞, with some discernible structure near the origin. Only in a metric reconstruction are straight lines and angles preserved, and thus it is easy to identify a correct autocalibration of a cube, as computed by our ML algorithm in Fig. 6.13d, by observing 90° angles between adjacent faces. It is important to note that beyond correcting for an overall warping factor, autocalibration cannot be expected to remove noise from the individual structure points or camera positions.

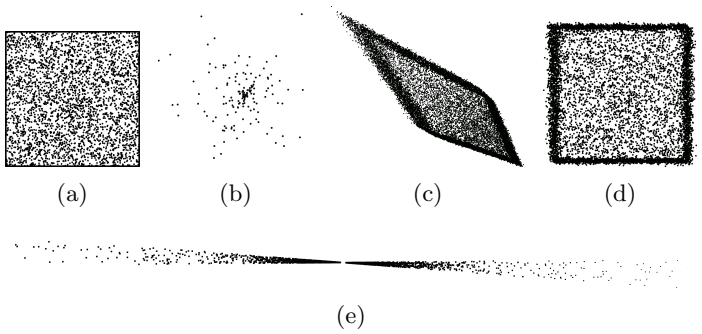


Figure 6.13. Example point clouds viewed from the top. (a) Points on the surface of a cube in the true configuration without noise. (b) A subset of the (unbounded) ML projective reconstruction after bundle adjustment. (c) A partially successful autocalibration has obtained a quasi-affine reconstruction where at least π_∞ does not intersect the convex hull. (d) A successful autocalibration is visually identified by preserving right-angles. (e) A failed autocalibration attempt where π_∞ intersects the convex hull, sending reconstructed points to infinity and producing a characteristic 'bow-tie' shape.

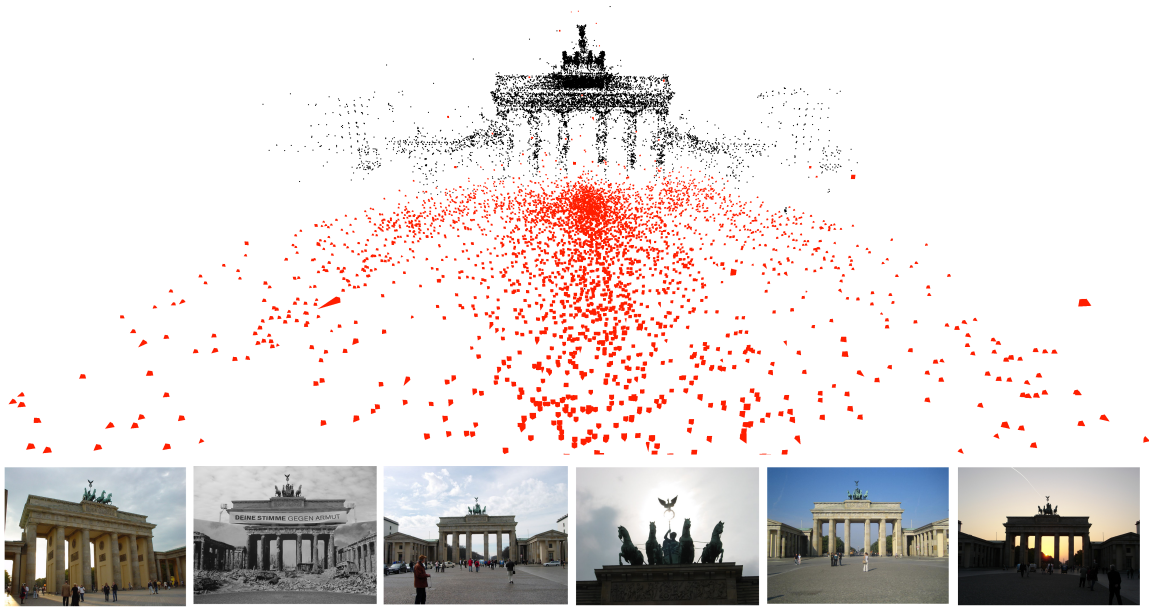


Figure 6.14. Autocalibrated reconstruction from a collection of 5,514 web photos (taken by different cameras) of *Brandenburg Gate* in Berlin, Germany. The reconstruction by Frahm et al. [2010] consists of 19,963 structure points (black dots), and cameras are shown as red pyramids. Some representative images used in the reconstruction are shown along the bottom.

In our first example, we used a reconstruction of *Brandenburg Gate* in Berlin, Germany that has been reconstructed by Frahm et al. [2010] from a collection of 5,514 photos gathered from the internet. The reconstruction consists of 19,963 structure points and has a total of 1,006,741 observations.

We obtained a ML projective reconstruction from the original measurements by using bundle adjustment and then autocalibrated using our ML+R method assuming zero skew and aspect ratio, to yield a metric reconstruction with mean squared reprojection error of 0.79 pixels (relative to 1000×1000 pixel images). A view of the reconstruction after our autocalibration is shown in Fig. 6.14, and does not appear to suffer from any overall distortion, indicating that autocalibration was successful.

The next example is a reconstruction of the *Piazza dei Signore* in Verona, Italy that was reconstructed by the SAMANTHA [Farenzena et al., 2009] pipeline from a collection of 1144×856 photos taken by the same physical camera. The reconstruction consists of 2971 structure points, 39 views, and had an initial mean squared reprojection error of 0.330401 pixels which was reduced to 0.253106 after our projective bundle adjustment. We autocalibrated using our ML method, assuming a search range for focal length in the range of 1-3 screen widths, and initially assumed a principal point at $(572, 428)$ (the center of the image) for the dual-stratified search. After the nonlinear improvement, the principal point was corrected to $(554.836, 452.762)$. After

Figure 6.15. Autocalibrated reconstruction from a collection of photos (taken by the same camera) of the *Piazza dei Signore* in Verona, Italy. The reconstruction is shown from a top down orthographic perspective. The reconstruction by Farenzena et al. [2009] consists of 39 views and a total of 2971 structure points. Some images from representative views are shown along the bottom (the aerial view was not used in the reconstruction and is presented only for reference).

factoring out the similarity transformation, the mean squared difference between the structure points in our metric rectified reconstruction and the original was just $2.734\,26 \times 10^{-5}$, which agrees quite closely with the original. An orthographic view of the reconstructed point cloud after our autocalibration is shown from a top down perspective in Fig. 6.15, where it can be verified from the density of points on the vertical walls that they are parallel.

Finally, we demonstrate autocalibration of a video reconstruction that was made by [Clipp et al., 2010] using a parallel real-time visual SLAM method. This reconstruction consists of a total of 23 keyframes that were selected out of a 300 frame video with a resolution of 1224×1024. There are a total of 1,473 structure points and 17,077 observations. The mean squared reprojection error of the reconstruction was 4.78 pixels, which we reduced to 0.529021 pixels using our projective bundle adjustment. Then we autocalibrated using our ML+R method assuming constant focal length, constant principal point, zero skew, and zero aspect ratio, which raised the mean square reprojection error only slightly to 0.530253 pixels. A view of the reconstruction after our autocalibration is shown in Fig. 6.16.

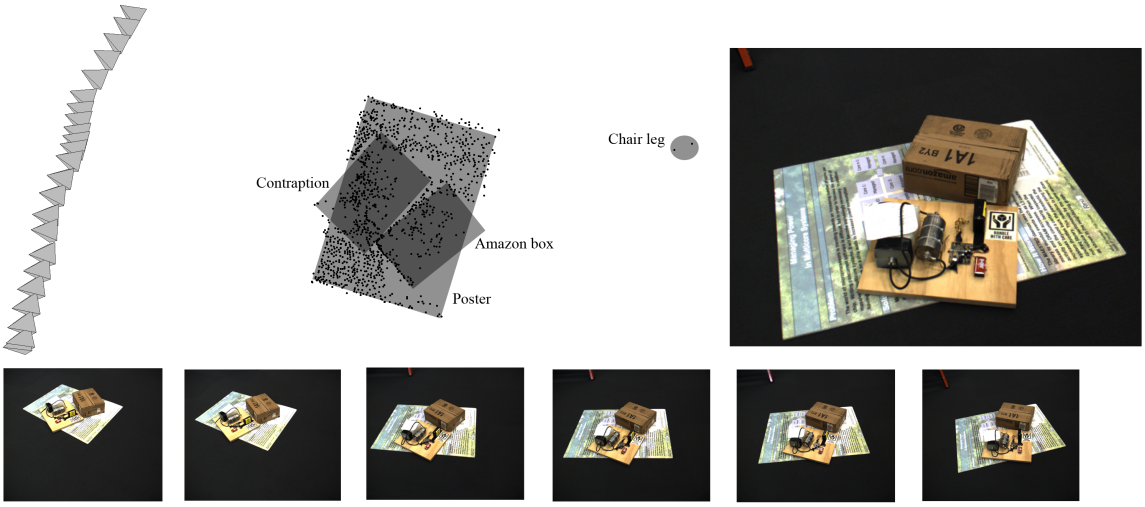


Figure 6.16. A view of the point cloud of the autocalibrated reconstruction from a video reconstruction with 23 views, 1,473 structure points and 17,077 observations by Clipp et al. [2010]. Some representative views are shown along the bottom. The approximate location of the scene elements was determined based on point elevations and is pictured here using opacity mapped squares for reference.

6.8 Conclusions

It has been thought that likelihood cannot be exploited during autocalibration because multiplication by any homography does not change reprojection error, and all previous autocalibration algorithms have instead opted to minimize various heuristics based on manipulating the algebraic constraints that arise based on the assumption of zero skew, unit aspect ratio, or constant principal point/focal length. However, these heuristic cost functions are unstable because even the ideal ML projective reconstruction is not exactly within a homography from a true metric reconstruction.

We have shown that by taking into account metric constraints, a likelihood can be associated with any homography, thereby allowing one to seek the maximum likelihood rectifying homography. The ML homography can be found reliably by using a dual-stratified initialization followed by nonlinear improvement, and this method is more robust and accurate than any of the other algorithms we have tested for both small and large problems.

The advantages of maximizing likelihood as opposed to minimizing some arbitrary heuristic are many. First, the solution is invariant to the initial projective ambiguity; unlike all previous approaches, the projective reconstruction can be multiplied by any homography without changing the ML rectified result. Second, it is very robust to noise because it minimizes reprojection errors, which are geometrically meaningful. Third, because the error is geometrically meaningful, it is simple to incorporate early termination for improved performance whenever

a reasonable error tolerance has been met. Fourth, likelihood can be used as an objective way to compare the performance of various autocalibration algorithms when a ground truth reconstruction is not available. And finally, it does not require any configuration-dependent weighting coefficients.

Chapter 7

Bundle Adjustment

Consider a set of n homogeneous structure points $\bar{\mathbf{X}}_i, i = 1 \dots n$ in the projective space $\mathbb{P}^3$, viewed by a set of m cameras having 3×4 projection matrices $\bar{\mathbf{P}}^j = \bar{\mathbf{K}}^j[\bar{\mathbf{R}}^j|\bar{\mathbf{t}}^j]$, $j = 1 \dots m$, where $\bar{\mathbf{R}}^j$ is a rotation matrix and $\bar{\mathbf{K}}^j$ is a non-singular upper triangular calibration matrix with positive diagonal elements. We refer to the combined set of this information as the *true configuration*, denoted by

$$\bar{\Theta} = \{\, \bar{\mathbf{X}}_i, \bar{\mathbf{P}}^j | \forall i, j \,\}, \tag{7.1}$$

and any estimate $\widehat{\Theta}$ of the configuration from some measurements as a *reconstruction* of the configuration.

The perspective projection of a homogeneous structure point $\mathbf{X} \in \mathbb{P}^3$ as viewed by a camera with projection matrix $\mathbf{P}$ is accomplished by multiplication, yielding a homogeneous image point $\mathbf{x} \in \mathbb{P}^2$,

$$\mathbf{x} \propto \mathbf{P}\mathbf{X}. \tag{7.2}$$

Let the measured coordinates of the image of the ith structure point in the jth image be denoted by $\tilde{\mathbf{x}}_i^j$. If we assume, as is commonly done [Hartley and Sturm, 1997], that measurement error is normally distributed with standard deviation σ, then the probability (or likelihood) of a measurement is

$$P(\tilde{\mathbf{x}}_i^j|\bar{\Theta}) = \frac{1}{2\pi\sigma^2} \exp\left(-d(\tilde{\mathbf{x}}_i^j, \bar{\mathbf{x}}_i^j)^2/(2\sigma^2)\right), \tag{7.3}$$

where $\bar{\mathbf{x}}_i^j$ is the true image of $\bar{\mathbf{X}}_i$ in the jth view, and $d(\mathbf{a}, \mathbf{b})$ is the Euclidean distance between the inhomogeneous points represented by homogeneous points $\mathbf{a}$ and $\mathbf{b}$. The log-probability of

a measurement is

$$\log P(\tilde{\mathbf{x}}_i^j|\bar{\Theta}) = -\frac{1}{2\sigma^2} d(\tilde{\mathbf{x}}_i^j, \bar{\mathbf{x}}_i^j)^2 + \underbrace{\log\left(1/(2\pi\sigma^2)\right)}_{constant}, \tag{7.4}$$

and therefore the maximum likelihood (ML) projective reconstruction $\widehat{\Theta}_{ML}$ from measurements $\{\tilde{\mathbf{x}}_i^j\}$ is given by

$$\widehat{\Theta}_{ML} = \underset{\Theta}{\operatorname{argmax}} \prod_{i,j} P(\tilde{\mathbf{x}}_i^j|\Theta) \tag{7.5}$$

$$= \underset{\Theta}{\operatorname{argmax}} -\frac{1}{2\sigma^2} \sum_{i,j} d(\tilde{\mathbf{x}}_i^j, \mathbf{x}_i^j)^2 \tag{7.6}$$

$$= \underset{\Theta}{\operatorname{argmin}} \sum_{i,j} d(\tilde{\mathbf{x}}_i^j, \mathbf{x}_i^j)^2. \tag{7.7}$$

The distance $d(\tilde{\mathbf{x}}_i^j, \mathbf{x}_i^j)$ is known as *reprojection error*, so the ML reconstruction is the one that minimizes the sum of squared reprojection errors. This nonlinear minimization is known as *bundle adjustment* [Triggs et al., 2000; Hartley and Zisserman, 2004; Lourakis and Argyros, 2004].

7.1 Parameterization

Bundle adjustment is a very generic term that refers to any method of nonlinearly minimizing reprojection errors. Generally the reconstruction is parameterized by a set of 3-dimensional points and a set of cameras, but depending on the way that cameras are parameterized the method can have very different uses. The most fundamental dichotomy is between projective and metric parameterizations.

7.1.1 Projective

In projective bundle adjustment, points are parameterized as homogeneous 4-vectors $\mathbf{X} \in \mathbb{P}^3$, and each camera view is parameterized by an unconstrained 3×4 projection matrix. In this case the partial derivatives are simple to evaluate analytically and the linearized approximation is fairly good, leading to an efficient algorithm with a wide basin of attraction.

However, one usually has knowledge of many other constraints which are not enforced by the projection matrices, such as known aspect ratio, lack of image skew, known principal point, or constant intrinsic parameters between views that are taken by the same camera. These may be regarded as metric constraints.

Multiplying a projective reconstruction by a homography does not change reprojection error, and therefore the result of projective bundle adjustment will also be ambiguous up to multiplication by an arbitrary homography. Autocalibration is an attempt to resolve this homography in a second phase by using the previously neglected metric constraints. However, an autocalibrated solution will still never be able to satisfy the metric constraints exactly.

7.1.2 Metric

In metric bundle adjustment, points are parameterized by inhomogeneous 3-vectors $\mathbf{X} \in \mathbb{R}^3$, and cameras are parameterized by their pose (a rigid motion) as well as some set of calibration matrices.

Calibration matrices are parameterized only by their unknown elements; generally, aspect ratio, skew and possibly principal point are known, so that focal length is the only unknown element of a calibration matrix. Views that are known to be taken by the same camera should share the same calibration parameters in the parameterization.

The rotational aspect of camera pose can be parameterized minimally using a 3-vector for axis-angle using Rodrigues' rotation formula [Hartley and Zisserman, 2004, p. 585], or as a 4-vector quaternion. The advantage of using the axis-angle parameterization is that it leads to a slightly smaller system of equations that can be solved more quickly. However, the partials are more complex to calculate analytically, and there is a singularity at $(0, 0, 0)$ that can lead to division by zero or cause instability when a rotation is near this singularity. Using quaternions there is no singularity, and the partials come out simpler and have more redundant terms allowing them to be analytically computed more easily.

7.2 Generic Nonlinear Minimization

Having decided upon a parameterization, minimization can be accomplished as a generic model fitting problem using traditional nonlinear least squares. The measured image points (correspondences) define the observation vector $\mathbf{Y}$, and the transformation function $f(\mathbf{X})$ is simply perspective projection. The objective is to find the parameter vector $\mathbf{X}$ such that $f(\mathbf{X}) \approx \mathbf{Y}$.

7.2.1 Gradient Descent

Defining the residual as $\epsilon = f(\mathbf{X}) - \mathbf{Y}$, then there is always a unique solution that minimizes $S(\mathbf{X}) = ||\epsilon||^2$, the sum of squared errors. The gradient of $S(\mathbf{X})$ is given by

$$\frac{\partial}{\partial \mathbf{X}} S(\mathbf{X}) = \frac{\partial}{\partial \mathbf{X}} \left((f(\mathbf{X}) - \mathbf{Y})^{\mathsf{T}} (f(\mathbf{X}) - \mathbf{Y}) \right) \tag{7.8}$$

$$= 2 \left(\frac{\partial f(\mathbf{X})}{\partial \mathbf{X}} \right)^{\mathsf{T}} (f(\mathbf{X}) - \mathbf{Y}) \tag{7.9}$$

$$= 2\mathbf{J}^{\mathsf{T}} \epsilon, \tag{7.10}$$

where $\mathbf{J} = \frac{\partial f(\mathbf{X})}{\partial \mathbf{X}}$ is the Jacobian of f. Taking a sufficiently small step in the direction of the gradient will therefore reduce the error, suggesting an update rule of

$$\mathbf{X} \leftarrow \mathbf{X} - \lambda \mathbf{J}^{\mathsf{T}} \epsilon, \tag{7.11}$$

for some sufficiently small step size λ. Applying this update equation, with some constant value or automatic method (e.g., line search) for choosing λ, is known as the method of *gradient descent*.

