Geometric and Photometric Constraints: Motion and Structure from Three Views

by

Gideon P. Stein

B.Sc., Technion, Israel Institute of Technology (1990)
M.S., Massachusetts Institute of Technology (1993)

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

at the

Massachusetts Institute of Technology

June 1998

© Massachusetts Institute of Technology 1998

Signature of Author .....
Department of Electrical Engineering and Computer Science
February 27,1998

Certified by .....
W. Eric L. Grimson
Professor of Computer Science and Engineering
Thesis Supervisor

Certified by .....
Amnon Shashua
Senior Lecturer of Computer Science,
Hebrew University of Jerusalem, Israel
Thesis Supervisor

Accepted by .....
Arthur C. Smith
Chairman, Departmental Committee on Graduate Students

Geometric and Photometric Constraints: Motion and Structure from Three Views

by

Gideon P. Stein

Submitted to the
Department of Electrical Engineering and Computer Science
on February 27,1998, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

Abstract

This thesis addresses the problem of recovering both the camera motion and scene structure given a sequence of images obtained from a camera moving through a rigid environment. The core of the thesis is the *model-based brightness constraint*, a new method for computing the camera motion and dense scene structure from three views. We combine photometric constraints together with geometric constraints to directly compute the camera motion from the image brightness derivatives (spatial and temporal). This type of method is dubbed a *direct method* because it does not require prior computation of optical flow or point correspondences. We describe the advantages and limitations of this method and present results with real images.

We apply the same framework to the analysis of image sequences obtained from a moving stereo rig. We show that the motion of the stereo rig can be computed directly from the image derivatives using linear equations. Correspondences are not required. The motion estimates between stereo pairs enable us to combine depth maps from all the pairs in the sequence to form an extended scene reconstruction and we show results from a real image sequence.

The tensor brightness constraint is intimately connected to 3D reconstruction from line correspondences. For line correspondences, the linear solution is degenerate when all the lines come from a Linear Line Complex (i.e. when all the lines intersect a single line in space). This configuration is quite common in practice and in particular, it arises in the context of the model-based brightness constraint. We show a method for overcoming the degeneracy of the general case and show some experimental results.

To accurately disambiguate between camera translation and rotation a wide field-of-view is required. For wide angle lenses, non-linear lens distortion can be a significant factor and affect the accuracy of motion and structure estimates. We present a new method for determining the lens distortion parameters using geometric constraints on a set of point correspondences in multiple views. There is no need to know the 3D location of the points.

Thesis Co-Supervisor: Prof. W. Eric L. Grimson

Thesis Co-Supervisor: Dr. Amnon Shashua

Acknowledgments

My graduate studies here at the MIT Artificial Intelligence Laboratory have been most exciting, stimulating and fun. I would like to thank all those people who have helped to make it so: the professors whose doors are always open, my fellow students with whom I have had stimulating discussions and fun times, and the warm and friendly support staff who have always tried to make our life here that much easier.

In particular, I would like to thank my thesis advisors Eric Grimson and Amnon Shashua, and my committee members Berthold Horn and Tommy Poggio, for their support throughout my PhD research, and without whose contributions and guidance this project would not have flourished. They provided both the exceptional freedom to pursue my ideas, and the experience and knowledge to help keep my work grounded and in context. I would also like to thank Berthold Horn, John Wyatt and Chris Atkeson for their support and guidance during the early part of my graduate studies here at MIT.

The support staff here at the AI Lab do a great job of keeping this place running. In particular Jill Fekete, Laurel Simmons, Bruce Walton and Ron Wiken have all gone out of their way to help me put together the hardware and software for the experiments presented here.

I would like to thank my fellow students and lab members, William (Sandy) Wells, Polina Golland, Raquel Romano, Lily Lee, Greg Galperin, Carlin Vieri, J. P. Mellor, Holly Yanco, Oded Maron, Marina Miella, Carl de Marcken, Charles Isbell, Liana Lorigo, Tian Kappur, Erik Miller and many others for all their help and friendship over the years. In particular I would like to thank Jeremy De Bonet and Chris Stauffer for collaborating with me on the vision software libraries, and for being there to bounce ideas back and forth.