7.2.2 Gauss-Newton Method

An alternative to gradient descent is to take the step that would take a linearized approximation of the error to zero. The first-order approximation of f at $\mathbf{P} + \delta$ is given by

$$f(\mathbf{P} + \delta) \approx f(\mathbf{X}) + \mathbf{J}\delta, \tag{7.12}$$

and substituting this into the objective function $S(\mathbf{X})$ yields a first order approximation of the objective at a slightly offset point,

$$S(\mathbf{P} + \delta) \approx ||f(\mathbf{X}) + \mathbf{J}\delta - \mathbf{Y}||^2. \tag{7.13}$$

At the minimum of $S(\mathbf{X})$ the gradient will be zero. Therefore, by taking the derivative of (7.13) and setting it to zero, one obtains a set of equations that can be used to solve for the update δ. Ignoring second-order terms, this gives

$$0 = \frac{\partial}{\partial \mathbf{X}} \left(f(\mathbf{X}) + \cancel{\mathbf{J}\delta} - \mathbf{Y}\right)^\mathsf{T} \left(f(\mathbf{X}) + \mathbf{J}\delta - \mathbf{Y}\right) \tag{7.14}$$

$$= 2\mathbf{J}^\mathsf{T} \left(f(\mathbf{X}) + \mathbf{J}\delta - \mathbf{Y}\right) \tag{7.15}$$

$$= 2\mathbf{J}^\mathsf{T} \left(\mathbf{J}\delta + \epsilon\right) \tag{7.16}$$

$$\mathbf{J}^\mathsf{T}\mathbf{J}\delta = -\mathbf{J}^\mathsf{T}\epsilon \tag{7.17}$$

These are known as the *normal equations*, and can be solved using any method of linear least squares (SVD, Cholesky decomposition, QR factorization, LU decomposition, matrix inverse, etc).

Iterative application of (7.17) is called the Gauss-Newton method [Nocedal and Wright, 1999], and results in very fast convergence when $S(\mathbf{X})$ is well approximated by a first or second order function. However, the update is not guaranteed to result in improvement, and the method can easily diverge or become stuck in a local minima when the first order approximation is particularly poor.

7.2.3 Levenberg-Marquardt

Levenberg [Levenberg, 1944] modified the normal equations by adding a damping term $\lambda \mathtt{I}$,

$$\left(\mathbf{J}^\mathsf{T}\mathbf{J} + \lambda \mathtt{I}\right) \delta = -\mathbf{J}^\mathsf{T}\epsilon. \tag{7.18}$$

When λ is very small then (7.18) becomes identical to (7.17). On the other hand when λ is very large, the system becomes approximated by $\lambda\delta = -\mathbf{J}^\mathsf{T}\epsilon$ and the update rule becomes

$$\mathbf{X} \leftarrow \mathbf{X} - \frac{1}{\lambda}\mathbf{J}^\mathsf{T}\epsilon, \tag{7.19}$$

which is equivalent to (7.11) using $1/\lambda$.

Rather than doing a line search for λ, it can be determined adaptively. In typical implementations λ is initialized to be 10^{-3} times the average diagonal element of $\mathbf{J}^\mathsf{T}\mathbf{J}$. On each update that leads to a reduction of the error, λ is divided by 10 before the next iteration. Otherwise the update is rejected and λ is multiplied by 10 [Hartley and Zisserman, 2004]. Thus, each iteration of (7.18) moves seamlessly between Gauss-Newton, which has rapid convergence when the linearized approximation works very well, and gradient descent, which guarantees moving in the right direction when the Gauss-Newton method fails.

Marquardt [Marquardt, 1963] provided the insight that each component of the gradient can

be scaled according to the curvature so that there is larger movement along directions where the gradient is smaller, in order to avoid slow convergence in the direction of small gradient. Therefore, Marquardt replaced the identity damping matrix from (7.18) with the diagonal of $\mathbf{J}^\mathsf{T}\mathbf{J}$,

$$\left(\mathbf{J}^\mathsf{T}\mathbf{J} + \lambda \operatorname{diag} \mathbf{J}^\mathsf{T}\mathbf{J}\right) \delta = -\mathbf{J}^\mathsf{T}\epsilon. \tag{7.20}$$

Iterative application of (7.20) is known as the LevenbergMarquardt (LM) algorithm, and is the preferred method for minimizing most nonlinear optimization problems in this work. See Algorithm 3 for pseudo-code.

Algorithm 3 Generic Levenberg-Marquardt

Require: A solution $\mathbf{X}$ having error ϵ_{start}.

Ensure: An improved $\mathbf{X}$ having $\epsilon \leq \epsilon_{start}$

$badSteps \leftarrow 0$
$\epsilon \leftarrow \epsilon_{start}$
$\lambda \leftarrow 0.01$
loop
  $\mathbf{L} \leftarrow \mathbf{J}^\top\mathbf{J}$
  $\mathbf{R} \leftarrow -\mathbf{J}^\top\mathbf{E}$
  $\epsilon_{test} \leftarrow \infty$
  $badSteps \leftarrow 0$
  repeat
    $\mathbf{L}_{test} \leftarrow \mathbf{L} + \lambda \operatorname{diag} \mathbf{L}$
    Solve $L_{test}\delta = \mathbf{R}$ for δ
    $\mathbf{X}_{test} \leftarrow \mathbf{X} + \delta$
    $\epsilon_{test} \leftarrow \text{evaluate}(\mathbf{X}_{test})$
    if $\epsilon_{test} \geq \epsilon$ **then**
      $badSteps \leftarrow badSteps + 1$
      if $badSteps > maxBadSteps$ **then**
        return ϵ
      end if
      $\lambda \leftarrow 10\lambda$
    end if
  until $\epsilon_{test} < \epsilon$
  $\mathbf{X} \leftarrow \mathbf{X}_{test}$
  $\epsilon \leftarrow \epsilon_{test}$
  $\lambda \leftarrow 0.3\lambda$
end loop

7.3 Performance Optimizations

Straight-forward application of (7.20) is computationally impractical for most bundle adjustment problems due to the large number of parameters involved. In the projective bundle adjustment problem, for a reconstruction with m views and n points, the Jacobian $\mathbf{J}$ is $2mn \times 3n + 12(m-1)$.

For example, a modest reconstruction of 25 views and 800 points per view would therefore

have 60,288 parameters, and this would require solving a $60,288 \times 60,288$ linear least squares system for *each* nonlinear update. Solving an $n \times n$ linear least squares problem is an $O(n^3)$ problem, and it could easily take hundreds of updates to converge.

For systems of this size, even forming the linear equations, let alone solving them, can be computationally impractical when done naively. Thus, in order to make the problem tractable it is imperative to consider sparsity whenever possible.

7.3.1 Sparsity due to Projection Independence

Because the observation vector consists of measured image points in each view, the projections into one view are largely independent from the projections into another view. In the case of projective bundle adjustment, projection matrices are treated as being completely independent. In the metric case, there may be a shared focal length, but otherwise camera pose is still independent. This results in a sparse structure of the normal equations which can be naturally partitioned as

$$\begin{bmatrix} \mathbf{U} & \mathbf{W} \\ \mathbf{W}^\mathsf{T} & \mathbf{V} \end{bmatrix} \begin{bmatrix} \delta^P \\ \delta^S \end{bmatrix} = \begin{bmatrix} \epsilon^P \\ \epsilon^S \end{bmatrix}, \tag{7.21}$$

where δ^P is the update vector for all the camera parameters and δ^S is the update vector for all the structure points.

The number of structure points n is typically much larger than the number of cameras m. For example, if there are 600 new feature points spotted in each image, then $n \approx 600m$. The number of parameters per camera ranges from 6 to 12 depending on the parameterization. Thus, for $m = 100$, we could expect $\mathbf{U}$ is roughly 1000×1000 and $\mathbf{V}$ is roughly 180000×180000.

Linear least squares methods require $O(n^3)$ flops to solve an $n \times n$ system [Golub and Van Loan, 1996a]; thus, naively solving (7.21) directly would require about 7×10^{15} flops per iteration (for $m = 100$). However, reasonable performance can be achieved by taking advantage of the inherent sparsity. For example, after multiplying both sides of (7.21) by

$$\begin{bmatrix} \mathtt{I} & -\mathbf{W}\mathbf{V}^{-1} \\ 0 & \mathtt{I} \end{bmatrix}, \tag{7.22}$$

one obtains a much smaller system to solve for δ^P,

$$(\mathbf{U} - \mathbf{W}\mathbf{V}^{-1}\mathbf{W}^\mathsf{T})\delta^P = \epsilon^P - \mathbf{W}\mathbf{V}^{-1}\epsilon^S. \tag{7.23}$$

In any bundle adjustment problem, $\mathbf{V}$ is symmetric positive-semidefinite and block diagonal. Thus its inverse can be easily computed as a series of symmetric 3×3 inverses.

Once (7.23) has been solved, the other half of the solution is given by back substitution, which simply requires multiplication using the already computed $\mathbf{V}^{-1}$,

$$\mathbf{W}^{\mathsf{T}}\delta^P + \mathbf{V}\delta^S = \epsilon^S \tag{7.24}$$

$$\delta^S = \mathbf{V}^{-1}(\epsilon^S - \mathbf{W}^{\mathsf{T}}\delta^P). \tag{7.25}$$

This decomposition is frequently used in modern implementations [Hartley and Zisserman, 2004; Engels et al., 2006].

7.3.2 Sparsity due to Feature Visibility

The matrix $\mathbf{U}$ is also symmetric positive-semidefinite, and if the parameters of each camera are independent, then it is also block-diagonal. If the feature tracks are sparse then $\mathbf{W}$ is mostly sparse with vertical banding wherever a structure point is visible in a particular view. When $\mathbf{U}$ is block diagonal and $\mathbf{W}$ is banded, then $\mathbf{U} - \mathbf{W}\mathbf{V}^{-1}\mathbf{W}^{\mathsf{T}}$ will have a sparse skyline structure. In this case, the most efficient way to solve (7.23) is to use a custom $\mathbf{LDL}^{\mathsf{T}}$ decomposition for skyline matrices [Golub and Van Loan, 1996a], because the skyline sparsity will be preserved in $\mathbf{L}$, thereby keeping the time complexity proportional to the number of non-zero elements rather than the overall size of the matrix. After decomposing as $\mathbf{LDL}^{\mathsf{T}}$, a sparse-aware function can be written for forward and back-substitution necessary to finish solving the system. Further details are given in [Bathe and Wilson, 1976].

In practice, $\mathbf{W}$ will always be sparse for large problems. The only practical scenario in which $\mathbf{W}$ would be fully dense is when bundle adjustment is used within a RANSAC [Fischler and Bolles, 1981] framework to improve a linear estimate of the fundamental matrix or trifocal tensor. However, as these problems only have two or three views, the system can still be solved efficiently without taking advantage of the sparsity.

7.3.3 Alternation

The presence of any camera parameters that are shared between views will eliminate the sparse structure in $\mathbf{L}$ when doing the $\mathbf{LDL}^{\mathsf{T}}$ decomposition. There will still be sparsity in $\mathbf{U} - \mathbf{W}\mathbf{V}^{-1}\mathbf{W}^{\mathsf{T}}$ that could be exploited to reduce the iteration over zero entries in the input matrix, but the output will be fully dense.

In the projective case this could occur if one wanted to use a radial distortion coefficient. In the metric case it would occur if any the views were taken by the same camera (and therefore

share calibration parameters). In these situations, the most efficient way to solve (7.23) as a dense system would be Cholesky decomposition [Golub and Van Loan, 1996a]. There is a parallel Cholesky decomposition [Khazal and M.M.Chawla, 2004] that may be useful for larger problems, but the overall time complexity quickly grows prohibitive.

Therefore, when there are globally shared camera parameters, we suggest using the method of alternation; one would alternate by first solving for the global parameters while holding all structure and remaining camera parameters fixed, and then switch and hold the global parameters fixed while solving for the structure and remaining camera parameters. This will restore the sparse structure allowing any remaining camera parameters (e.g., camera pose) to be treated independently.

7.3.4 Parallel Computation

Due to the large size of matrices, the matrix multiplications necessary to construct (7.23) and (7.25) also become computationally prohibitive if treated naively. In this section, we describe a fully data parallel method for constructing and solving the linear system in five parallel bursts.

Taking advantage of the internal sparsity, we first write (7.23) and (7.25) using non-zero sub blocks as

$$(\mathbf{U} - \sum_{i=1}^{n} \mathbf{W}_i \mathbf{V}_i^{-1} \mathbf{W}_i^{\mathsf{T}}) \delta^P = \epsilon^P - \sum_{i=1}^{n} \mathbf{W}_i \mathbf{V}_i^{-1} \epsilon_i^S \tag{7.26}$$

$$\delta_i^S = \mathbf{V}_i^{-1} \left(\epsilon_i^S - \mathbf{W}_i{}^{\mathsf{T}} \delta^P \right) \tag{7.27}$$

These sub-blocks are related to the larger blocks by

$$\mathbf{W} = [\mathbf{W}_1 | \dots | \mathbf{W}_n] \tag{7.28}$$

$$\mathbf{W}_i = [\mathbf{W}_i^{1\mathsf{T}} | \dots | \mathbf{W}_i^{m\mathsf{T}}]^{\mathsf{T}} \tag{7.29}$$

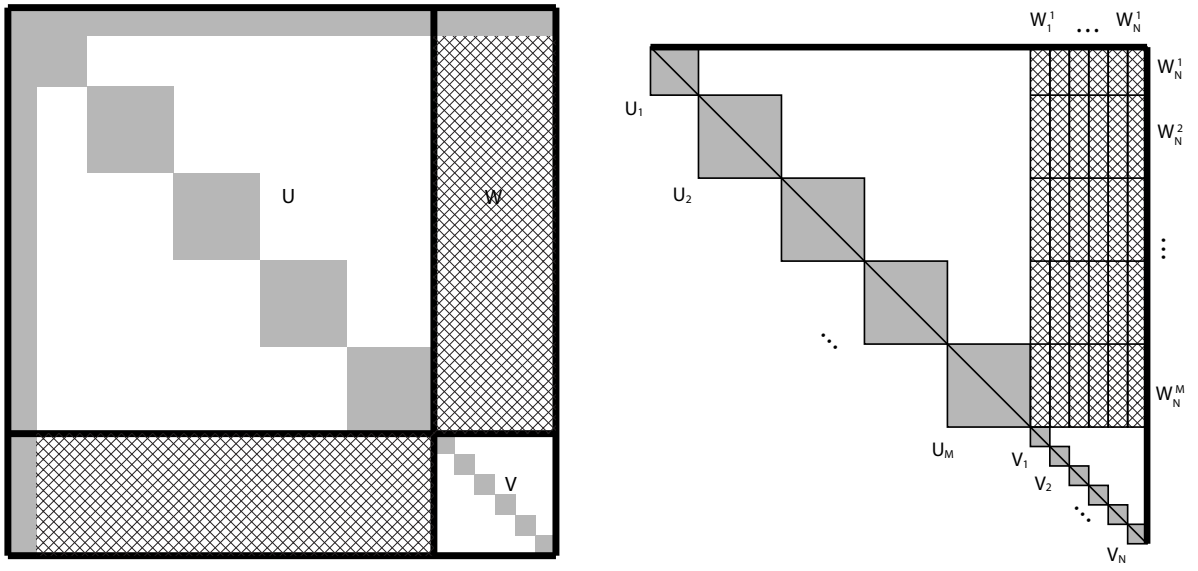
$$\epsilon^P = [\epsilon_1^{P\mathsf{T}} | \dots | \epsilon_m^{P\mathsf{T}}]^{\mathsf{T}} \tag{7.30}$$

$$\epsilon^S = [\epsilon_1^{S\mathsf{T}} | \dots | \epsilon_n^{S\mathsf{T}}]^{\mathsf{T}} \tag{7.31}$$

$$\delta^P = [\delta_1^{P\mathsf{T}} | \dots | \delta_m^{P\mathsf{T}}]^{\mathsf{T}} \tag{7.32}$$

$$\mathbf{V} = \begin{bmatrix} \mathbf{V}_1 & & \\ & \ddots & \\ & & \mathbf{V}_n \end{bmatrix}, \tag{7.33}$$

as shown in Fig. 7.1.



(a) (b)

Figure 7.1. Sparse structure of $\mathbf{JTJ}^{\mathsf{T}}$. (a) primary blocks of $\mathbf{U}$, $\mathbf{V}$, $\mathbf{W}$ on a generic bundle adjustment problem having some global parameters; (b) sub-blocks on a bundle adjustment problem with no global parameters. Non-zero blocks are colored in gray, and blocks that may be zero depending on visibility are cross-hatched.

Assuming that there are no global parameters (since these can be removed by alternation when they exist), $\mathbf{U}$ can be broken down as

$$\mathbf{U} = \begin{bmatrix} \mathbf{U}_1 & & \\ & \ddots & \\ & & \mathbf{U}_m \end{bmatrix}. \tag{7.34}$$

The blocks in (7.28-7.34) can be constructed directly in terms of blocks in the Jacobian and residual vector. Let $\mathbf{P}_i^j$ be the $2 \times CamParams$ matrix of partial derivatives of the projection of the ith structure point into the jth image w.r.t each free parameter of the jth camera. Let $\mathbf{S}_i^j$ be the 2×3 matrix of partial derivatives of the projection of the ith structure point in the jth image w.r.t the coordinates of the ith structure point. Let $\mathbf{C}_j$ be the $CamPriors \times CamParams$ matrix of partial derivatives of the prior constraints on the jth camera w.r.t the free parameters of the jth camera. Let ϵ_i^j be the 2×1 matrix of residual reprojection errors of the ith structure point in the jth image and ϵ_j^C be the $CamPriors \times 1$ matrix of residual prior errors of the jth camera. Then, we can write the non-zero blocks of (7.26) in terms of the non-zero blocks of the Jacobian as

$$\mathbf{U}_j = \mathbf{C}_j^\mathsf{T}\mathbf{C}_j + \sum_{i=1}^{n} \mathbf{P}_i^{j\mathsf{T}}\mathbf{P}_i^j \tag{7.35}$$

$$\mathbf{V}_i = \sum_{j=1}^{m} \mathbf{S}_i^{j\mathsf{T}}\mathbf{S}_i^j \tag{7.36}$$

$$\mathbf{W}_i^j = \mathbf{P}_i^{j\mathsf{T}}\mathbf{S}_i^j \tag{7.37}$$

$$\epsilon_j^P = \mathbf{C}_j^\mathsf{T}\epsilon_j^C + \sum_{i=1}^{n} \mathbf{P}_i^{j\mathsf{T}}\epsilon_i^j \tag{7.38}$$

$$\epsilon_i^S = \sum_{j=1}^{m} \mathbf{S}_i^{j\mathsf{T}}\epsilon_i^j \tag{7.39}$$

Note that we have omitted the LM damping factors here for notational simplicity, but they are implemented simply by multiplying the diagonal elements of $\mathbf{U}_j$ and $\mathbf{V}_i$ by the LM scaling factor.

The projection function may be simple perspective projection (e.g., according to the pinhole projection model) or may also take into account some model for lens distortion, such as radial distortion coefficients [Devernay and Faugeras, 1995; Wang et al., 2009], which may be treated as additional intrinsic camera parameters. When dealing with a video, the radial distortion can be estimated independently and then a more efficient option is to simply correct for radial distortion from the image points in the feature tracks so that pinhole projection is used in all subsequence calculations (such as bundle adjustment).