I would like to thank my parents and family, and all my friends here in the Boston area for their encouragement and support throughout my graduate studies here at MIT.

This report describes research done at the Artificial Intelligence Laboratory of the Massachusetts Institute of Technology. Support for this research was provided in part by the Advanced Research Projects Agency of the Department of Defense under Office of Naval Research contract N00014-94-01-0994. G.P.S. is supported by Alphatech/ DARPA "Automatic Target Recognition" project 95009-5381 and TASC/ DARPA "MSTAR Match Module" project J-08011-S95042, ONR contracts N00014-95-1-0600 "A Trainable Modular Vision System", and DARPA/ONR contracts N00014-97-1-0363 and N00014-94-01-0994 "Intelligent Integration of Information". G.P.S. would also like to acknowledge the support provided by a fellowship from National Science Fellowship during the early years of graduate studies.

Contents

1 Introduction **15**

1.1 Model-based Brightness Constraints: Motion and Structure from Three Views . . . 17
1.2 Combining Motion and Stereo . . . 19
1.3 Degeneracy of the Linear Solution to the Trilinear Tensor in the Case of a Linear Line Complex . . . 21
1.4 Lens Distortion Calibration . . . 22
1.5 An Example Application . . . 23
1.6 Overview of the Thesis . . . 23

2 Mathematical Background **29**

2.1 Notation . . . 29
2.1.1 Points and Lines . . . 29
2.1.2 Tensor Notation . . . 30
2.2 Imaging Geometry and Motion Models . . . 31
2.2.1 Perspective and Projective Cameras . . . 31
2.2.2 Motion Model for the Projective Case . . . 32
2.2.3 Motion Model for Calibrated Cameras and Small Rotation . . . 35
2.2.4 Small Motion Model . . . 36
2.2.5 Weak Perspective, Orthographic and Affine . . . 37
2.3 The Geometry of Two Views: the Epipolar Constraint . . . 40
2.4 The Geometry of Three Views - the Trilinear Tensor . . . 42
2.4.1 Tensor Slices and Contractions . . . 44
2.4.2 Tensor Admissibility Constraints . . . 45
2.5 Photometric Constraints - the Optical Flow Constraint Equation . . . 46
2.5.1 The Aperture Problem . . . 47
2.5.2 The Optical Flow Constraint as a Point-Line Correspondence 48
2.6 Summary . . . 49

3 Model-based Brightness Constraints: Direct Estimation of Motion and Structure **51**

3.1 Introduction . . . 51
3.1.1 Related Work . . . 52
3.1.2 Overview of this Chapter . . . 56
3.2 The Tensor Brightness Constraint: the Projective Model . . . 57
3.3 The Small Rotation Model with Calibrated Cameras . . . 58
3.4 The 15-Parameter, Small Motion Model . . . 60

3.5 Pure Translation . . . 61
3.5.1 The Singular Case of Collinear Motion . . . 62
3.6 Solving for Translation and Rotation . . . 62
3.6.1 Solution Using the Rank=2 Constraint. . . . 63
3.6.2 The Algorithm for Finding Motion Parameters . . . 64
3.6.3 Solve as a Nonlinear Optimization . . . 65
3.6.4 Reduce to Pure Translation by Planar Alignment . . . 65
3.7 Implementation Details . . . 66
3.7.1 Computing the Depth, Smoothing and Interpolation . . . 66
3.7.2 Iterative Refinement . . . 66
3.7.3 Image Warping . . . 68
3.7.4 Coarse-to-Fine Processing . . . 69
3.7.5 Field of View . . . 69
3.7.6 Final Touches - Recompute the Depth Keeping the Motion Constant . . . 75
3.8 Summary of Theoretical Results and Future Work . . . 75
3.8.1 Future Work - Collinear Motion . . . 76