We compute each δ_i^S in parallel during the back-substitution phase of (7.27). We also construct the system in (7.26) as a series of independent matrix slices which we compute in parallel. Specifically, if we define

$$\mathbf{L}_i = \begin{bmatrix} \mathbf{P}_i^{1\mathsf{T}}\mathbf{P}_i^1 & & \\ & \ddots & \\ & & \mathbf{P}_i^{m\mathsf{T}}\mathbf{P}_i^m \end{bmatrix} - \mathbf{W}_i\mathbf{V}_i^{-1}\mathbf{W}_i^\mathsf{T} \tag{7.40}$$

$$\mathbf{R}_i = \begin{bmatrix} \mathbf{P}_i^{1\mathsf{T}}\epsilon_i^1 \\ \vdots \\ \mathbf{P}_i^{m\mathsf{T}}\epsilon_i^m \end{bmatrix} - \mathbf{W}_i\mathbf{V}_i^{-1}\epsilon_i^S, \tag{7.41}$$

then the slices $\mathbf{L}_i$ and $\mathbf{R}_i$ can be computed independently for each i. The multiplication $\mathbf{V}_i^{-1}\mathbf{W}_i$ can also be reused. Then (7.26) can be written as

$$\left(\begin{bmatrix} \mathbf{C}_1^\mathsf{T}\mathbf{C}_1 & & \\ & \ddots & \\ & & \mathbf{C}_m^\mathsf{T}\mathbf{C}_m \end{bmatrix} + \sum_{i=1}^{n} \mathbf{L}_i \right) \delta^P = \begin{bmatrix} \mathbf{C}_1^\mathsf{T}\epsilon_1^C \\ \vdots \\ \mathbf{C}_m^\mathsf{T}\epsilon_m^C \end{bmatrix} + \sum_{i=1}^{n} \mathbf{R}_i \tag{7.42}$$

Thus, we can construct and solve the normal equations in five parallel bursts:

1. Parallel loop over $i = 1 \dots n$ to compute the independent slices $\mathbf{L}_i$ and $\mathbf{R}_i$.
2. Parallel summation over $i = 1 \dots n$ to accumulate $\sum_{i=1}^{n} \mathbf{L}_i$ and $\sum_{i=1}^{n} \mathbf{R}_i$ (i.e., each thread computes a partial sum, and the partial sums are finally added in series).
3. Parallel loop over $j = 1 \dots m$ to add the effect of the priors for each camera to left and right hand sides.
4. Solve (7.42) for δ^P. This can be done, for example, using a custom $\mathbf{LDL}^\mathsf{T}$ routine for skyline matrices if $\mathbf{W}$ is sparse, or a parallel Cholesky decomposition if it is dense.
5. Parallel loop over $i = 1 \dots n$ to compute each δ_i^S in (7.27).

We show a performance comparison of our parallel decomposition vs traditional single-threaded bundle adjustment in Fig. 7.2. These tests were run on a Core i7-920 which has 4 hyper-threaded processors. It should be noted that in these tests we have solved (7.42) using SVD which is not parallelized. Even so, using four threads we achieved roughly 3x speedup, and using 8 threads (to take advantage of hyper-threading) we achieved roughly 4x speedup. The ability to achieve 4x speedup on a 4 processor machine shows that our parallel decomposition is quite effective.

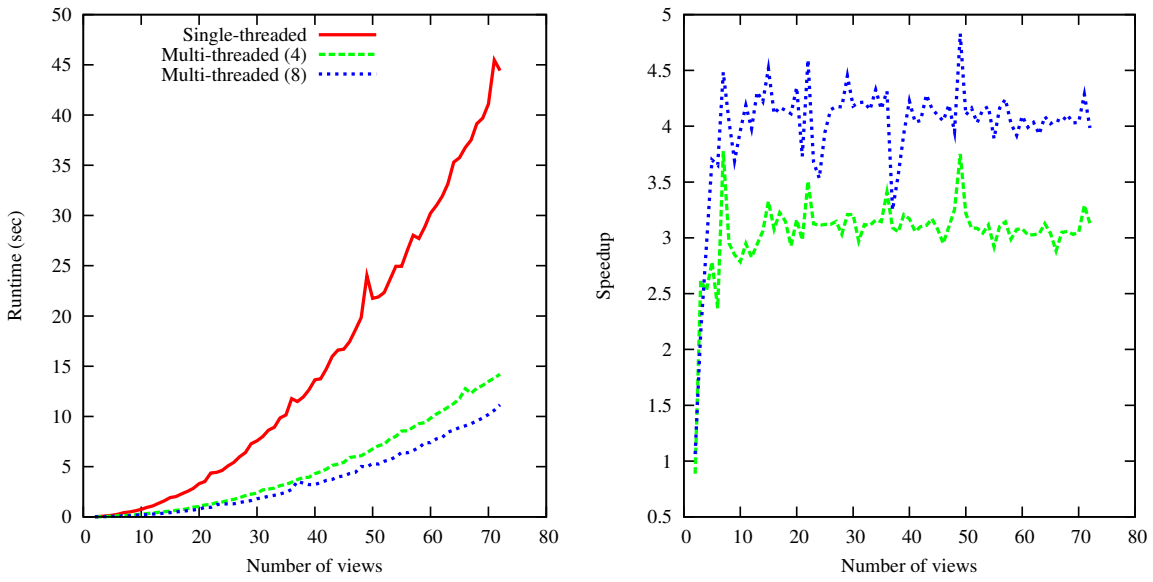


Figure 7.2. Performance comparison of our parallel bundle adjustment to conventional single-threaded bundle adjustment. Both versions were run on a Core i7-920 which has 4 hyper-threaded processors.

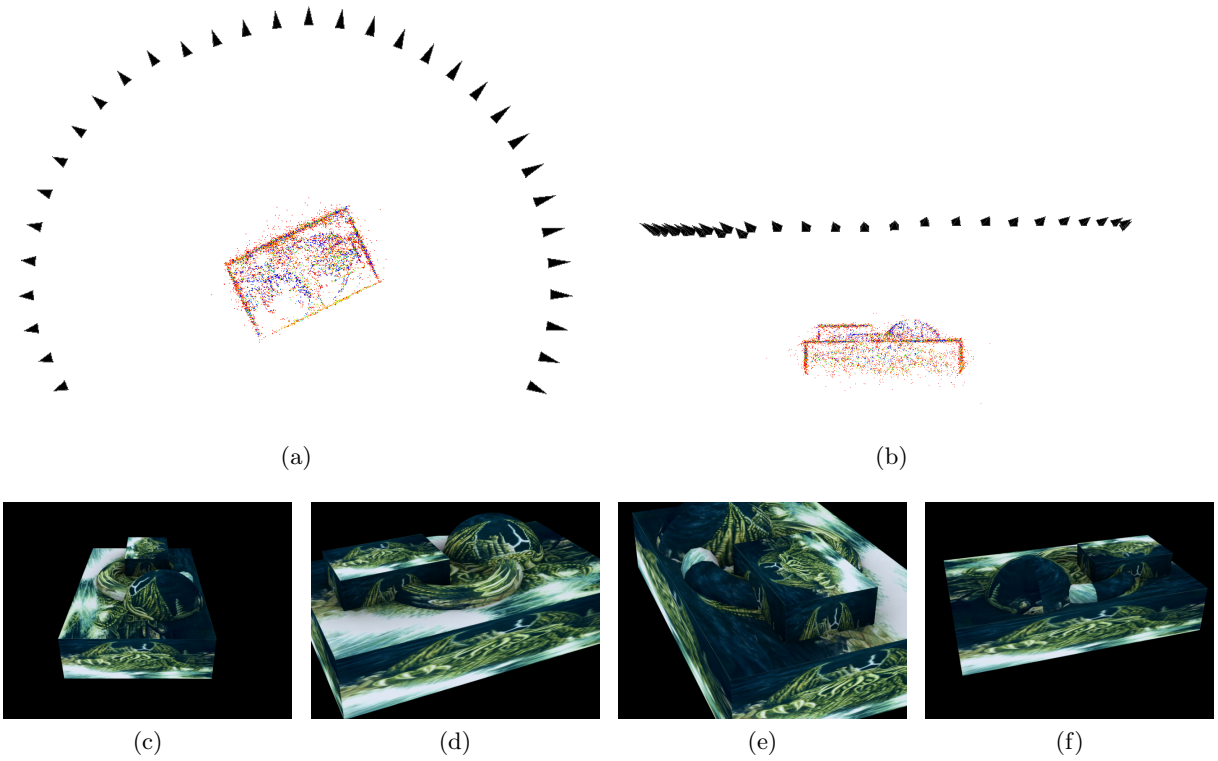
7.4 Example Sparse Reconstructions

An example reconstruction computed from 33 views using the proposed methods is shown in Fig. 7.3. Note that although the views were rendered synthetically from a 3D model, correspondence measurements were still identified automatically using the methods of Chapter 3 as if it were a regular video from any other source.

The reconstructed points are color-coded according to the number of views they were triangulated from; red points were seen in just two views (and hence are less accurate), yellow points were seen in three views, green points in four views, and blue points in five or more views.

As can be seen from the reconstruction, the camera arcs around a central object. At the start of the video the field of view is 45°, which is reduced to 25° at maximum zoom, and then begins to zoom back out again slightly. The reconstructed focal length can be seen visually from the size of the triangles at each view position. A graph of the recovered field of view (after the nonlinear transformation from focal length) is shown in Fig. 7.4.

Some example reconstructions from real data are shown in Fig. 7.5 and Fig. 7.6. Again, the reconstructed structure points are color-coded according to the number of views they were visible in.



(a) (b)

(c) (d) (e) (f)

Figure 7.3. Sparse reconstruction of *WeirdZoom*. The camera rotates around a central object while zooming in and out. Initially the field of view is 45°, which is reduced to 25° at maximum zoom, and then begins to zoom back out again slightly. The reconstructed focal length can be seen visually from the size of the triangles at each view position. (a) top view; (b) side view; (c) frame 0; (d) frame 16; (e) frame 28; (f) frame 40.

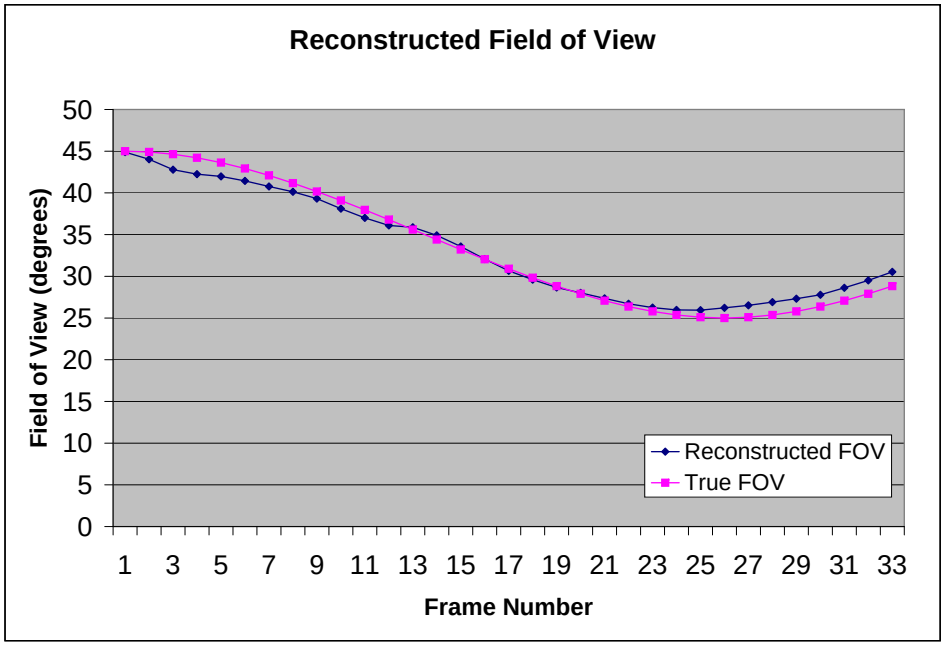
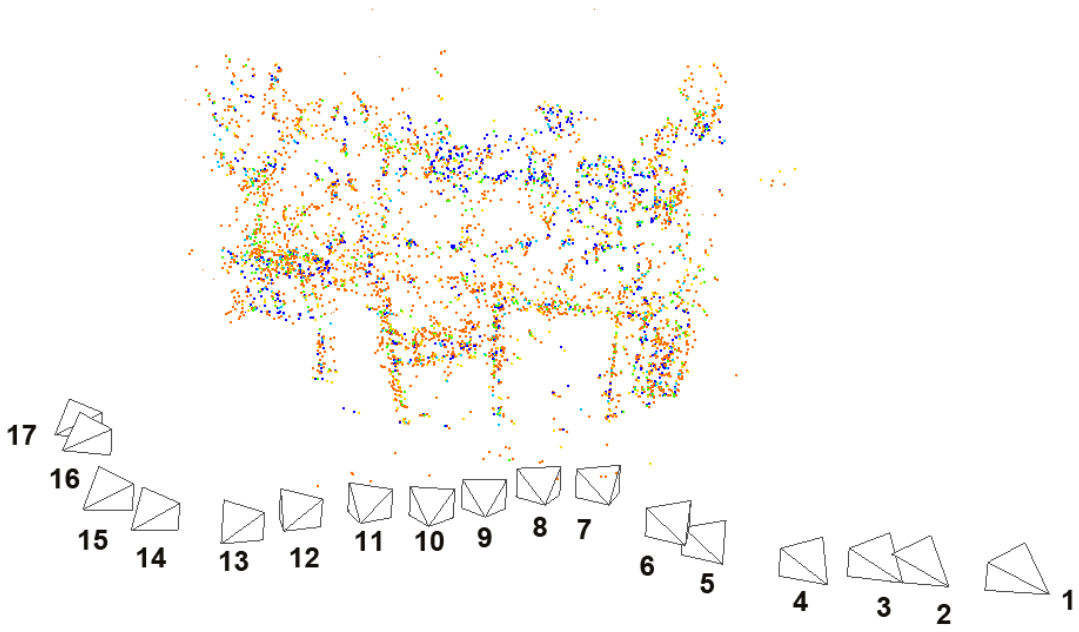
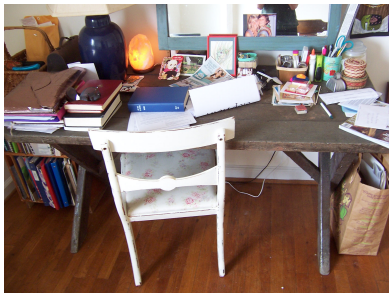


Figure 7.4. Reconstructed field-of-view in WeirdZoom reconstruction. Field of view is obtained by transforming the reconstructed focal length.



(a)

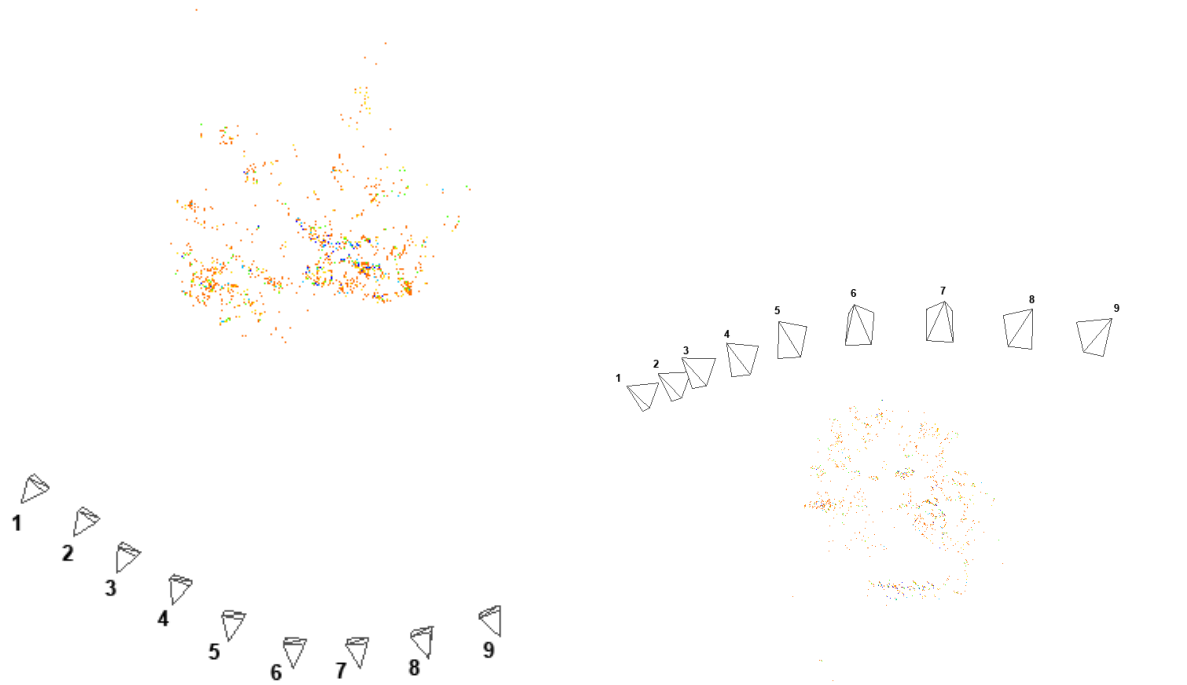
(b)



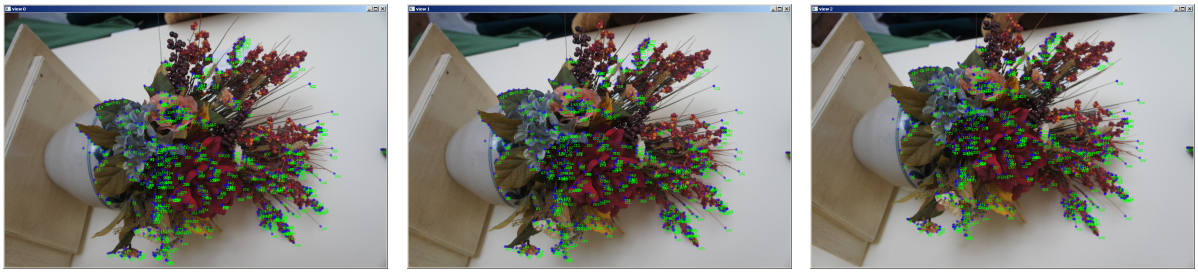
(c)

(d)

Figure 7.5. Sparse reconstruction from *DeskRecon*. (a) reconstruction. (b) frame 1. (c) frame 9. (d) frame 17.



(a) (b)



(c) (d) (e)

Figure 7.6. Sparse reconstruction of *Bouqet* with 9 views and 1349 points. The images from one of the view triplets is shown in (a,b,c), with 557 correspondences (a typical number).

Chapter 8

Surface Reconstruction

The structure from motion (SfM) techniques discussed in previous chapters have focused on building a reconstruction consisting of cameras and a set of structure points that correspond to interesting feature points that were tracked through the images. These structure points were integral for the purpose of deriving constraints on the relative pose of cameras, but they provide only a very sparse representation of scene geometry.

In this chapter we discuss techniques of constructing a conventional texture mapped mesh representation of the scene, beginning with a thorough survey of previous approaches (Section 8.1). Although surface reconstruction is not the focus of this work, we have implemented a proof of concept system that is described herein.

The first step in our approach is to compute depth maps associated with each image (Section 8.2). Given the known camera poses, each pixel in a depth map can then be back projected into a common 3D space to yield an extremely dense point cloud consisting of millions of points. From this point cloud we reconstruct a surface mesh (Section 8.3) and then back-project the images onto the surface mesh to reconstruct surface material (Section 8.3.1).

8.1 Background

The problem of reconstructing a dense scene representation from images taken by cameras with known pose is one of the oldest and most widely studied problems in computer vision. Essentially, this problem boils down to finding dense correspondences between images, because any two corresponding points can be triangulated to yield a depth value. The field has roots in the two related problems of stereo correspondence and optical flow.

In optical flow, the problem is to determine the motion field of pixels in video. In stereo correspondence the problem is to determine the dense disparity map between two images taken by a stereo camera. The only significant fundamental difference between these two problems

is that in stereo correspondence the relative pose between cameras is assumed to be known, allowing the search for correspondences to be restricted to epipolar lines, whereas this constraint does not exist in the optical flow problem.

Because of the large degree of similarity between these two problems, it is unsurprising that the literature under both umbrellas has converged to nearly identical matching algorithms. Both typically attempt some global or semi-global minimization of an energy cost that is formulated in terms of a matching cost and smoothness cost.

In the optical flow problem this has been done using variational methods [Bruhn and Weickert, 2006; Bruhn et al., 2005; Aubert et al., 1999; Cohen, 1993; Deriche et al., 1995; Nesi, 1993], PDE-based methods [Alvarez et al., 1999; Weickert and Schnorr, 2001], Markov random fields [Heitz and Bouthemy, 1993], diffusion [Proesmans, 1994], and various other methods [Black, 1991; Kumar et al., 1996; Nagel, 1983; Schnörr, 1994; Shulman and Herve, 1989]. Sometimes a coarse-to-fine approach or image warping is used to improved robustness and efficiency, as in Bruhn and Weickert [2006]; Anandan [1989]; Black and Jepson [1996]; Memin and Perez [2002]; Bruhn et al. [2005].

Increased accuracy has been obtained by using non-linearized models, as in Bruhn et al. [2005]; Nagel and Enkelmann [1986]; Alvarez et al. [2000], and additional constancy constraints beyond gray value constancy, such as constancy of the gradient [Uras, 1988; Tistarelli, 1994; Bruhn et al., 2005], constancy of the Hessian, the Laplacian, the gradient norm, the Hessian norm, and the determinant of the Hessian [Bruhn et al., 2005].

A good review of stereo correspondence algorithms is found in Scharstein and Szeliski [2002]. The first approach that attempts to minimize a semi-global minimization was dynamic programming on scanlines [Ohta and Kanade, 1985]. This approach finds a true global minimum but only considers horizontal smoothness constraints, resulting in streaking. This approach has been further developed in Scharstein and Szeliski [2002]; Falkenhagen [1997]; Kim et al. [2005]; Gonzlez et al. [1999]; Bobick and Intille [1999]; Geiger et al. [1995]; Woetzel and Koch [2004]; van Meerbergen et al. [2001], and a GPU implementation was demonstrated by Woetzel and Koch [2004].

More general minimization using full spatial smoothness has been achieved using belief propagation [Yang et al., 2006a,b; Sun et al., 2003; Felzenszwalb and Huttenlocher, 2006] and graph cuts [Boykov et al., 1998; Birchfield and Tomasi, 1999; Boykov et al., 2001; Kolmogorov and Zabih, 2001, 2002; Scharstein and Szeliski, 2002; Kim et al., 2003; Boykov and Kolmogorov, 2004]. Belief propagation is more easy to parallelize and has been implemented in real time on the GPU [Yang et al., 2006b], whereas graph cuts are slower but provide theoretical guarantees of a near-optimal solution that is much more robust in practice.

One of the remaining challenges is in obtaining good sub-pixel precision, which is often attempting by extracting planes and then trying to assign pixels to these planes. This approach

was first seen in [Birchfield and Tomasi, 1999] and has since been extended in [Klaus et al., 2006; Bleyer and Gelautz, 2005; Hong and Chen, 2004].

The first structure from motion systems have used methods of stereo correspondence between successive frames to compute depth maps. However, this is inferior to approaches that take advantage of additional views. The more generalized multi-view stereo problem has been approached from several perspectives using different surface representations, such as voxel grids [Vogiatzis et al., 2005; Labatut et al., 2007; Kutulakos, 2000], evolving level sets [Faugeras et al., 1999], evolving mesh [Esteban and Schmitt, 2004; Pons et al., 2007], graph cuts on convex hulls [Furukawa and Ponce, 2009; Prakoonwit and Benjamin, 2007], and multiple fused depth maps [Gargallo and Sturm, 2005; Kolmogorov et al., 2003; Kolmogorov and Zabih, 2002; Kang et al., 2001]. For a summary of these and other approaches, we refer the reader to the Seitz et al. [2006]; Dyer [2001]; Slabaugh et al. [2001].

As discussed in Collins [1996], voxel grids are a somewhat naive methodology due to their high time and space complexities and inherently low precision. We consider the voxel grid and convex hull approaches to be better suited to reconstructing simple objects in turn-table sequences. The evolving mesh approaches are attractive but their nonlinear nature makes them more applicable to refinement of some existing mesh.

The most general multi-view stereo approach is to compute depth maps by using a multi-view matching function. The concept of a multi-view matching cost dates back to Okutomi and Kanade [1993] where it was first used in a multiple baseline stereo rig, and later with an omni-direction multi-baseline stereo rig in Kang and Szeliski [1997]. As with all previous stereo correspondence and optical flow algorithms, these early techniques used a simple translation offset for image patches when searching in other views.

The same basic concept was generalized to arbitrary camera configurations in the plane sweep approach of Collins [1996], where the warping function was further improved by approximating all surface points by fronto-parallel planar patches. The plane sweep approach is largely equivalent to the optical flow approach of Szeliski [1999], which also took into account the effect of occlusions and formulated the problem in terms of a more global energy minimization, although it was not globally minimized. A GPU implementation of Collins [1996] was used in Yang and Pollefeys [2003], and an affine-illumination invariant version using the normalized cross correlation and low-confidence rejection heuristic was used in Goesele et al. [2006]. Multiple sweep directions for dominant planes in architectural scenes was used in Gallup et al. [2007]; Merrell et al. [2007].

8.2 Depth Map Estimation

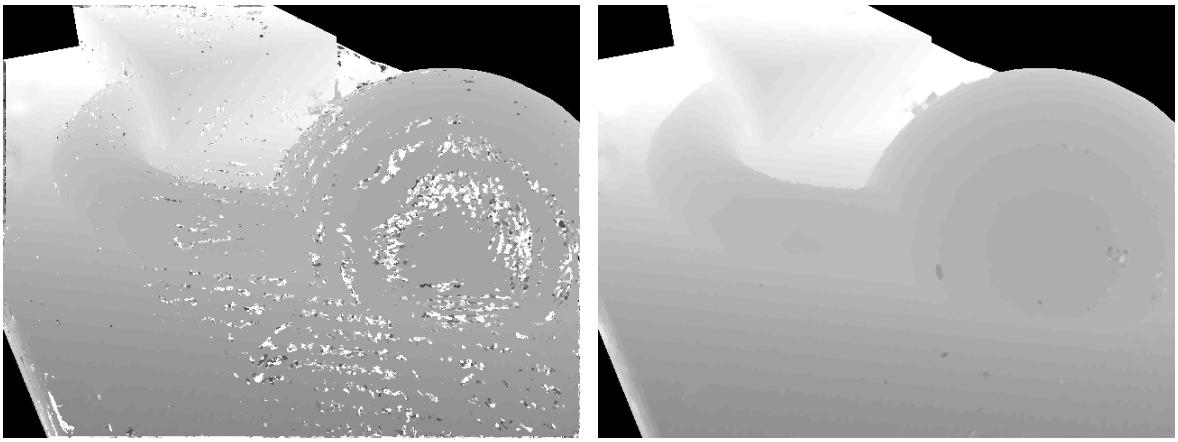
We compute depth maps by solving a global energy minimization problem in image space. The energy cost function consists of a primary term to enforce multi-view image similarity and a secondary term to enforce piecewise-smoothness constraints on the recovered depth map, which helps to resolve ambiguous matching in areas of low texture.

In other words, for each pixel location $\mathbf{x}$ in the ith image $\mathbf{f}_i$, we minimize a global cost function of the form

$$\sum_{\mathbf{x}\in\mathbf{f}_i} \mathcal{C}_i(\mathbf{x}, z(\mathbf{x})) + \sum_{\mathbf{x}'\in\mathcal{N}(\mathbf{x})} \mathcal{D}(z(\mathbf{x}), z(\mathbf{x}')), \tag{8.1}$$

where $z(\mathbf{x})$ is the depth at $\mathbf{x}$, $\mathcal{C}_i(\mathbf{x}, z)$ is the multi-view matching cost for pixel $\mathbf{x}$ in image i having depth z, $\mathcal{D}(z_1, z_2)$ is the smoothness cost for two adjacent depth values, and $\mathcal{N}(\mathbf{x})$ is the neighborhood of pixel $\mathbf{x}$.

This is the same general form of energy minimization that has been traditionally used in stereo correspondence, and was generalized to the multi-view optical flow problem in Szeliski [1999]. However, other plane sweep approaches for solving the multi-view stereo problem have only minimized the local matching cost [Yang and Pollefeys, 2003; Goesele et al., 2006; Gallup et al., 2007] without using a smoothness term. See Fig. 8.1 for an example of the benefits of global minimization with a smoothness term.



(a) (b)

Figure 8.1. Resolving ambiguous matchings using global optimization of piecewise smoothness constraint. (a) depth map obtained by locally maximizing the multi-view matching cost at each pixel. (b) Depth map obtained after global minimization with Graph Cuts after adding a discontinuity cost. Note that the camera poses used in this example were estimated using the proposed structure from motion approach and the depth search range for dense matching was automatically computed as in Section 8.2.4.

The multi-view matching cost is a normalized sum of weighted pairwise matching functions (see Section 8.2.1), where the weighting function can designed to give lower weight for view pairs that are expected to match poorly due to occlusions, perspective distortions or lighting changes (see Section 8.2.3).

Writing the pairwise matching between frames i and j of the image point ($\mathbf{x}$ having depth z as $\mathcal{M}_{ij}(\mathbf{x}, z)$, and similarly denoting the weighting by $w_{ij}(\mathbf{x}, z)$, we suggest using a multi-view matching function of the following form:

$$\mathcal{C}_i(\mathbf{x}, z) = \frac{\sum_{j \neq i} w_{ij}(\mathbf{x}, z) \mathcal{M}_{ij}(\mathbf{x}, z)}{\sum_{j \neq i} w_{ij}(\mathbf{x}, z)}. \tag{8.2}$$

8.2.1 Perspective Correct Matching

Given any point $\mathbf{x}$ in the ith image having depth z, the image point can be back-projected to a structure point $\mathbf{X}$, according to

$$\mathbf{X}(\mathbf{x}, z) = \mathbf{P}_i^{+} \mathbf{x} z + \mathbf{C}_i. \tag{8.3}$$

This structure point can be reprojected into any other image allowing the two image neigh-

borhoods to be compared using a photo-consistency function. In stereo correspondence it is typical to measure photo-consistency of the local neighborhood using the truncated Sum of Absolute Differences (SAD) under a simple translation (disparity) model.

Ideally, this image warping would be represented by a homography which is a perspective correct warping for planar surfaces. Previous plane-sweep approaches have assumed that each image point is the projection of some fronto-parallel surface patch [Collins, 1996; Goesele et al., 2006] so that the homography can be parameterized simply by depth.

Some recent approaches have attempted to determine the overall orientation of building facades [Gallup et al., 2007; Merrell et al., 2007] and then used the plane sweep with multiple sweep directions (i.e., multiple directions for each dominant surface normal in the scene), although this approach is highly specific to architectural scenes. We propose a more general approach by using multiple iterations, because the local surface normal can be estimated from the nearby surface points of the previous iteration.

Specifically, let $\mathbf{a}_1 \dots \mathbf{a}_5$ be the four-corners and center of the patch surrounding $\mathbf{x}_i$ in image i (as shown in Fig. 8.2). Each one of these points corresponds to an eye-ray that can be ray-traced against the existing reconstruction to find a corresponding structure point $\mathbf{S}_1 \dots \mathbf{S}_5$. Because the patch is small in image space, the projected structure points will usually cover a small patch on the surface as well. A small surface patch is well represented by plane, and the validity of this approximation can be tested by measuring the residual error of the least squares plane. If the structure points do not pass the test for planarity, then the local window spans a depth discontinuity and should be discarded.

Otherwise, when the surface patch is well approximated by a plane, there exists a homography $\mathbf{H}$ of $\mathbb{P}^2$ that transfers the planar patch as seen in view i to view j. In order to determine this homography, project the structure points $\mathbf{S}_1, \dots, \mathbf{S}_4$ into view j yielding image points $\mathbf{b}_1 \dots \mathbf{b}_4$. It is then straight forward to calculate the homography $\mathbf{H} : \mathbf{a}_k \mapsto \mathbf{b}_k, k = 1 \dots 4$ in closed form [Heckbert, 1989].

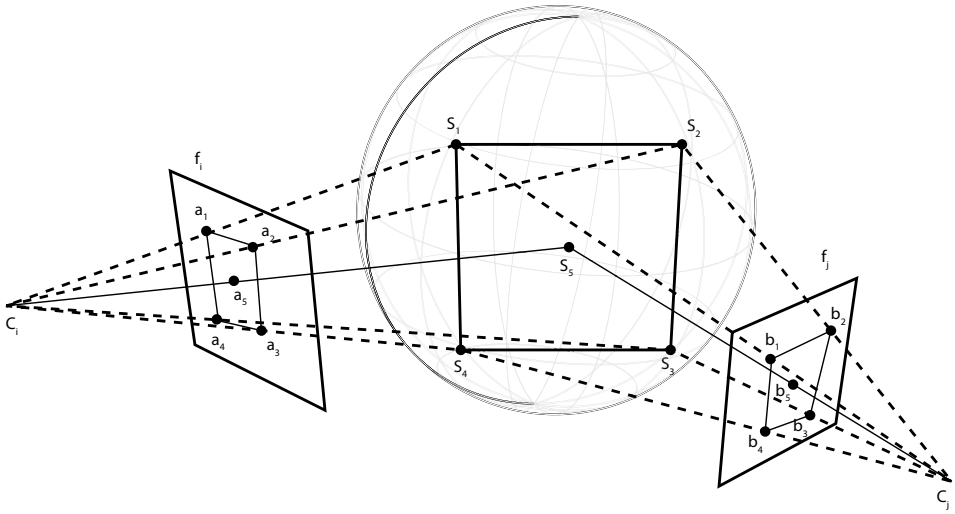


Figure 8.2. Diagram illustrating how to calculate the homography between two arbitrary views located at $\mathbf{C}_i$ and $\mathbf{C}_j$ by using the previous surface mesh. The four corners and center of the image patch in frame f_i are indicated by the points $\mathbf{a}_1 \dots \mathbf{a}_5$, and these rays intersect the previous surface (shown as a sphere for simplicity) at 3D points $\mathbf{S}_1 \dots \mathbf{S}_5$. Projecting these points onto frame f_j yields image points $\mathbf{b}_1 \dots \mathbf{b}_5$, and the homography can then be computed from the four correspondences $\mathbf{a}_i \leftrightarrow \mathbf{b}_i$, $i = 1 \dots 4$.

Because we wish to refine the estimate of depth while using the estimate of surface normal from the previous iteration, it is necessary to parameterize the homography by z. Specifically, if $\mathbf{S}_5 = (X, Y, Z)^\mathsf{T}$, and $\mathbf{N} = (N_x, N_y, N_z)^\mathsf{T}$ is the surface normal at $\mathbf{S}_5$, then this can be done by shifting the 3D structure points to

$$\mathbf{S}'_i = \mathbf{S}_i + \frac{z - Z}{N_z}\mathbf{N} \tag{8.4}$$

before reprojecting them to image points to calculate the homography.

8.2.2 Photo Consistency Function

Given two corresponding regions of an image, the *photo consistency* function returns the similarity between those regions. In stereo correspondence, a common choice is the truncated SAD or truncated SSD, which can also be used with an overall intensity gain multiplier as in Gallup et al. [2007]. Another choice is the Normalized Cross Correlation (NCC) which was used, for example, in Goesele et al. [2006]. Variance has also been used in Seitz and Dyer [1997] and Kutulakos [2000], although this is not very robust to illumination changes. A more general BRDF model was used in Treuille et al. [2004], but this increased generality comes at the cost of specificity.

The primary cause for outliers in traditional two-view stereo is partial occlusions, but in multi-view matching illumination changes and perspective distortions are more prevalent. Therefore, we prefer to use the NCC because it is invariant to affine changes in the local illumination.

8.2.3 Determining Visibility Weights

As the angle of camera view is increased, specular illumination changes become more significant. Therefore, we use an initial weighting function that takes this effect into account,

$$w_{ij}(\mathbf{x}, z) = \max(\mathbf{V}_i \cdot \mathbf{V}_j, 0)^{\alpha}, \tag{8.5}$$

where $\mathbf{V}_i$ is the ith view vector (i.e., camera z-axis), and α is a constant exponent that controls sensitivity.

After the initial pass has been completed a dense reconstruction will be available and this can be used to improve the visibility weighting by also using surface-dependent properties such as occlusions and surface normal to more accurately predict when perspective distortion and specular reflections will be high. Using this information, a better weighting function is given by

$$w_{ij}(\mathbf{x}, z) = \delta_i \delta_j \max(\mathbf{V}_i \cdot \mathbf{V}_j, 0)^{\alpha} \min(\max(\mathbf{N} \cdot \mathbf{V}_i, 0), \max(\mathbf{N} \cdot \mathbf{V}_j, 0))^{\beta}, \tag{8.6}$$

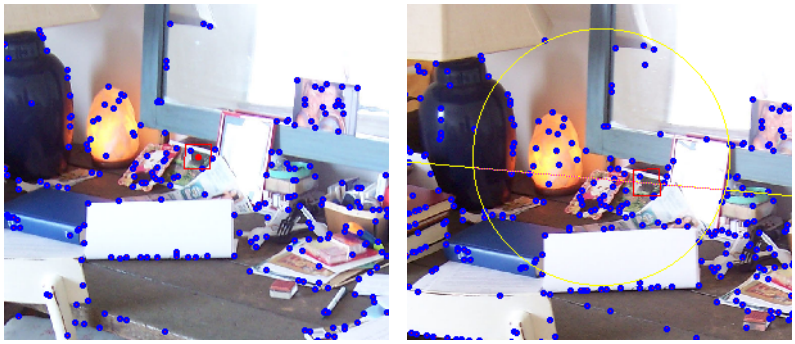
where δ_i is the binary visibility of $\mathbf{X}(\mathbf{x}, z)$ in view i (1 if visible) and $\mathbf{N}$ is the approximate surface normal at $\mathbf{X}(\mathbf{x}, z)$. The final term weights the photo-consistency function by the maximum angle between the surface normal and view vector of either view, which coincides with local perspective distortion of the patch.