4 Direct Estimation of Motion and Structure - Experiments 79
4.1 Overview . . . 79
4.2 Pure Translation . . . 79
4.2.1 Experiment 1: Accuracy of Heading Estimates for Pure Translation . . . 79
4.2.2 Experiment 2: Collinear Motion with Pure Translation . . . 80
4.2.3 Experiment 3: Comparison with Optical Flow Techniques. . . . 85
4.3 Translation and Rotation . . . 92
4.3.1 Experiment 4: Translation and Rotation (Reanalysis of Data from Stein & Shashua, CVPR97) . . . 92
4.3.2 Experiment 5: Motion Heading Estimation with Rotation . . . 96
4.3.3 Experiment 6: Euclidean Structure Estimation in the Presence of Rotation. . . . 102
4.4 Experiments with a Hand-held Camera . . . 102
4.4.1 Experiment 7: Euclidean Structure Estimation Using a Hand-held Camera . . . 102
4.4.2 Experiment 8: Outdoor Scenes - Church Wall . . . 106
4.4.3 Experiment 9: Outdoor Scene 2 - Side Entrance . . . 106
4.5 Summary of Experimental Results . . . 110

5 Direct Estimation of Motion and Extended Scene Structure from a Moving Stereo Rig 117
5.1 Introduction . . . 117
5.1.1 Overview of the Chapter . . . 118
5.2 Geometric Constraints for Combined Motion and Stereo . . . 119
5.3 The Brightness Constraint of a Moving Stereo Rig . . . 122
5.4 Implementation Details . . . 123

5.4.1 Calibration . . . 123
5.4.2 Computing Depth . . . 124
5.4.3 Coarse to Fine Processing and Iterative Refinement . . . 124
5.5 Experiments . . . 125
5.5.1 Experimental Details . . . 125
5.5.2 Extended Scene Reconstruction . . . 125
5.6 Discussion and Future Work . . . 130

6 On Degeneracy of Linear Reconstruction from Three Views: the Linear Line Complex 133
6.1 Introduction . . . 133
6.1.1 Notations . . . 134
6.2 Linear Line Complex Scene Structure . . . 135
6.2.1 Algorithm for Recovering Structure and Motion in the Case of a LLC . . . 139
6.3 Experiments . . . 140
6.3.1 The Experimental Procedure . . . 140
6.3.2 Determining the LLC . . . 140
6.3.3 Recovering Motion and Structure . . . 142
6.4 Summary . . . 143

7 Lens Distortion Calibration Using Point Correspondences 147
7.1 Introduction . . . 147
7.1.1 The Two Image Method . . . 147
7.1.2 The Three Image Method . . . 148
7.2 Related Work . . . 148
7.3 Mathematical Background . . . 149
7.3.1 The Epipolar Constraint . . . 149
7.3.2 The Trilinear Tensor Constraint . . . 149
7.3.3 The Lens Distortion Model . . . 150
7.4 The Algorithm . . . 150
7.5 Simulation . . . 151
7.6 Experiments with Real Images . . . 151
7.6.1 Experimental Details . . . 151
7.6.2 Experiment 1: Finding the Lens Distortion Parameters . . . 155
7.6.3 Experiment 2: Testing the Radial Distortion Parameters in the Projective Domain . . . 156
7.6.4 Experiment 3: Euclidean Reconstruction . . . 158
7.7 Discussion . . . 160

8 Summary and Future Work 161
8.1 Future Work . . . 162

List of Figures

1-1 Three input images (a,b,c) used for 3D reconstruction 19
1-2 3D rendering of the depth map in Figure 1-1d 20
1-3 A few real world examples of a Linear Line Complex 22
1-4 Stereo input images and resulting depth map 24
1-5 3D renderings of the scene 25
1-6 Euclidean reconstruction of the extended scene 26

2-1 Tensor notation 30
2-2 The geometry of three views 41
2-3 The aperture problem: an example 48

3-1 The three input images and the estimated depth map 53
3-2 Detail of optical flow computed for Figure 3-1 54
3-3 3D rendering of the estimated depth map from images in Figure 3-1 . 55
3-4 Motion flow field due to rotation around the Y axis 71
3-5 The X and Y components of the motion field due a rotation around the Y axis 72
3-6 The shape of the cost function for various fields of view 73
3-7 Local depth error fixed by a second pass down the pyramid performing 'depth-only' computation 75

4-1 Two views of the optical bench and the camera mounted on the motion stage 81
4-2 Three input images and the estimated depth map 82
4-3 Heading estimate. Pure translation was assumed. 4.9_{mm} lens 83
4-4 Heading estimate. Pure translation was assumed. 8.5_{mm} lens 84
4-5 Schematic diagram of the pure translation experiment 85
4-6 Input images and depth map with collinear motion 86
4-7 Heading estimates: collinear and non-collinear motion 87
4-8 Input images and depth map: collinear vertical motion 88
4-9 Optical flow computed for Figure 3-1 90
4-10 Detail of optical flow computed for Figure 3-1 91
4-11 Optical flow computed for Figure 4-12 93
4-12 Three input images pure translation. Another image with rotation . . 94
4-13 The estimated depth map and 3D rendering of the surface 95
4-14 Estimated rotation as a function of true rotation 97
4-15 Experiment to test motion heading estimation in the presence of rotation 98
4-16 Heading estimates from a moving camera 99

4-17 Heading estimate error: the difference between heading estimate and motion stage reading . . . 100
4-18 Rotation estimates from a camera translating with some added rotation 101
4-19 Depth map estimated after computing both translation and rotation . 101
4-20 A simple scene used to test 3D Euclidean reconstruction: a cube . . . 103
4-21 3D rendering of the recovered scene . . . 104
4-22 3D rendering of the recovered cube . . . 105
4-23 A simple scene used to test 3D Euclidean reconstruction . . . 107
4-24 3D renderings of the depth map in Figure 4-23d . . . 108
4-25 3D renderings of the depth map in Figure 4-23f . . . 109
4-26 3D renderings of the depth map in Figure 4-23e . . . 110
4-27 Three input views from a hand-held camera and the resulting depth map 111
4-28 *3D renderings of the depth map in Figure 4-27d.* . . . 112
4-29 Detail of church wall . . . 113
4-30 *3D Euclidean renderings of the depth map in Figure 4-29d.* . . . 114
4-31 Three input views of a side entrance to a building. . . . 115
4-32 3D renderings of the depth map in Figure 4-31d . . . 116

5-1 The three input images (a,b,c) . . . 120
5-2 3D renderings of the scene. Very distant surfaces have been removed . 121
5-3 Schematic overhead view of the 'fly-through' scene . . . 126
5-4 Estimated Rotation and translation . . . 127
5-5 The 14th image in the sequence and the depth map . . . 127
5-6 The 28th image in the sequence and the depth map . . . 128
5-7 The 49th image in the sequence and the depth map . . . 129
5-8 Euclidean reconstruction of the extended scene . . . 131

6-1 Typical urban indoor and outdoor scenes . . . 135
6-2 Figure to accompany Theorem 1 . . . 136
6-3 The three input images . . . 141
6-4 The smallest singular values of the estimation matrix $\mathbf{W}$, for a degenerate and non degenerate set of lines . . . 144
6-5 The test lines from Image 2 and Image 3 are reprojected back into Image 1 using the recovered tensors . . . 145

7-1 Simulated camera setup . . . 152
7-2 Simulation results . . . 153
7-3 The images used for calibration . . . 154
7-4 The tensor reprojection error as a function of the value of the radial distortion K_1 . . . 157
7-5 The height of the reconstructed 3D points from the best fit plane . . 159

8-1 Euclidean reconstruction of the extended scene created by aligning four separate reconstructions into one coordinate system . . . 164

List of Tables

4.1 Rotation estimates (in degrees) from the images in Figure 4-23 106

6.1 Values of the error cost function for estimating the LLC 142

7.1 Radial distortion parameter K_1 obtained using various ordering of the images 156

Chapter 1

Introduction

Motion gives us a strong perception of the three dimensional world. As we walk down the street or through the woods we sense the space around us. We sense that we are moving. We know if the path ahead is clear. We know when to duck to avoid hitting a low overhead branch. We might notice an opening in the wall to our left. How do we explain this?

How can we get a robot to do the same? More precisely, given a sequence of images obtained from a camera moving through a rigid environment, how do we recover both the camera motion and scene structure? That problem is the focus of this thesis.

The geometry of multiple views of a 3D scene is well understood. There exist geometric constraints, which relate corresponding points and lines in multiple views to the camera geometry. These constraints take the form of the trilinear tensor equations for three views, and the epipolar constraints for two views. Given a set of feature correspondences we can recover the camera geometry, and in particular camera motion (or the relative position of multiple cameras). But finding correspondences is a hard problem, and the features must be recovered accurately in order to correctly recover camera motion.