8.2.4 Depth Autoranging

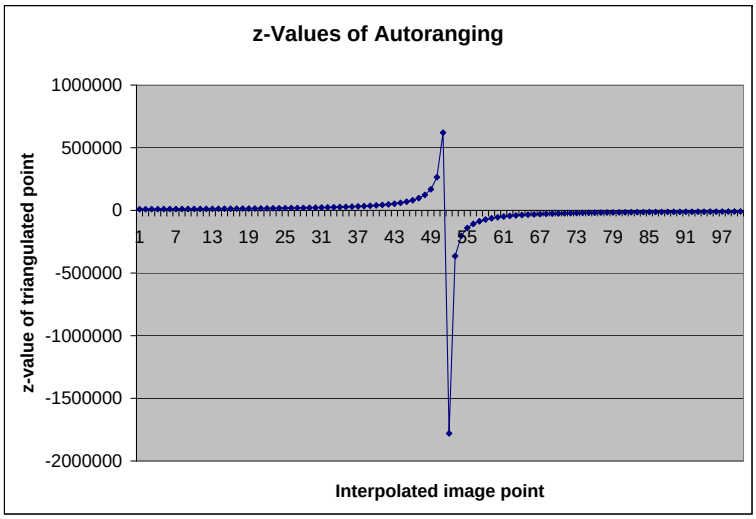
The determination of depth range is an important problem hat has not been well addressed in previous approaches, which typically use a hard-coded range, or assume that the scene is contained in some bounding box or convex hull. However, this information is not available in general, and the convex hull of sparse points from SfM is not guaranteed to contain all visible scene surfaces. We present here an approach that overcomes these limitations.

Specifically, we use the image space guided matching constraints; between any two consecutive frames in the image sequence one can estimate the homography to yield an approximate point match (circular radius) for any pixel. We intersect this search circle with the epipolar

line and then interpolate between the two intersection points to acquire potential match points in the second image. Each potential match corresponds to a depth value that can be obtained via triangulation (see Fig. 8.3).



(a) (b)



(c)

Figure 8.3. Example of depth autoranging. (a) a feature point is selected in the first frame. (b) a search range (yellow circle) is determined from spatio-temporal constraint in image space. This is intersected with the epipolar line (yellow) to get two endpoints. Now potential match points are interpolated along the epipolar line (purple dots). (c) Graph showing the z-values corresponding to each potentially matching interpolated image point. Note that this is not a uniform sampling of depth, and half the points have negative depth (behind the camera).

8.2.5 Confidence Heuristic

When the matching cost function exhibits a single easily identifiable minimum then this minimum clearly represents the true depth of a point. However, it is common that the matching

function has a low value for a large range of depths, or has multiple specific minima. In either case, the match is ambiguous and should be trusted less.

We have constructed a heuristic function to evaluate the confidence of depth assessments that has has a low value *only* when the response has a *single* unique minimum. If we denote the matching cost and z-value for the ith check point by the tuple (c_i, z_i), with a minimum at (c_{min}, z_{min}), then our heuristic function is given by

$$\sum_i |z_i - z_{min}| \exp(-(c_i - c_{min})/(c_{max} - c_{min})\alpha). \tag{8.7}$$

We note that this is somewhat similar to the heuristic used in Merrell et al. [2007], although their heuristic fails to distinguish between functions with *multiple* minima, which ours gracefully handles by down-weighting due to the inclusion of $|z_i - z_{min}|$.

8.3 Surface Mesh Reconstruction

Fitzgibbon and Zisserman [1996], and later Nister [2001a], have built surface models by extracting RANSAC planes, but this is a slow non-optimal process that does not produce clean meshes. There have been several other simple methods suggested, such as projecting a depth map onto a tesselated quad as in Pollefeys et al. [2004], or the quad tree approach used in Pollefeys et al. [2008]. However these approaches are biased by the order of views. An unbiased alternative is to back-project all depth maps into structure points in a common 3D space and then use a method of surface interpolation such as Poisson reconstruction Kazhdan et al. [2006].

8.3.1 Reconstructed Surface Material

In a Lambertian model (accurate for most non-specular materials), the radiance from a point on the surface is independent of outgoing direction. As a result, the spatially varying diffuse albedo can be represented by a single texture map that is over-determined by each additional view with a non-occluded view of the surface point.

In general, no single view will have an unobstructed view of every surface point, so all views should be projected and combined into a single texture. Many texels will contain multiple samples, and from this set of samples the diffuse albedo can be estimated. Nishino et al. [2001] used the minimum sample as the diffuse albedo, but this is not very robust, because any single image could contain some corruptions or have high projective distortion, and in reality diffuse reflections are not entirely view independent. Therefore, we instead use a weighted average of the non-occluded samples, weighted by the dot product between the local surface normal and

the view vector. This gives lower weight to views that projected from nearly perpendicular angles (having greater distortion effects). The local surface normal can be found as a function of texture coordinates in the same way as the spatial position – that is, by first rendering to texture.

Yu et al. [1999] proposed the method of "inverse global illumination" that uses images, model, and lighting information to iteratively estimate parameters of the Ward [1992] reflectance model. They also assumed a known partitioning of the scene into large regions of constant BRDF. Unfortunately, neither the direct lighting information nor this partitioning would be available after making an automatic structure from motion reconstruction.

Lensch et al. [2001] proposed a method that improves upon the partitioning problem by using a variant of Lloyd's [Lloyd, 1982] hierarchical divisive clustering algorithm to cluster the lumitexels into regions of roughly constant BRDF (using a Lafortune [Lafortune et al., 1997] reflectance model). Their clustering method chooses the principal eigenvector as the split plane, and they used a novel iterative procedure to find the optimal split location on that plane. After each split, they re-balance by assigning each lumitexel to the best fitting cluster. After finding these large clusters, they increase detail by representing the BRDF of each lumitexel as the linear combination of basis BRDFs. The basis BRDFs for each cluster are determined by Principal Function Analysis (PFA). Although their algorithm works well for simple objects with largely constant texture, significant detail will invariably be lost on more complex textures, and direct lighting information is still required.

Because diffuse albedo is largely view-independent (in the Lambertian model, it is completely view-independent), some researchers have chosen to clearly separate detailed spatially varying diffuse albedo from the more difficult to estimate and slowly varying specular components. For example, Sato et al. [1997] used the Torrence-Sparrow [Torrance and Sparrow, 1967] reflection model and estimated diffuse albedo for each surface texel, but interpolated specular properties much more sparsely. Still, their method requires known lighting.

Nishino et al. [2001] proposed a method similar to Sato et al. [1997] that does not rely on known lighting. Instead, they attempt to separate the view-independent radiance from the view-dependent radiance at each texel, and then back-project the view-dependent radiance onto an 'illumination hemisphere'. A set of point lights are made to approximate the illumination hemisphere, and used to estimate the view-dependent (specular) component of a global (i.e., not spatially varying) Torrence-Sparrow reflection model. Clearly, this assumption of a global specular constant is not applicable to reconstructions containing many different types of materials, but could be relaxed using an automatic partitioning strategy as in Lensch et al. [2001], or an interpolation method such as Sato et al. [1997].

8.4 Example Surface Reconstruction

Depth maps were reconstructed to correspond with all 50 frames in the test sequence, as shown in Fig. 8.4. After reconstructing all depth buffers, the total set of data points (roughly 15 million) was fused into a large point cloud where cross-validation was used to reject points that do not agree from multiple depth maps (Fig. 8.5).

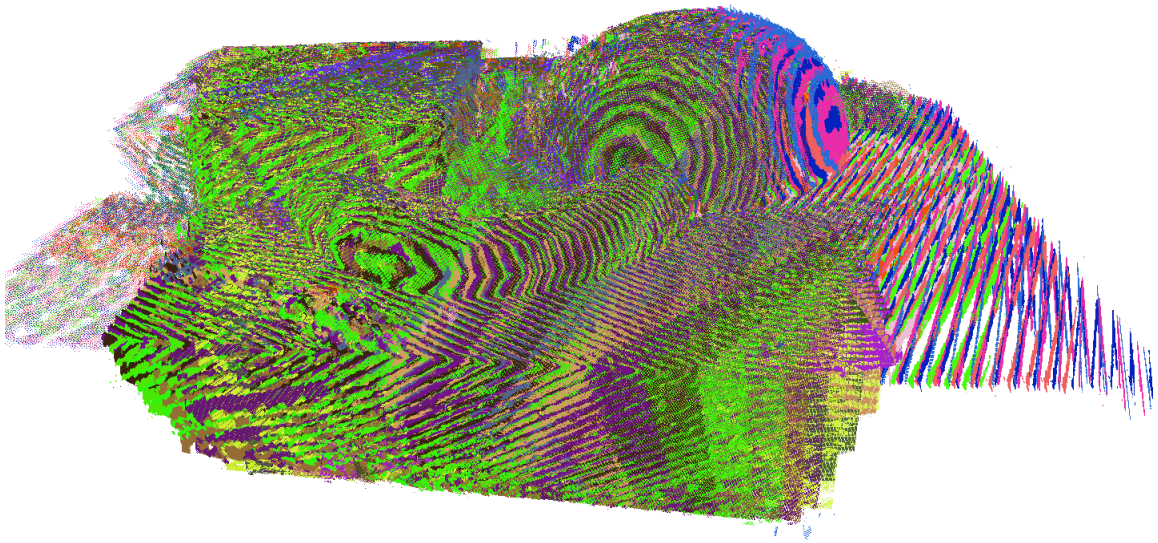


Figure 8.5. Fused depth maps. Each depth map is back-projected into a set of 3D points which is fused together in a common space. The controbution from each depth map is randomly colored for visual distinction.

Then, Poisson reconstruction was used to generate a surface mesh, as shown in Fig. 8.6. It is shown here from 2 novel views with a reconstructed texture applied, as well as being overlayed onto the true model with no texture. It should be noted that the reason the reconstructed model has a large bulge on the underside is that the model was never observed from this angle, but Poisson reconstruction always tries to form a closed isosurface. This region can be easily culled if desired.

The reconstruction of texture was based on that described in Section 8.3.1, although it is not fully complete. First, the point cloud was used to generate surface normals. Then the model was hand-mapped into texture groups, as shown in Fig. 8.7. This figure shows color groups which were mapped as planes (or, in the case of the dome, spherically) in (a). In (b), the spatial surface position was shown rendered into texture space, and in (c) the local normals were rendered into texture space. Note that these textures have been rendered using a custom

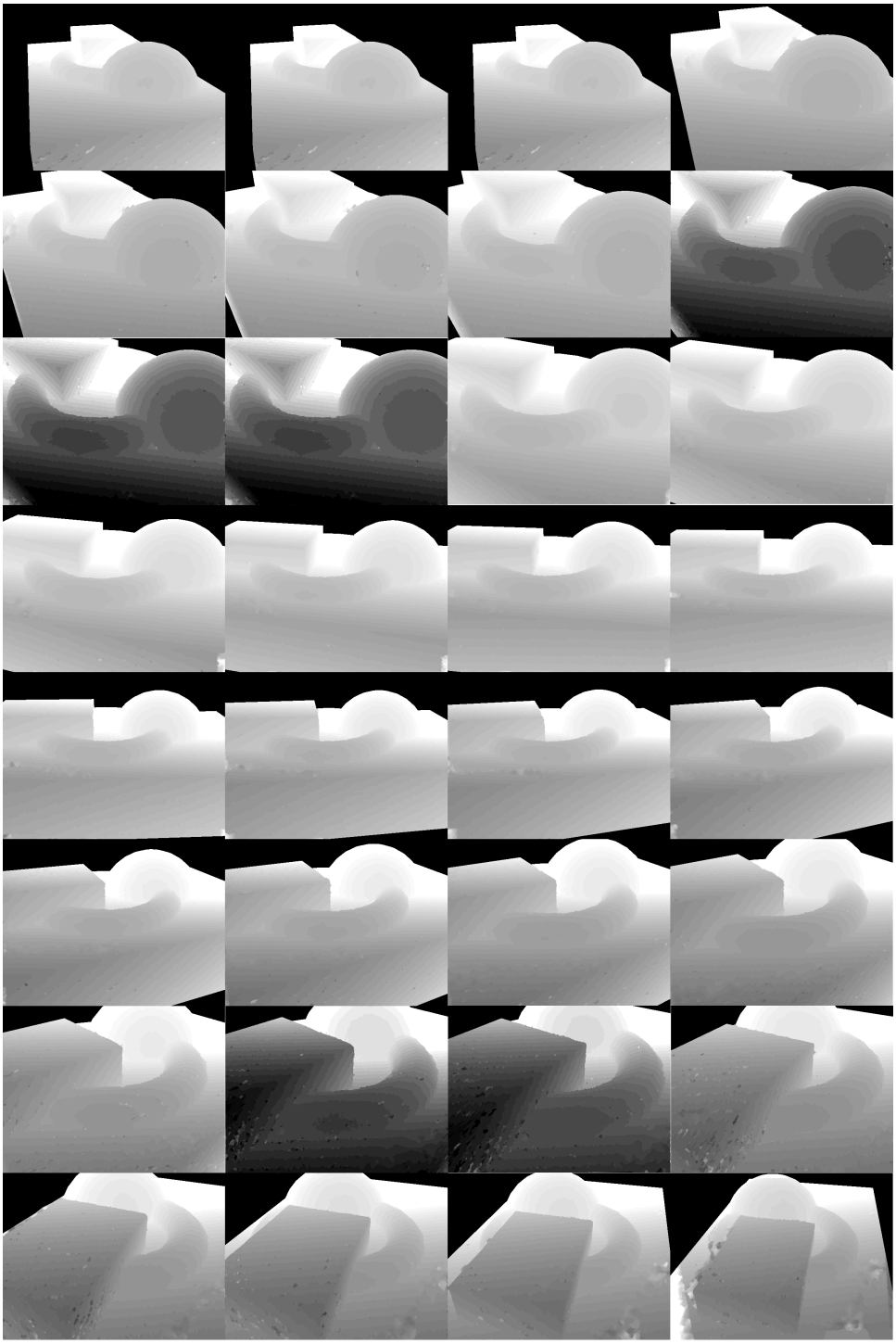
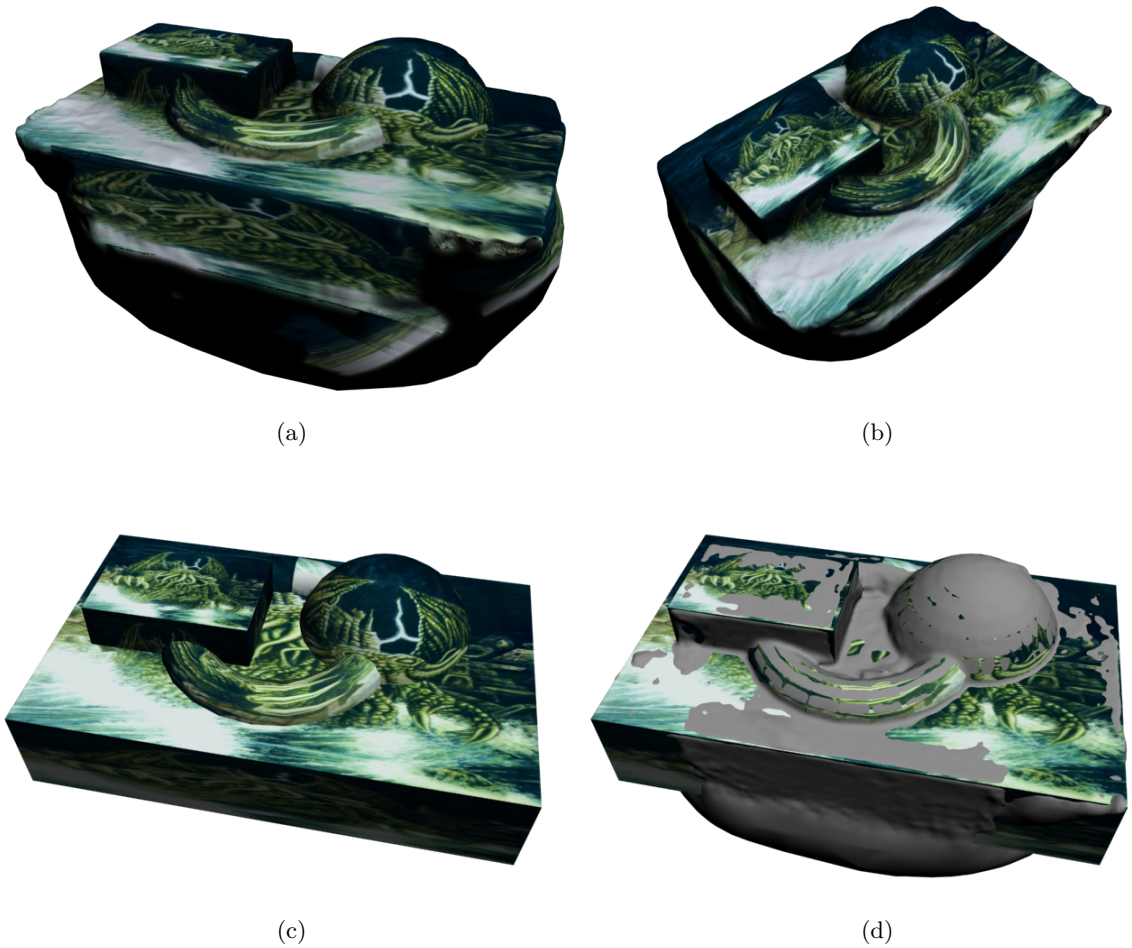


Figure 8.4. *WeirdTest* image sequence (in book-reading order). 50 frames total.



(a) (b)

(c) (d)

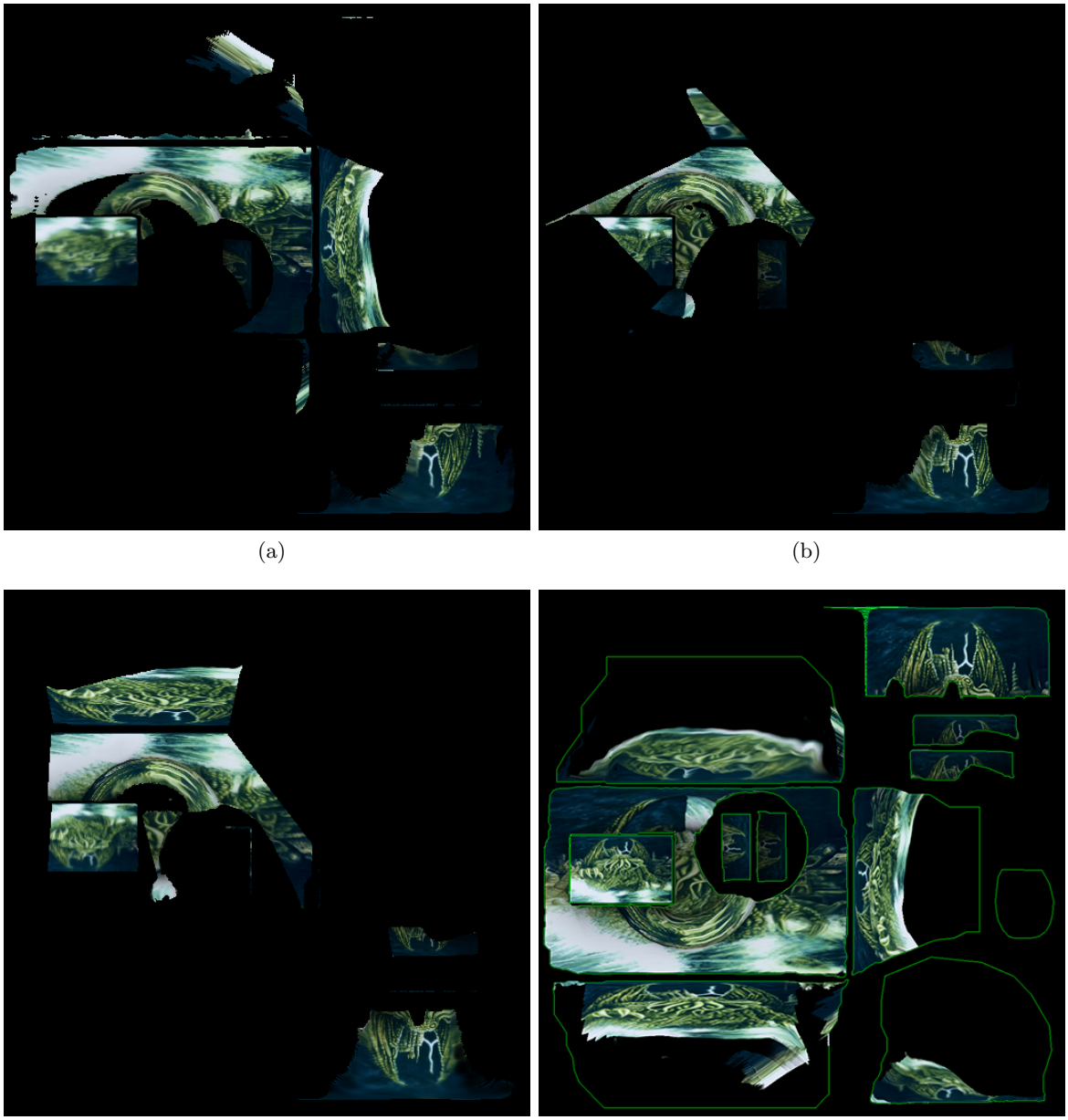
Figure 8.6. Initial mesh reconstruction. (a) and (b) are two views of the reconstructed model. (c) the original model. (d) untextured reconstructed model overlaid onto the original model.

rasterization engine because of the need to draw outside of triangle borders to avoid seams when bilateral filtering the texture maps. Thus, it is not possible to use existing rasterization engines. This manual mapping will be replaced with automatic mapping in the future, but for now it is helpful to make the maps more visually discernible.

(a) (b) (c)

Figure 8.7. Mapped model. (a) mapping groups used to parameterize model surface. (b) XYZ map rendered into texture space. (c) Normal map rendered into texture space.

After mapping the model, each individual view was projected onto the visible model surface in texture space, taking into account occlusions. A few examples are shown in (a) (b) (c) of Fig. 8.8. Part (d) shows the overall reconstruction of texture as linear combination weighted by visibility and surface normal orientation. This is outlined in green just to show the uv-groups more clearly. Note that regions which remain purely black were not visible from any view (e.g., the underside of the model), so they could be culled if desired.



(a) (b)

(c) (d)

Figure 8.8. View image projected onto model texture. Figures (a) - (c) show selected views. Figure (d) shows the composite texture based on the weighted linear combination from all 50 views. UV-groups are outlined in green to show areas of the model that were not visible in any view.

Chapter 9

Conclusions

This dissertation has approached one of the largest and most difficult tasks in computer vision: that of reconstructing an accurate 3D representation of the world from an uncalibrated video or image series. The overall architecture and design of this system has been carefully thought out and motivated in terms of theoretical concerns to achieve maximum accuracy and robustness.

Design and implementation of the system has required integration from many sub-problems including feature point detection, tracking, wide-baseline matching, robust estimation of the fundamental matrix and trifocal tensor, keyframe detection, camera resectioning, triangulation, autocalibration, bundle adjustment, dense multi-view stereo correspondence and surface reconstruction.

Minor contributions have been made in many of these sub-problems, but the most significant contributions consist of an overall system architecture for general uncalibrated reconstruction motivated by theoretical concerns of accuracy and stability (Chapter 2), a simple and effective algorithm for keyframe detection (Algorithm 1), a normalization scheme to promote uniform feature point distribution (Section 3.2), a derivation of the circular constraints on the trifocal tensor (Section 4.1.2.5), a structure invariant maximum likelihood solution to merging projective reconstructions (Section 5.4), a maximum likelihood solution to the autocalibration problem (Section 6.2), and a parallel decomposition for bundle adjustment (Section 7.3.4).

Although this work has focused mainly on SfM techniques, some advancements to dense reconstruction have also been made, including the use of graph-cut optimized global consistency for multi-view stereo correspondence (Section 8.2), a geometry based planar approximation to improve the perspective correctness of plane-sweep based approaches during stereo matching (Section 8.2.1), visibility weighting for stereo matching function that takes into account specular lighting (Section 8.2.3), a method of depth auto-ranging to eliminate need for a bounding box during ray marching (Section 8.2.4), and a confidence heuristic that takes into account multiple minima (Section 8.2.5).

Future work will focus on fine tuning the initial sparse reconstruction to obtain improved precision, as we have learned that methods of dense reconstruction are extremely sensitive to the quality of the sparse reconstruction. We will also focus on methods for surface generation that are more robust than Poisson reconstruction, and we propose to integrate the evolving mesh based approaches in the final stages.

REFERENCES

M. Agrawal. On automatic determination of varying focal lengths using semidefinite programming. In *Image Processing, 2004. ICIP '04. 2004 International Conference on*, volume 5, pages 3379 – 3382 Vol. 5, oct. 2004. doi: 10.1109/ICIP.2004.1421839.

M. Agrawal. Practical camera auto calibration using semidefinite programming. In *Motion and Video Computing, 2007. WMVC '07. IEEE Workshop on*, pages 20 –20, feb. 2007. doi: 10.1109/WMVC.2007.39.

L. Alvarez, J. Esclarin, M. Lefebure, and J. Sanchez. A pde model for computing optical flow. In *Proc. XVI Congreso de Ecuaciones Diferenciales y Aplicaciones*, pages 1349–1356, Las Palmas de Gran Canaria, Spain, 1999.

L. Alvarez, J. Weickert, and J. Sanchez. Reliable estimation of dense optical flow fields with large displacements. *Intl. Journal of Computer Vision*, 39(1):41–56, 2000.

P. Anandan. A computational framework and an algorithm for the measurement of visual motion. *Intl. Journal of Computer Vision*, 2:283–310, 1989.

G. Aubert, R. Deriche, and P. Kornprobst. Computing optical flow via variational techniques. *SIAM Journal on Applied Mathematics*, 60(1):156–182, 1999.

Simon Baker and Iain Matthews. Lucas-kanade 20 years on: A unifying framework. *Int. J. Comput. Vision*, 56:221–255, February 2004. ISSN 0920-5691. doi: 10.1023/B:VISI.0000011205.11775.fd. URL `http://portal.acm.org/citation.cfm?id=964568.964604`.

Klaus-Jurgen Bathe and Edward L. Wilson. *Numerical Methods in Finite Element Analysis*. Prentice-Hall, 1976.

P. A. Beardsley, A. Zisserman, and D. W. Murray. Sequential updating of projective and affine structure from motion. *Int. J. Comput. Vision*, 23:235–259, June 1997. ISSN 0920-5691. doi: 10.1023/A:1007923216416. URL `http://portal.acm.org/citation.cfm?id=261678.261681`.

Christian Beder and Richard Steffen. Determining an initial image pair for fixing the scale of a 3d reconstruction from an image sequence. In *Pattern Recognition*, volume 4174 of *Lecture Notes in Computer Science*, pages 657–666. Springer Berlin / Heidelberg, 2006.

Jon L Bentley. A survey of techniques for fixed radius near neighbor searching. Technical report, Stanford University, Stanford, CA, USA, 1975.

Paul J. Besl and Neil D. McKay. A method for registration of 3-d shapes. *IEEE Trans. Pattern Anal. Mach. Intell.*, 14:239–256, February 1992. ISSN 0162-8828. doi: 10.1109/34.121791. URL `http://portal.acm.org/citation.cfm?id=132013.132022`.

S. Birchfield and C. Tomasi. Multiway cut for stereo and motion with slanted surfaces. In *Intl. Conf. on Computer Vision*, pages 489–495, 1999.

M. J. Black. Recursive non-linear estimation of discontinuous flow fields. In *European Conference on Computer Vision*, volume 800, pages 138–145, 1991.

M.J. Black and A. Jepson. Estimating optical flow in segmented images using variable-order parametric models with local deformations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(10):972–986, 1996.

M. Bleyer and M. Gelautz. Graph-based surface reconstruction from stereo pairs using image segmentation. In *Proc. of SPIE Symposium on Electronic Imaging*, volume 5665, pages 288–299, 2005.

A. F. Bobick and S. S. Intille. Large occlusion stereo. *Intl. Journal of Computer Vision*, 33(3): 181–200, 1999.

Benoit Bocquillon, Adrien Bartoli, Pierre Gurdjos, and Alain Crouzil. On constant focal length self-calibration from multiple views. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, 0:1–8, 2007. doi: http://doi.ieeecomputersociety.org/10.1109/CVPR.2007.383066.

Sylvain Bougnoux. From projective to euclidean space under any practical situation, a criticism of self-calibration. In *ICCV '98: Proceedings of the Sixth International Conference on Computer Vision*, page 790, Washington, DC, USA, 1998. IEEE Computer Society. ISBN 81-7319-221-9.

Y. Boykov and V. Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *IEEE Trans. on PAMI*, 26(9):1124–1137, 2004.

Y. Boykov, Olga Veksler, and Ramin Zabih. Markov random fields with efficient approximations. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 648–655, 1998.

Y Boykov, O Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23:1222–1239, 2001.

A. Bruhn and J. Weickert. A multigrid platform for real-tyime motion computation with discontinuity-preserving variational methods. *International Journal of Computer Vision*, 70 (3):257–277, 2006.

A. Bruhn, T. Brox, S. Didas, and J. Weickert. Highly accurate optic flow computation with theoretically justified warping. *International Journal of Computer Vision*, 67(2):141–158, 2005.

Nikos Canterakis. A minimal set of constraints for the trifocal tensor. In *Proc. of the 6th European Conf. on Computer Vision*, pages 84–99, 2000.

Stefan Carlsson and Daphna Weinshall. Dual computation of projective shape and camera positions from multiple images. *Int. J. Comput. Vision*, 27(3):227–241, 1998. ISSN 0920-5691. doi: http://dx.doi.org/10.1023/A:1007961913417.

M. Chandraker, S. Agarwal, F. Kahl, D. Nister, and D. Kriegman. Autocalibration via rank-constrained estimation of the absolute quadric. In *Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on*, pages 1 –8, june 2007a. doi: 10.1109/CVPR.2007.383067.

M. Chandraker, S. Agarwal, D. Kriegman, and S. Belongie. Globally optimal affine and metric upgrades in stratified autocalibration. In *ICCV07*, pages 1–8, 2007b.

Manmohan Chandraker, Sameer Agarwal, David Kriegman, and Serge Belongie. Globally optimal algorithms for stratified autocalibration. *International Journal of Computer Vision*, 90: 236–254, 2010. ISSN 0920-5691. URL `http://dx.doi.org/10.1007/s11263-009-0305-2`. 10.1007/s11263-009-0305-2.

Y. Chen and G. Medioni. Object modeling by registration of multiple range images. In *Robotics and Automation, 1991. Proceedings., 1991 IEEE International Conference on*, pages 2724 – 2729 vol.3, apr 1991. doi: 10.1109/ROBOT.1991.132043.

S. Christy and R. Horaud. Euclidean shape and motion from multiple perspective views by affine iterations. *IEEE Trans. on Pattern Analysis and Machine Int.*, 18(11), 1996.

Ondrej Chum, Jiri Matas, and Josef Kittler. Locally optimized ransac. In *DAGM-Symposium*, pages 236–243, 2003.

Ondrej Chum, Tomas Werner, and Jiri Matas. Two-view geometry estimation unaffected by a dominant plane. In *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 1 - Volume 01*, CVPR '05, pages 772–779, Washington, DC, USA, 2005. IEEE Computer Society. ISBN 0-7695-2372-2. doi: http://dx.doi.org.www.lib.ncsu.edu:2048/10.1109/CVPR.2005.354. URL `http://dx.doi.org.www.lib.ncsu.edu:2048/10.1109/CVPR.2005.354`.

Brian Clipp, Jongwoo Lim, Jan-Michael Frahm, and Marc Pollefeys. Parallel, real-time visual slam. In *Computer Vision and Pattern Recognition, 2010. Proceedings. IEEE Conference on*, 2010.

I. Cohen. Nonlinear variational method for optic flow computation. In *Proc. Eighth Scandinavian Conference on Image Analysis*, volume 1, pages 523–530, Tromso, Norway, 1993.

R.T. Collins. A space-sweep approach to true multi-image matching. In *Computer Vision and Pattern Recognition, 1996. Proceedings CVPR '96, 1996 IEEE Computer Society Conference on*, pages 358 –363, jun 1996. doi: 10.1109/CVPR.1996.517097.

R. Deriche, P. Kornprobst, and G. Aubert. Optical-flow estimation while preserving its discontinuities: a variational approach. In *Proc. Second Asian Conference on Computer Vision*, volume 2, pages 290–295, Singapore, 1995.

F. Devernay and O.D. Faugeras. Automatic calibration and removal of distortion from scenes of structured environments. In *SPIE*, volume 2567, pages 62–72, July 1995.

C.R. Dyer. Volumetric scene reconstruction from multiple views. In *FIU01*, pages 469–489, 2001.

Chris Engels, Henrik Stewenius, and David Nister. Bundle adjustment rules. In *Photogrammetric Computer Vision*, 2006.

Carlos Hernández Esteban and Francis Schmitt. Silhouette and stereo fusion for 3d object modeling. *Comput. Vis. Image Underst.*, 96(3):367–392, 2004. ISSN 1077-3142. doi: http://dx.doi.org/10.1016/j.cviu.2004.03.016.

L. Falkenhagen. Hierarchical block-based disparity estimation considering neighborhood constraints. In *Proc. Intl. Workshop on SNHC and 3D Imaging*, Rhodes, Greece, 1997.

M. Farenzena, A. Fusiello, and R. Gherardi. Structure-and-motion pipeline on a hierarchical cluster tree. In *Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference on*, pages 1489 –1496, 2009. doi: 10.1109/ICCVW.2009.5457435.

O. Faugeras and B. Mourrain. On the geometry and algebra of the point and line correspondences between n images. In *ICCV '95: Proceedings of the Fifth International Conference on Computer Vision*, page 951, Washington, DC, USA, 1995. IEEE Computer Society. ISBN 0-8186-7042-8.

O. Faugeras and T. Papadopoulo. A nonlinear method for estimating the projective geometry of 3 views. In *ICCV '98: Proceedings of the Sixth International Conference on Computer Vision*, page 477, Washington, DC, USA, 1998. IEEE Computer Society. ISBN 81-7319-221-9.

Olivier Faugeras and Renaud Keriven. Complete dense stereovision using level set methods. In *in Proc. 5th European Conf. on Computer Vision*, pages 379–393, 1998.

Olivier Faugeras, Renaud Keriven, and Keywords Variational Principles. Variational principles, surface evolution, pde's, level set methods and the stereo problem. *IEEE Transactions on Image Processing*, 7:336–344, 1999.

Olivier Faugeras, Quang-Tuan Luong, and T. Papadopoulou. *The Geometry of Multiple Images: The Laws That Govern The Formation of Images of A Scene and Some of Their Applications*. MIT Press, Cambridge, MA, USA, 2001. ISBN 0262062208.

Olivier D. Faugeras. What can be seen in three dimensions with an uncalibrated stereo rig. In *ECCV '92: Proceedings of the Second European Conference on Computer Vision*, pages 563–578, London, UK, 1992. Springer-Verlag. ISBN 3-540-55426-2.

Olivier D. Faugeras, Quang-Tuan Luong, and Stephen J. Maybank. Camera self-calibration: Theory and experiments. In *ECCV '92: Proceedings of the Second European Conference on Computer Vision*, pages 321–334, London, UK, 1992. Springer-Verlag. ISBN 3-540-55426-2.

P. F. Felzenszwalb and D. P. Huttenlocher. Efficient belief propagation for early vision. *Int. Journal of Computer Vision*, 70(1), 2006.

Martin A. Fischler and Robert C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24: 381–395, June 1981. ISSN 0001-0782. doi: http://doi.acm.org/10.1145/358669.358692. URL `http://doi.acm.org/10.1145/358669.358692`.

A. Fitzgibbon and A. Zisserman. Automatic camera recovery for closed or open image sequences. In *Proc. of the 5th European Conf. on Computer Vision*, pages 311–326, 1998.

Andrew W. Fitzgibbon and Andrew Zisserman. Automatic 3d model acquisition and generation of new images from video sequences. In R. Koch and L. VanGool, editors, *Proceedings of European Signal Processing Conference (EUSIPCO)*, page 12611269, 1996.

Jan-Michael Frahm and Marc Pollefeys. Ransac for (quasi-)degenerate data (qdegsac). In *Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Volume 1*, pages 453–460, Washington, DC, USA, 2006. IEEE Computer Society. ISBN 0-7695-2597-0. doi: 10.1109/CVPR.2006.235. URL `http://portal.acm.org.www.lib.ncsu.edu:2048/citation.cfm?id=1153170.1153504`.

Jan-Michael Frahm, Pierre Fite-Georgel, David Gallup, Tim Johnson, Rahul Ragauram, Changchang Wu, Yi-Hung Jen, Enrique Dunn, Brian Clipp, Svetlana Lazebnik, and Marc Pollefeys. Building rome on a cloudless day. In *ECCV 2010: Proceedings of the 11th European Conference on Computer Vision*, pages 368–381, 2010.