Feature correspondence, whether optical flow or discrete features, is based in some form or another, on the constant brightness assumption. That is, we assume the same feature in two views share some photometric properties. In it's strictest form, the *constant brightness constraint*, assumes that the brightness of the corresponding point does not change between views. Alternatively, we might look at some function of the brightness, possibly a nonlinear function such as brightness edges. The constant brightness constraint is a good approximation for many surfaces in the real world, especially if the motion is small. It is well known that the constant brightness constraint is not strong enough to give us true correspondences. Local measurements, for example, cannot give 'optical flow' but only *normal flow*, that is, the image flow estimates in the direction of the image brightness gradient. This is a particular problem in scenes with long, nearly straight, edges. We will show some examples later in the thesis.

In this thesis we have developed the *model-based brightness constraints*, where we combine geometric motion models with the brightness constraint. This provides us with a stronger constraint that can be used for direct estimation of structure and motion. We can threreby bypass the correspondence problem, and recover the camera motion directly from the image brightness values. This results in accurate

motion estimates, and good 3D reconstruction, in very challenging scenes.

What happens if we know some but not all of the camera geometry? How can this be used in the geometric constraints? We have studied these questions in the context of stereo image sequences, where we know the displacement between the two cameras, but do not know the motion of the stereo rig as a whole. In this case multi-view geometric constraints provide an elegant and simple way to combine both motion and stereo. This is true if we have feature correspondences and can also be used for direct methods. The latter can be implemented using the framework of model-based constraints. We show as an example, how we can easily recover the motion of the stereo rig directly from image brightness values. One can then use the images from both pairs combined to compute a dense depth map at every camera location. Using the motion estimates between stereo pairs we bring all the depth reconstructions into one coordinate frame.

The model-based brightness constraints are intimately connected to 3D reconstruction from line correspondences. We prove that for line correspondences, the linear solution is degenerate when all the lines come from a Linear Line Complex (i.e. when all the lines intersect a single line in space). This configuration is shown to be quite common in practice and in particular, it arises in the context of the model-based brightness constraints. We show a method for overcoming the degeneracy of the general case and show some experimental results.

In order to correctly disambiguate between rotation and translation, we need to use a wide field-of-view. This requires wide angle lenses, which typically have large amounts of non-linear lens distortion. We investigated how lens distortion affects the geometric constraints. In the process, we developed a new method for lens distortion calibration using a set of point correspondences in multiple views. A calibration object is not required, and there is no need to know the 3D location of the points. This makes the method very practical to use.

Where Versus What

In this thesis we focus on determining *where* objects are in the scene relative to the camera and how the camera is moving relative to the scene. This is what we call the perception of 'space'. To accurately recover camera motion and distance to an object requires a perspective camera model and wide-angle view. (In Section 3.7.5 we show that a wide field of view is required to disambiguate camera rotation and translation).

The other part of the structure from motion problem is the perception of the *shape* of an object. We can think of this as answering the question *what* is the object. The main requirements are reasonable image resolution, reasonable camera baseline and camera geometry that disambiguates rotation and translation. While not the main focus of this thesis, we show some good shape reconstruction results (for example Figure 1-2). Good results have also been achieved using orthographic or weak perspective camera models and an object centered coordinate system [79, 35, 81, 75, 54]. Using the orthographic camera model, the distance to the object cannot be recovered.

The rest of the chapter

In the following sections we will briefly expand on the four main topics of the thesis: the model-based brightness constraints, combining motion and stereo, the linear degeneracy of the linear line complex, and finally, lens distortion calibration. In Section 1.5 we provide an example application: extended scene reconstruction from a stereo image sequence. In Section 1.6 we describe the organization of the rest of the thesis.

1.1 Model-based Brightness Constraints: Motion and Structure from Three Views

The core of the thesis is a new method for computing the camera motion and dense scene structure from three views in an image sequence. This is done without first computing point correspondences or optical flow. Instead we combine the photometric constraints between pairs of views together with the geometric constraints between three views to directly compute the camera motion. After computing the motion we can compute dense structure, in other words, the depth to every pixel in the image. This type of method is dubbed a *direct method* because it does not require prior computation of optical flow.