Yasutaka Furukawa and Jean Ponce. Carved visual hulls for image-based modeling. *Int. J. Comput. Vision*, 81(1):53–67, 2009. ISSN 0920-5691. doi: http://dx.doi.org/10.1007/s11263-008-0134-8.

A. Fusiello, E. Trucco, T. Tommasini, and V. Roberto. Improving feature tracking with robust statistics. *Pattern Analysis and Applications*, 2:312–320, 1999. ISSN 1433-7541. URL `http://dx.doi.org/10.1007/s100440050039`. 10.1007/s100440050039.

A. Fusiello, A. Benedetti, M. Farenzena, and A. Busti. Globally convergent autocalibration using interval analysis. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 26(12):1633 – 1638, dec. 2004. ISSN 0162-8828. doi: 10.1109/TPAMI.2004.125.

D. Gallup, J.-M. Frahm, P. Mordohai, Qingxiong Yang, and M. Pollefeys. Real-time plane-sweeping stereo with multiple sweeping directions. In *Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on*, pages 1 –8, june 2007. doi: 10.1109/CVPR.2007.383245.

Yongying Gao and H. Radha. A multistage camera self-calibration algorithm. In *Acoustics, Speech, and Signal Processing, 2004. Proceedings. (ICASSP '04). IEEE International Conference on*, volume 3, pages iii – 537–40 vol.3, may. 2004. doi: 10.1109/ICASSP.2004.1326600.

Pau Gargallo and Peter Sturm. Bayesian 3d modeling from images using multiple depth maps. In *CVPR '05: Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 2*, pages 885–891, Washington, DC, USA, 2005. IEEE Computer Society. ISBN 0-7695-2372-2. doi: http://dx.doi.org/10.1109/CVPR.2005.84.

D. Geiger, B. Ladendorf, and A. Yuille. Occlusions and binocular stereo. *Int. Journal of Computer Vision*, 14(3):211–226, 1995.

Riccardo Gherardi and Andrea Fusiello. Practical autocalibration. In *ECCV 2010: Proceedings of the 11th European Conference on Computer Vision*, 2010.

Michael Goesele, Brian Curless, and Steven M. Seitz. Multi-view stereo revisited. In *CVPR '06: Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 2402–2409, Washington, DC, USA, 2006. IEEE Computer Society. ISBN 0-7695-2597-0. doi: http://dx.doi.org/10.1109/CVPR.2006.199.

Gene H. Golub and Charles F. Van Loan. *Matrix Computations (Johns Hopkins Studies in Mathematical Sciences)(3rd Edition)*. The Johns Hopkins University Press, 3rd edition, October 1996a. ISBN 0801854148.

Gene H. Golub and Charles F. Van Loan. *Matrix Computations (Johns Hopkins Studies in Mathematical Sciences)(3rd Edition)*. The Johns Hopkins University Press, 3rd edition, October 1996b. ISBN 0801854148.

Rafael C. Gonzlez, Jose A. Cancelas, Juan C. Alvarez, Jose A. Fernndez, and Ignacio Alvarez. Dynamic programming stereo vision algorithm for robotic applications. In *Proc. of Vision Interface*, Trois-Rivieres, Canada, May 1999.

C. Harris and M. Stephens. A combined corner and edge detector. In *Proceedings of The Fourth Alvey Vision Conference*, pages 147–151, 1988.

R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second edition, 2004.

R.I. Hartley. A linear method for reconstruction from lines and points. *Computer Vision, IEEE International Conference on*, 0:p. 882, 1995. doi: http://doi.ieeecomputersociety.org/10.1109/ICCV.1995.466843.

R.I. Hartley. Kruppa's equations derived from the fundamental matrix. *PAMI*, 19(2):133–135, February 1997a.

R.I. Hartley. Minimizing algebraic error in geometric estimation problems. In *Computer Vision, 1998. Sixth International Conference on*, pages 469 –476, jan 1998a. doi: 10.1109/ICCV.1998.710760.

R.I. Hartley and N.Y. Dano. Reconstruction from six-point sequences. In *Computer Vision and Pattern Recognition, 2000. Proceedings. IEEE Conference on*, volume 2, pages 480 –486 vol.2, 2000. doi: 10.1109/CVPR.2000.854888.

Richard Hartley and Gilles Debunne. Dualizing scene reconstruction algorithms. In *SMILE'98: Proceedings of the European Workshop on 3D Structure from Multiple Images of Large-Scale Environments*, pages 14–31, London, UK, 1998. Springer-Verlag. ISBN 3-540-65310-4.

Richard I. Hartley. Estimation of relative camera positions for uncalibrated cameras. In *ECCV '92: Proceedings of the Second European Conference on Computer Vision*, pages 579–587, London, UK, 1992. Springer-Verlag. ISBN 3-540-55426-2.

Richard I. Hartley. Lines and points in three views and the trifocal tensor. *Int. J. Comput. Vision*, 22(2):125–140, 1997b. ISSN 0920-5691. doi: http://dx.doi.org/10.1023/A:1007936012022.

Richard I. Hartley. Chirality. *Int. J. Comput. Vision*, 26(1):41–61, 1998b. ISSN 0920-5691. doi: http://dx.doi.org/10.1023/A:1007984508483.

Richard I. Hartley and Fredrik Kahl. Global optimization through rotation space search. *Int. J. Comput. Vision*, 82(1):64–79, 2009. ISSN 0920-5691. doi: http://dx.doi.org/10.1007/s11263-008-0186-9.

Richard I. Hartley and Peter Sturm. Triangulation. *Computer Vision and Image Understanding*, 68(2):146 – 157, 1997. ISSN 1077-3142.

Richard I. Hartley, Eric Hayman, Lourdes de Agapito, and Ian Reid. Camera calibration and the search for infinity. *Computer Vision, IEEE International Conference on*, 1:510, 1999. doi: http://doi.ieeecomputersociety.org/10.1109/ICCV.1999.791264.

Paul S. Heckbert. Fundamentals of texture mapping and image warping. Master's thesis, University of California at Berkeley, Berkeley, CA, USA, 1989.

Johan Hedborg, Johan Skoglund, and Michael Felsberg. KLT tracking implementation on the GPU. In *Proceedings SSBA 2007*, Linkoping, Sweden, Mars 2007.

F. Heitz and P. Bouthemy. Multimodal estimation of discontinuous optical flow using markov random fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(12):1217–1232, 1993.

Anders Heyden. Reduced multilinear constraints: Theory and experiments. *Int. J. Comput. Vision*, 30(1):5–26, 1998. ISSN 0920-5691. doi: http://dx.doi.org/10.1023/A:1008020228557.

L. Hong and G. Chen. Segment-based stereo matching using graph cuts. In *IEEE Conf. on Computer Vision and Pattern Recognition*, volume 1, pages 74–81, 2004.

Berthold K. P. Horn, H.M. Hilden, and Shariar Negahdaripour. Closed-form solution of absolute orientation using orthonormal matrices. *Journal of the Optical Society of America*, 5(7):1127–1135, 1988.

T. S. Huang and O. D. Faugeras. Some properties of the e matrix in two-view motion estimation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 11(12):1310–1312, 1989. ISSN 0162-8828. doi: http://dx.doi.org/10.1109/34.41368.

Myung Hwangbo, Jun-Sik Kim, and T. Kanade. Inertial-aided klt feature tracking for a moving camera. In *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, pages 1909 –1916, oct. 2009. doi: 10.1109/IROS.2009.5354093.

H.J. Jia and A.M. Martinez. Low-rank matrix fitting based on subspace perturbation analysis with applications to structure from motion. *PAMI*, 31(5):841–854, May 2009.

Hailin Jin, P. Favaro, and S. Soatto. Real-time feature tracking and outlier rejection with changes in illumination. In *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*, volume 1, pages 684 –689 vol.1, 2001. doi: 10.1109/ICCV.2001.937588.

Sing Bing Kang and Richard Szeliski. 3-d scene data recovery using omnidirectional multibaseline stereo. *Int. J. Comput. Vision*, 25(2):167–183, 1997. ISSN 0920-5691. doi: http://dx.doi.org/10.1023/A:1007971901577.

Sing Bing Kang, R. Szeliski, and Jinxiang Chai. Handling occlusions in dense multi-view stereo. In *CVPR '01: Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, volume 1, pages 103–110, 2001.

Michael Kazhdan, Matthew Bolitho, and Hugues Hoppe. Poisson surface reconstruction. In *SGP '06: Proceedings of the fourth Eurographics symposium on Geometry processing*, pages 61–70, Aire-la-Ville, Switzerland, Switzerland, 2006. Eurographics Association. ISBN 30905673-36-3.

R.R. Khazal and M.M.Chawla. A parallel cholesky algorithm for the solution of symmetric linear systems. *International Journal of Mathematics and Mathematical Sciences*, 2004:1315–1327, 2004.

Jae Chul Kim, Kyoung Mu Lee, Byoung Tae Choi, and Sang Uk Lee. A dense stereo matching using two-pass dynamic programming with generalized ground control points. In *IEEE Comp. Society Conf. on Computer Vision and Pattern Recognition*, volume 2, pages 1075 – 1082, June 2005.

Jun-Sik Kim, Myung Hwangbo, and T. Kanade. Realtime affine-photometric klt feature tracker on gpu in cuda framework. In *Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference on*, pages 886 –893, 2009. doi: 10.1109/ICCVW.2009.5457608.

Junhwan Kim, Vladimir Kolmogorov, and Ramin Zabih. Visual correspondence using energy minimization and mutual information. In *Intl. Conf. on Computer Vision*, pages 1033–1040, 2003.

A. Klaus, M. Sormann, and K. Karner. Segment-based stereo matching using belief propagation and a self-adapting dissimilarity measure. In *Proc. of 18th Intl. Conf. on Pattern Recognition*, volume 3, pages 15–18, 2006.

Vladimir Kolmogorov and Ramin Zabih. Computing visual correspondence with occlusions using graph cuts. In *Intl. Conf. on Computer Vision*, pages 508–515, 2001.

Vladimir Kolmogorov and Ramin Zabih. Multi-camera scene reconstruction via graph cuts. In *ECCV '02: Proceedings of the 7th European Conference on Computer Vision-Part III*, pages 82–96, London, UK, 2002. Springer-Verlag. ISBN 3-540-43746-0.

Vladimir Kolmogorov, Ramin Zabih, and Steven Gortler. Generalized multi-camera scene reconstruction using graph cuts. In *In Proceedings of the International Workshop on Energy Minimization Methods in Computer Vision and Pattern Recognition*, pages 501–516, 2003.

E. Kruppa. Zur ermittlung eines objektes aus zwei perspektiven mit innerer orientierung. *Sitzungsberichte der Mathematisch Naturwissenschaftlichen Kaiserlichen Akademie der Wissenschaften*, 122:1939–1948, 1913.

A. Kumar, A. R. Tannenbaum, and G. J. Balas. Optic flow: a curve evolution approach. *IEEE Transactions on Image Processing*, 5(4):598–610, 1996.

Kiriakos N. Kutulakos. Approximate n-view stereo. In *ECCV '00: Proceedings of the 6th European Conference on Computer Vision-Part I*, pages 67–83, London, UK, 2000. Springer-Verlag. ISBN 3-540-67685-6.

P. Labatut, J.P. Pons, and R. Keriven. Efficient multi-view reconstruction of large-scale scenes using interest points, delaunay triangulation and graph cuts. In *ICCV07*, pages 1–8, 2007.

Eric P. F. Lafortune, Sing-Choong Foo, Kenneth E. Torrance, and Donald P. Greenberg. Non-linear approximation of reflectance functions. In *SIGGRAPH '97: Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, pages 117–126, New York, NY, USA, 1997. ACM Press/Addison-Wesley Publishing Co. ISBN 0-89791-896-7. doi: http://doi.acm.org/10.1145/258734.258801.

Weilun Lao, Zhaolin Cheng, A.H. Kam, T. Tan, and A. Kassim. Focal length self-calibration based on degenerated kruppa's equations: method and evaluation. In *Image Processing, 2004. ICIP '04. 2004 International Conference on*, volume 5, pages 3391 – 3394 Vol. 5, oct. 2004. doi: 10.1109/ICIP.2004.1421842.

S. Laveau. *Geometry of a system of N cameras. Theory, estimation and applications.* PhD thesis, INRIA, 1996.

Hendrik P. A. Lensch, Michael Goesele, Jan Kautz, Wolfgang Heidrich, and Hans-Peter Seidel. Image-based reconstruction of spatially varying materials. In *Proceedings of the 12th Eurographics Workshop on Rendering Techniques*, pages 103–114, London, UK, 2001. Springer-Verlag. ISBN 3-211-83709-4.

Kenneth Levenberg. A method for the solution of certain non-linear problems in least squares. *The Quarterly of Applied Mathematics*, 2:164–168, 1944.

A. Levin and R. Szeliski. Visual odometry and map correlation. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, volume 1, pages I–611 – I–618 Vol.1, 2004. doi: 10.1109/CVPR.2004.1315088.

Tony Lindeberg. Feature detection with automatic scale selection. *Int. J. Comput. Vision*, 30 (2):79–116, 1998. ISSN 0920-5691. doi: http://dx.doi.org/10.1023/A:1008045108935.

S. Lloyd. Least squares quantization in pcm. *Information Theory, IEEE Transactions on*, 28(2):129–137, 1982. URL `http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1056489`.

M.I.A. Lourakis and A.A. Argyros. The design and implementation of a generic sparse bundle adjustment software package based on the levenberg-marquardt algorithm. Technical Report

340, Institute of Computer Science - FORTH, Heraklion, Crete, Greece, Aug. 2004. Available from http://www.ics.forth.gr/~lourakis/sba.

David G. Lowe. Object recognition from local scale-invariant features. In *ICCV '99: Proceedings of the International Conference on Computer Vision-Volume 2*, page 1150, Washington, DC, USA, 1999. IEEE Computer Society. ISBN 0-7695-0164-8.

Bruce D. Lucas and Takeo Kanade. An iterative image registration technique with an application to stereo vision. In *Proceedings of the 7th international joint conference on Artificial intelligence - Volume 2*, pages 674–679, San Francisco, CA, USA, 1981. Morgan Kaufmann Publishers Inc. URL http://portal.acm.org/citation.cfm?id=1623264.1623280.

Q.T. Luong. *matrice fundamentale et autocalibration en vision par ordinateur*. PhD thesis, Universite de Paris-Sud, Paris, France, 1992.

Q.T. Luong and O. D. Faugeras. Self-calibration of a moving camera from pointcorrespondences and fundamental matrices. *Int. J. Comput. Vision*, 22(3):261–289, 1997. ISSN 0920-5691. doi: http://dx.doi.org/10.1023/A:1007982716991.

Donald W. Marquardt. An algorithm for least-squares estimation of nonlinear parameters. *SIAM Journal on Applied Mathematics*, 11(2):431–441, 1963. doi: 10.1137/0111030. URL http://link.aip.org/link/?SMM/11/431/1.

D. Marr and E. Hildreth. Theory of edge detection. *Proceedings of the Royal Society of London. Series B, Biological Sciences,*, 207(1167):187–217, February 1980. URL http://www.jstor.org/cgi-bin/jstor/printpage/00804649/ap000006/00a00030/0.pdf?backcontext=page&dowhat=Acrobat&config=jstor&userID=8935c88a@ohsu.edu/01c0a80a6b005015e8e3&0.pdf.

B. Matei and P. Meer. Optimal rigid motion estimation and performance evaluation with bootstrap. In *Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on.*, volume 1, pages 2 vol. (xxiii+637+663), 1999. doi: 10.1109/CVPR.1999.786961.

L. Matthies and S. Shafer. Error modeling in stereo navigation. *Robotics and Automation, IEEE Journal of*, 3(3):239 –248, june 1987. ISSN 0882-4967. doi: 10.1109/JRA.1987.1087097.

E. Memin and P. Perez. Hierarchical estimation and segmentation of dense motion fields. *Intl. Journal of Computer Vision*, 46:129–155, 2002.

P.R.S. Mendonça and R. Cipolla. A simple technique for self-calibration. In *CVPR99*, pages I: 500–505, 1999.

P. Merrell, A. Akbarzadeh, Liang Wang, P. Mordohai, J.-M. Frahm, Ruigang Yang, D. Nister, and M. Pollefeys. Real-time visibility-based fusion of depth maps. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1 –8, oct. 2007. doi: 10.1109/ICCV.2007.4408984.

K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. Van Gool. A comparison of affine region detectors. *Int. J. Comput. Vision*, 65(1-2): 43–72, 2005. ISSN 0920-5691. doi: http://dx.doi.org/10.1007/s11263-005-3848-x.

Krystian Mikolajczyk and Cordelia Schmid. Scale & affine invariant interest point detectors. *Int. J. Comput. Vision*, 60(1):63–86, 2004. ISSN 0920-5691. doi: http://dx.doi.org/10.1023/B:VISI.0000027790.02288.f2.

H. Nagel. Constraints for the estimation of displacement vector fields from image sequences. In *Proc. Eighth Intl. Joint Conf. on Artificial Intelligence*, volume 2, pages 945–951, Karlsruhe, West Germany, 1983.

H. Nagel and W. Enkelmann. An investigation of smoothness constraints for the estimation of displacement vector fields from image sequences. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8:565–593, 1986.

P. Nesi. Variational approach to optical flow estimation managing discontinuities. *Image and Vision Computing*, 11(7):419–439, 1993.

Ko Nishino, Zhengyou Zhang, and Katsushi Ikeuchi. Determining reflectance parameters and illumination distribution from a sparse set of images for view-dependent image synthesis. *Computer Vision, IEEE International Conference on*, 1:599, 2001. doi: http://doi.ieeecomputersociety.org/10.1109/ICCV.2001.10077.

D. Nister. *Automatic dense reconstruction from uncalibrated video sequences*. PhD thesis, Royal Institute of Technology KTH, Stockholm, Sweden, March 2001a.

D Nister. Calibration with robust use of cheirality by quasi-affine reconstruction of the set of camera projection centres. In *IEEE International Conference on Computer Vision (ICCV 2001)*, volume 2, pages 116–123, 2001b.

David Nistér. An efficient solution to the five-point relative pose problem. *IEEE Trans. Pattern Anal. Mach. Intell.*, 26(6):756–777, 2004. ISSN 0162-8828. doi: http://dx.doi.org/10.1109/TPAMI.2004.17.

Jorge Nocedal and Stephen Wright. *Numerical optimization*. Springer, New York, 1999.

Julius Fabian Ohmer and Nicholas J. Redding. Gpu-accelerated klt tracking with monte-carlo-based feature reselection. In *Proceedings of the 2008 Digital Image Computing: Techniques and Applications*, pages 234–241, Washington, DC, USA, 2008. IEEE Computer Society. ISBN 978-0-7695-3456-5. doi: 10.1109/DICTA.2008.50. URL `http://portal.acm.org/citation.cfm?id=1469127.1470333`.

Y. Ohta and Takeo Kanade. Stereo by intra- and inter-scanline search using dynamic programming. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 7(2):139–154, March 1985.