This is in contrast to previous approaches which in general first find point correspondences between consecutive images using photometric (i.e. brightness) constraints. After computing the correspondences the camera motion between two or three views and the 3D structure of the scene are recovered using geometric constraints. Here we include both 'optical flow' based methods [42, 43, 28, 55, 77] and feature based methods [79, 16, 65, 67, 4]. The main problem with these approaches is that computing the correspondence is in itself a difficult and error prone process. Dense correspondence methods such as 'optical flow' suffer from the aperture problem and it is necessary to assume smoothness in the flow [33, 43, 49]. Sparse methods try to find matching features such as lines or corner points. It is still an open subject of research as to what makes a good feature[11, 83, 16]. Some points and lines in the image do not correspond to any real feature in 3D (e.g. the occluding contour of a cylinder or of the head in Figure 1-1). This issue is discussed in more depth in the introduction to Chapter 3. These errors in computing the correspondences get carried through to the motion and structure estimation stage.

Our method avoids these problems by directly computing the motion from image brightness measurements without first computing correspondences, and by using information from all the pixels in the image to compute the motion estimates. The wide area of support for the computation means that the motion estimates are less affected by regions where the brightness constraints are violated, such as on occluding contours.

We now provide the key ideas behind the new method. The *optical flow constraint equation* [33] provides a matching constraint between each point in one image, and a line passing through the corresponding point in the second image. In other words,

for every point in one image it gives us the equation of a line along which the corresponding point must lie in the second image. By *point* we refer to every pixel in the image which has a none zero brightness derivative. The equation is given in terms of the spatio-temporal derivatives of the image brightness, and the line is parallel to the iso-brightness contour in the first image. The 'optical flow constraint equation' is a first order approximation and assumes that the image motion (and hence typically, the camera motion) is very small.

There are no geometric constraints on point-line correspondences between two views, as can be seen using the following reasoning. A point in the image defines a line in space. A line in the second image defines a plane in space. A line and a plane always intersect. Therefore, given a set of points in one image and a set of corresponding lines in the second image, for every camera geometry there exists a valid 3D interpretation and no constraint on the geometry exists. We must therefore use three view geometry, where a point in one view, and lines through the corresponding points in two other views, provide a constraint, which can be written in the form of the *trilinear tensor* [57, 27, 69]. The 27 coefficients of the 'trilinear tensor' encapsulate the camera motions and the internal parameters of the camera such as the focal length. The new method can therefore be viewed as a necessary extension of the 'direct methods' of Horn & Weldon [34] from two views (one motion) to three views (two motions).

By combining the optical flow constraint equation with the geometric model of the 'trilinear tensor' we obtain the *tensor brightness constraint* [74] that describes the relationship between the spatio-temporal brightness derivatives at each pixel in the image, with the camera parameters modeled by the 27 coefficients of the trilinear tensor. This 'tensor brightness constraint' provides one linear equation per pixel in the image which results in a highly over-constrained set of equations. The 'tensor brightness constraint' is valid for the most general case (projective) where the cameras undergo general motion and we do not know the internal camera parameters which might vary from frame to frame.

We then proceed through a hierarchy of simpler motion models first by assuming calibrated cameras and small rotations, then by assuming the Longuet-Higgins & Prazdny small motion model [42], where we assume also that the camera translation in depth is small relative to the depth of the objects in the scene. Thus we have a hierarchy of *model-based brightness constraints*. In particular the Longuet-Higgins & Prazdny model leads to a 15-parameter brightness constraint, where the 15 parameters are bilinear combinations of the unknown translations and rotations. One constraint equation for each pixel leads to a set of linear homogeneous equations in the 15 intermediate parameters.

By avoiding the need to explicitly detect feature points we can use information from areas where gradient information is weak such as shadow edges. This advantage is highlighted in Figure 1-1. The images are of low contrast without many distinctive features. The strongest 'features' are the occluding contours of the head which do not correspond to a real feature in space (as the camera moves different points on the head form the occluding contour). Nevertheless, our method produces accurate motion and depth estimates (Figure 1-2). The full quantitative and qualitative results

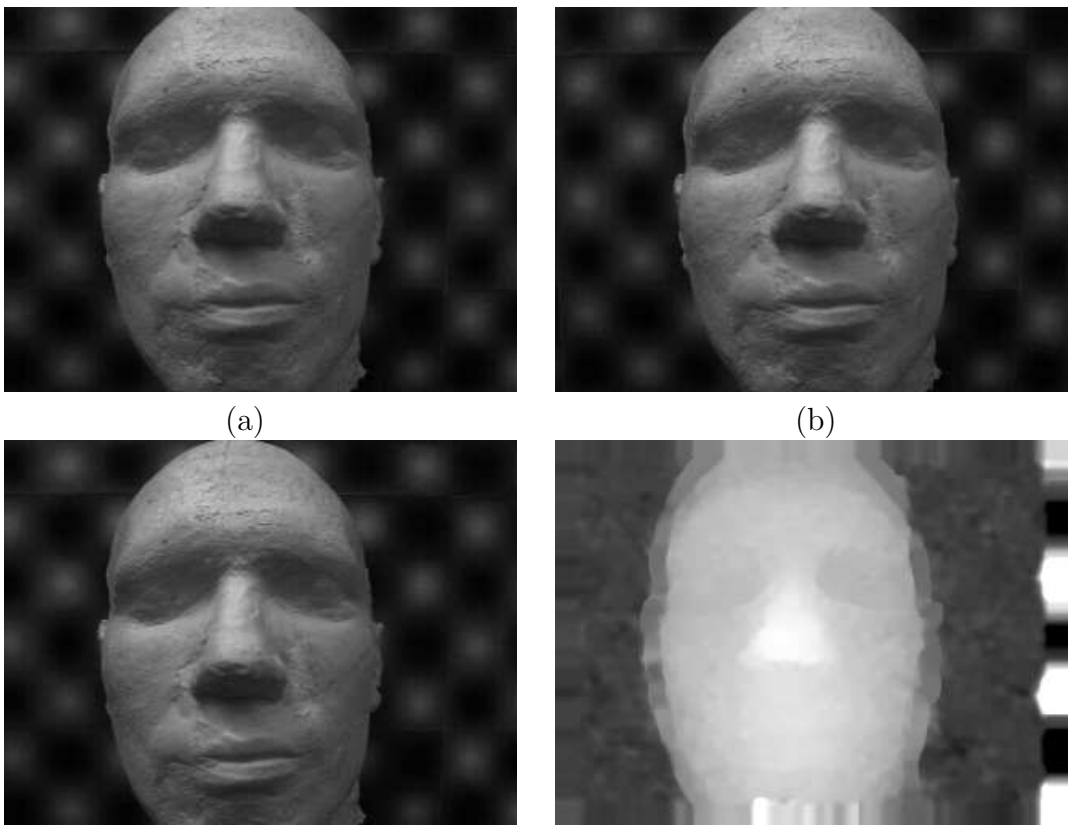


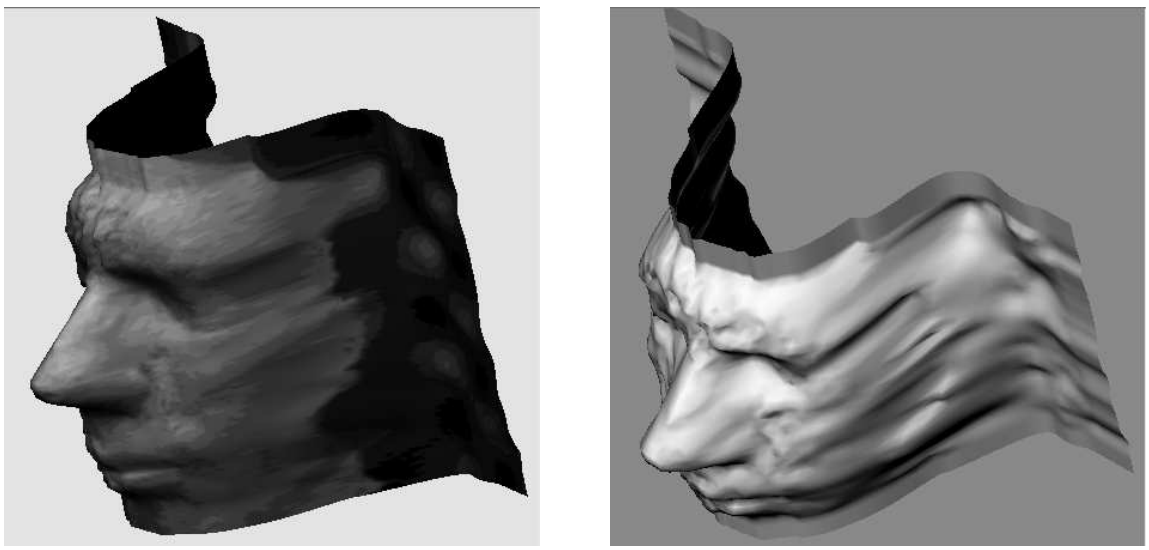
Figure 1-1: *The three input images (a,b,c) used for 3D reconstruction. Motion* $(a \rightarrow b)$ *was a horizontal translation with a* 1.8^o *rotation around the* Y *axis. Motion* $(a \rightarrow c)$ *was a vertical translation. This is a challenging set of images because the the texture is smooth with low contrast. The strongest edge 'features' appear along the occluding contour of the head but this edge does not in fact correspond to a real feature in the world. (d) The estimated depth map. The estimated rotation was* 1.62^o*. There are of course errors around the boundary of the image where there is no overlap between the images.*