M. Okutomi and T. Kanade. A multiple-baseline stereo. *IEEE Trans. Pattern Anal. Mach. Intell.*, 15(4):353–363, 1993. ISSN 0162-8828. doi: http://dx.doi.org/10.1109/34.206955.

J. Oliensis. Fast and accurate self-calibration. In *ICCV '99: Proceedings of the International Conference on Computer Vision-Volume 2*, page 745, Washington, DC, USA, 1999. IEEE Computer Society. ISBN 0-7695-0164-8.

C.F. Olson, L.H. Matthies, H. Schoppers, and M.W. Maimone. Robust stereo ego-motion for long distance navigation. In *Computer Vision and Pattern Recognition, 2000. Proceedings. IEEE Conference on*, volume 2, pages 453 –458 vol.2, 2000. doi: 10.1109/CVPR.2000.854879.

C.F. Olson, L.H. Matthies, M. Schoppers, and M.W. Maimone. Stereo ego-motion improvements for robust rover navigation. In *Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on*, volume 2, pages 1099 – 1104 vol.2, 2001. doi: 10.1109/ROBOT.2001.932758.

Théodore Papadopoulo and Olivier D. Faugeras. A new characterization of the trifocal tensor. In *ECCV '98: Proceedings of the 5th European Conference on Computer Vision-Volume I*, pages 109–123, London, UK, 1998. Springer-Verlag. ISBN 3-540-64569-1.

R. Phull, P. Mainali, Qiong Yang, H. Sips, and G. Lafruit. Robust low complexity feature tracking using cuda. In *Signal Processing Systems (SIPS), 2010 IEEE Workshop on*, pages 362 –367, oct. 2010. doi: 10.1109/SIPS.2010.5624818.

C. J. Poleman and T. Kanade. A paraperspective factorization method for shape and motion recovery. *IEEE Transactions on Pattern Analysis and Machine Int.*, 19:206–218, 1991.

M. Pollefeys, L. Van Gool, and A. Oosterlinck. The modulus constraint: a new constraint self-calibration. In *Pattern Recognition, 1996., Proceedings of the 13th International Conference on*, volume 1, pages 349 –353 vol.1, 25-29 1996. doi: 10.1109/ICPR.1996.546047.

M. Pollefeys, R. Koch, and L. V. Gool. Self-calibration and metric reconstruction in spite of varying and unknown instrinsic camera parameters. *Intl. Journal of Computer Vision*, 1998.

M. Pollefeys, L. V. Gool, M. Vergauwen, F. Verbiest, K. Cornelis, J. Tops, and R. Koch. Visual modeling with a hand-held camera. *Intl. Journal of Computer Vision*, 39(3):207–232, 2004.

M. Pollefeys, D. Nistér, J. M. Frahm, A. Akbarzadeh, P. Mordohai, B. Clipp, C. Engels, D. Gallup, S. J. Kim, P. Merrell, C. Salmi, S. Sinha, B. Talton, L. Wang, Q. Yang, H. Stewénius, R. Yang, G. Welch, and H. Towles. Detailed real-time urban 3d reconstruction from video. *Int. J. Comput. Vision*, 78:143–167, July 2008. ISSN 0920-5691. doi: 10.1007/s11263-007-0086-4. URL `http://portal.acm.org/citation.cfm?id=1355822.1355827`.

Marc Pollefeys and Luc Van Gool. A stratified approach to metric self-calibration. In *CVPR '97: Proceedings of the 1997 Conference on Computer Vision and Pattern Recognition (CVPR '97)*, page 407, Washington, DC, USA, 1997. IEEE Computer Society. ISBN 0-8186-7822-4.

Marc Pollefeys and Luc Van Gool. Stratified self-calibration with the modulus constraint. *IEEE Trans. Pattern Anal. Mach. Intell.*, 21(8):707–724, 1999. ISSN 0162-8828. doi: http://dx.doi.org/10.1109/34.784285.

Marc Pollefeys, Luc Van Gool, Maarten Vergauwen, Kurt Cornelis, Frank Verbiest, and Jan Tops. Video-to-3d. In *Proceedings of Photogrammetric Computer Vision*, 2002a.

Marc Pollefeys, Frank Verbiest, and Luc J. Van Gool. Surviving dominant planes in uncalibrated structure and motion recovery. In *ECCV '02: Proceedings of the 7th European Conference on Computer Vision-Part II*, pages 837–851, London, UK, 2002b. Springer-Verlag. ISBN 3-540-43744-4.

Jean Ponce. On computing metric upgrades of projective reconstructions under the rectangular pixel assumption. In *SMILE '00: Revised Papers from Second European Workshop on 3D Structure from Multiple Images of Large-Scale Environments*, pages 52–67, London, UK, 2001. Springer-Verlag. ISBN 3-540-41845-8.

Jean-Philippe Pons, Renaud Keriven, and Olivier Faugeras. Multi-view stereo reconstruction and scene flow estimation with a global image-based matching score. *Int. J. Comput. Vision*, 72(2):179–193, 2007. ISSN 0920-5691. doi: http://dx.doi.org/10.1007/s11263-006-8671-5.

Simant Prakoonwit and Ralph Benjamin. 3d surface point and wireframe reconstruction from multiview photographic images. *Image Vision Comput.*, 25(9):1509–1518, 2007. ISSN 0262-8856. doi: http://dx.doi.org/10.1016/j.imavis.2006.12.019.

M. et al Proesmans. Determination of optical flow and its discontinuities using non-linear diffusion. In *Proc. European Conference on Computer Vision*, volume 801, pages 295–304, 1994.

Long Quan. Invariants of six points and projective reconstruction from three uncalibrated images. *IEEE Trans. Pattern Anal. Mach. Intell.*, 17(1):34–46, 1995. ISSN 0162-8828. doi: http://dx.doi.org/10.1109/34.368154.

Rahul Raguram, Jan-Michael Frahm, and Marc Pollefeys. A comparative analysis of ransac techniques leading to adaptive real-time random sample consensus. In *Proceedings of the 10th European Conference on Computer Vision: Part II*, pages 500–513, Berlin, Heidelberg, 2008. Springer-Verlag. ISBN 978-3-540-88685-3. doi: 10.1007/978-3-540-88688-4_37.

J. Repko and M. Pollefeys. 3d models from extended uncalibrated video sequences: addressing key-frame selection and projective drift. In *3-D Digital Imaging and Modeling, 2005. 3DIM 2005. Fifth International Conference on*, pages 150 – 157, june 2005. doi: 10.1109/3DIM.2005.4.

P.D. Sampson. Fitting conic sections to 'very scattered' data: An iterarive refinement of the bookstein algorithm. *Computer Vision, Graphics, and Image Processing*, 18(1):97–108, January 1982.

Yoichi Sato, Mark D. Wheeler, and Katsushi Ikeuchi. Object shape and reflectance modeling from observation. In *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '97, pages 379–387, New York, NY, USA, 1997. ACM Press/Addison-Wesley Publishing Co. ISBN 0-89791-896-7. doi: http://dx.doi.org/10.1145/258734.258885. URL `http://dx.doi.org/10.1145/258734.258885`.

D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *Int. Journal of Computer Vision*, 47:7–42, 2002.

C. Schnörr. Bewegungssegmentation von bildfolgen durch die minimierung konvexer nichtquadratischer funktionale. In W.G. Kropatsch and H. Bischof, editors, *Mustererkennung 1994*, volume 5 of *Informatik Xpress*, pages 178–185. Technische Universität Wien, September 1994.

Steven M. Seitz and Charles R. Dyer. Photorealistic scene reconstruction by voxel coloring. In *CVPR '97: Proceedings of the 1997 Conference on Computer Vision and Pattern Recognition (CVPR '97)*, page 1067, Washington, DC, USA, 1997. IEEE Computer Society. ISBN 0-8186-7822-4.

Steven M. Seitz, Brian Curless, James Diebel, Daniel Scharstein, and Richard Szeliski. A comparison and evaluation of multi-view stereo reconstruction algorithms. In *CVPR '06: Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 519–528, Washington, DC, USA, 2006. IEEE Computer Society. ISBN 0-7695-2597-0. doi: http://dx.doi.org/10.1109/CVPR.2006.19.

A. Shashua and M. Werman. Trilinearity of three perspective views and its associated tensor. In *ICCV '95: Proceedings of the Fifth International Conference on Computer Vision*, page 920, Washington, DC, USA, 1995. IEEE Computer Society. ISBN 0-8186-7042-8.

Amnon Shashua. Algebraic functions for recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 17(8):779–789, 1995. ISSN 0162-8828. doi: http://dx.doi.org/10.1109/34.400567.

Jianbo Shi and C. Tomasi. Good features to track. In *Computer Vision and Pattern Recognition, 1994. Proceedings CVPR '94., 1994 IEEE Computer Society Conference on*, pages 593 –600, jun 1994. doi: 10.1109/CVPR.1994.323794.

D. Shulman and J. Herve. Regularization of discontinuous flow fields. In *Proc. Workshop on Visual Motion*, pages 81–86, Irvine, CA, 1989.

Sudipta Sinha, Jan-Michael Frahm, Marc Pollefeys, and Yakup Genc. Feature tracking and matching in video using programmable graphics hardware. *Machine Vision and Applications*, 22:1–11, 2007.

G. Slabaugh, B. Culbertson, T. Malzbender, and R. Schafe. A survey of methods for volumetric scene reconstruction from photographs. In *International Workshop on Volume Graphics*, 2001.

Minas E. Spetsakis and John Aloimonos. Structure from motion using line correspondences. *International Journal of Computer Vision*, 4(3):171–183, June 1990. doi: 10.1007/BF00054994. URL `http://dx.doi.org/10.1007/BF00054994`.

Peter F. Sturm and Bill Triggs. A factorization based algorithm for multi-image projective structure and motion. In *ECCV '96: Proceedings of the 4th European Conference on Computer Vision-Volume II*, pages 709–720, London, UK, 1996. Springer-Verlag. ISBN 3540611231. URL `http://portal.acm.org/citation.cfm?id=645310.649025`.

Jian Sun, Nan-Ning Zheng, Senior Member, and Heung-Yeung Shum. Stereo matching using belief propagation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(7), July 2003.

Richard Szeliski. A multi-view approach to motion and stereo. *CVPR '99: Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, 1:1157, 1999. ISSN 1063-6919. doi: http://doi.ieeecomputersociety.org/10.1109/CVPR.1999.786933.

Y. Takaoka, Y. Kida, S. Kagami, H. Mizoguchi, and T. Kanade. 3d map building for a humanoid robot by using visual odometry. In *Systems, Man and Cybernetics, 2004 IEEE International Conference on*, volume 5, pages 4444 – 4449 vol.5, oct. 2004. doi: 10.1109/ICSMC.2004.1401231.

T. Thormahlen, N. Hasler, M. Wand, and H.-P. Seidel. Merging of feature tracks for camera motion estimation from video. In *Visual Media Production (CVMP 2008), 5th European Conference on*, pages 1 –8, nov. 2008.

M. Tistarelli. Multiple constraints for optical flow. In *Proc. European Conf. on Computer Vision*, volume 800, pages 61–70, 1994.

Carlo Tomasi and Takeo Kanade. Detection and tracking of point features. Technical Report CMU-CS-91-132, Carnegie Mellon University, April 1991. URL `http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.45.5770`.

Carlo Tomasi and Takeo Kanade. Shape and motion from image streams under orthography: a factorization method. *Int. J. Comput. Vision*, 9(2):137–154, 1992. ISSN 0920-5691. doi: http://dx.doi.org/10.1007/BF00129684.

T. Tommasini, A. Fusiello, E. Trucco, and V. Roberto. Making good features track better. In *Computer Vision and Pattern Recognition, 1998. Proceedings. 1998 IEEE Computer Society Conference on*, pages 178 –183, jun 1998. doi: 10.1109/CVPR.1998.698606.

P. H. S. Torr. Bayesian model estimation and selection for epipolar geometry and generic manifold fitting. *Int. J. Comput. Vision*, 50:35–61, October 2002. ISSN 0920-5691. doi: 10.1023/A:1020224303087. URL `http://portal.acm.org.www.lib.ncsu.edu:2048/citation.cfm?id=598435.598525`.

P. H. S. Torr and A. Zisserman. Robust parameterization and computation of the trifocal tensor. *Image and Vision Computing*, 15(8):591 – 605, 1997. ISSN 0262-8856. doi: DOI: 10.1016/S0262-8856(97)00010-3. British Machine Vision Conference.

P. H. S. Torr and A. Zisserman. Mlesac: a new robust estimator with application to estimating image geometry. *Comput. Vis. Image Underst.*, 78(1):138–156, 2000. ISSN 1077-3142. doi: http://dx.doi.org/10.1006/cviu.1999.0832.

P.H.S. Torr. *Outlier Detection and Motion Segmentation*. PhD thesis, University of Oxford, 1995.

P.H.S. Torr. An assessment of information criteria for motion model selection. In *Computer Vision and Pattern Recognition, 1997. Proceedings., 1997 IEEE Computer Society Conference on*, pages 47 –52, jun 1997. doi: 10.1109/CVPR.1997.609296.

K. E. Torrance and E. M. Sparrow. Theory for off-specular reflection from roughened surfaces. *J. Opt. Soc. Am.*, 57:pp.1105–1112, 1967.

Adrien Treuille, Aaron Hertzmann, and Steven M. Seitz. Example-based stereo with general brdfs. In *Proc. European Conf. Computer Vision (ECCV '04)*, pages pp. 457–469, 2004.

B. Triggs. Matching constraints and the joint image. In *ICCV '95: Proceedings of the Fifth International Conference on Computer Vision*, page 338, Washington, DC, USA, 1995. IEEE Computer Society. ISBN 0-8186-7042-8.

B. Triggs. Autocalibration and the absolute quadric. In *CVPR '97: Proceedings of the 1997 Conference on Computer Vision and Pattern Recognition (CVPR '97)*, page 609, Washington, DC, USA, 1997. IEEE Computer Society. ISBN 0-8186-7822-4.

Bill Triggs, P. McLauchlan, Richard Hartley, and A. Fitzgibbon. Bundle adjustment – a modern synthesis. In B. Triggs, A. Zisserman, and R. Szeliski, editors, *Vision Algorithms: Theory and Practice*, volume 1883 of *Lecture Notes in Computer Science*, pages 298–372. Springer-Verlag, 2000. URL `http://lear.inrialpes.fr/pubs/2000/TMHF00`.

Shinji Umeyama. Least-squares estimation of transformation parameters between two point patterns. *IEEE Trans. Pattern Anal. Mach. Intell.*, 13:376–380, April 1991. ISSN 0162-8828. doi: http://dx.doi.org/10.1109/34.88573. URL `http://dx.doi.org/10.1109/34.88573`.

S. Uras. A computational approach to motion perception. *Biological Cybernetics*, 60:79–87, 1988.

G. van Meerbergen, M. Vergauwen, M. Pollefeys, and L.J. Van Gool. A hierarchical stereo algorithm using dynamic programming. In *SMBV01*, pages xx–yy, 2001.

Thierry Vieville and D. Lingrand. Using singular displacements for uncalibrated monocular visual systems. In *ECCV '96: Proceedings of the 4th European Conference on Computer Vision-Volume II*, pages 207–216, London, UK, 1996. Springer-Verlag. ISBN 3-540-61123-1.

G. Vogiatzis, P. H. S. Torr, and R. Cipolla. Multi-view stereo via volumetric graph-cuts. In *CVPR '05: Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 2*, pages 391–398, Washington, DC, USA, 2005. IEEE Computer Society. ISBN 0-7695-2372-2. doi: http://dx.doi.org/10.1109/CVPR.2005.238.

Aiqi Wang, Tianshuang Qiu, and Longtan Shao. A simple method of radial distortion correction with centre of distortion estimation. *J. Math. Imaging Vis.*, 35:165–172, November 2009. ISSN 0924-9907. doi: 10.1007/s10851-009-0162-1.

Gregory J. Ward. Measuring and modeling anisotropic reflection. *SIGGRAPH Comput. Graph.*, 26(2):265–272, 1992. ISSN 0097-8930. doi: http://doi.acm.org/10.1145/142920.134078.

J. Weickert and C. Schnorr. Variational optic flow computation with a spatio-temporal smoothness constraint. *Journal of Mathematical Imaging and Vision*, 14:245–255, 2001.

J. Weng, P. Cohen, and N. Rebibo. Motion and structure estimation from stereo image sequences. *Robotics and Automation, IEEE Transactions on*, 8(3):362 –382, jun 1992a. ISSN 1042-296X. doi: 10.1109/70.143354.

Juyang Weng, Thomas S. Huang, and Narendra Ahuja. Motion and structure from line correspondences; closed-form solution, uniqueness, and optimization. *IEEE Trans. Pattern Anal. Mach. Intell.*, 14(3):318–336, 1992b. ISSN 0162-8828. doi: http://dx.doi.org/10.1109/34.120327.

J. Woetzel and R. Koch. Real-time multi-stereo depth estimation on gpu with approximative discontinuity handling. In *First European Conf. on Visual Media Production*, pages 245–254, 2004.

Q. Yang, L. Wang, R. Yang, H. Stewénius, and D.Nistér. Stereo matching with color-weighted correlation, hierarchical belief propagation and occlusion handling. In *CVPR*, pages 2347–2354, 2006a.

Q. Yang, L. Wang, R. Yang, S. Wang, M. Liao, and D. Nistér. Real-time global stereo matching using hierarchical belief propagation. In *BMVC*, pages 989–998. British Machine Vision Association, 2006b. doi: http://www.macs.hw.ac.uk/bmvc2006/proceedings.html.

Ruigang Yang and M. Pollefeys. Multi-resolution real-time stereo on commodity graphics hardware. In *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, volume 1, pages I–211 – I–217 vol.1, june 2003. doi: 10.1109/CVPR.2003.1211356.

Yizhou Yu, Paul Debevec, Jitendra Malik, and Tim Hawkins. Inverse global illumination: recovering reflectance models of real scenes from photographs. In *SIGGRAPH '99: Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pages 215–224, New York, NY, USA, 1999. ACM Press/Addison-Wesley Publishing Co. ISBN 0-201-48560-5. doi: http://doi.acm.org/10.1145/311535.311559.

C. Zach, D. Gallup, and J.-M. Frahm. Fast gain-adaptive klt tracking on the gpu. In *Computer Vision and Pattern Recognition Workshops, 2008. CVPRW '08. IEEE Computer Society Conference on*, pages 1 –7, june 2008. doi: 10.1109/CVPRW.2008.4563089.

C. Zeller. *Projective, Affine and Euclidean Calibration in Computer Vision and the Application of Three Dimensional Perception.* PhD thesis, RobotVis Group, INRIA Sophia-Antipolis, 1996.

Z. Zhang and O.D. Faugeras. Estimation of displacements from two 3d frames obtained from stereo. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 14(12):1141 –1156, dec 1992. ISSN 0162-8828. doi: 10.1109/34.177380.