are described in Chapter 4.

1.2 Combining Motion and Stereo

Stereo and motion together form a powerful combination [23, 91] with a variety of applications from camera motion estimation to extended scene reconstruction. A stereo system (with two or more cameras) can give accurate depth estimates but only from one view point. To build a complete model of a scene we would like to combine together multiple 3D views. One solution is to use a moving stereo rig (two cameras mounted onto the same rigid frame).

The standard approach to reconstruction from a moving stereo rig [39, 91, 85, 16] has been to first compute a depth map from each pair (or compute the depth of a



(a) (b)

Figure 1-2: *3D rendering of the depth map in Figure 1-1d. (b) texture map removed to show detail.*

set of features points), together with a confidence measure for each depth estimate. Next, one registers the stereo pair with the current 3D model to compute the motion, and finally, one updates the model (using a Kalman filter or a batch method). In contrast, Navab et al [51] combine motion and stereo for the reconstruction of motion and structure from line correspondences. Given two distinct non-parallel lines in two stereo image pairs they obtain a set of 8 linear equations in the 6 motion parameters. The method requires accurate tracking of straight lines in an image.

In Chapter 5 we describe a new method for directly estimating the motion of a stereo rig . We show that given two stereo pairs one can compute the motion of the stereo rig directly from the image derivatives (spatial and temporal). Correspondences are not required.

Directly computing the motion of the stereo rig simplifies the problem of registering all the depth reconstructions into one coordinate frame. Our method gives good estimates of the camera motion between frames. There is no need to explicitly compute confidence values for the depth estimates, since the causes for error are taken into account implicitly in the equations. By concatenating these motion estimates, one can bring all the depth maps into one coordinate frame.

The core of the method is to apply the multi-view geometric constraints (Chapter 2) to a situation where some, but not all, of the camera motions are known. We first develope the equations for feature correspondences in the case of calibarted cameras. In order to avoid the need to find the correspondences we use then use the 15-parameter model-based brightness constraints (Chapter 3). We apply the model-based brightness constraint to the case where the motion between one pair of views is known. In general, the model-based brightness constraint equations are bilinear in the motion parameters but if one of the motions is known they result in a set of linear equations for the second motion parameters. There is one equation for each point in

the image and 6 unknowns (the translation and rotation parameters), which results in a highly over-constrained, and very stable, computation.

The use of a stereo pair rather than a single image sequence (monocular structure from motion) has further advantages. The method is stable in the case of collinear motion, in the case of pure rotation and also for planar objects. There is no scale ambiguity since all motions are relative to the known baseline length [23]. We present an example of a 3D reconstruction from a sequence of 50 stereo image pairs.

1.3 Degeneracy of the Linear Solution to the Trilinear Tensor in the Case of a Linear Line Complex

The tensor brightness constraint combines both photometric and geometric constraints. The result is a system of linear homogeneous equations (one equation per pixel) in the 27 unknown coefficients of the trilinear tensor. A unique linear solution would be abtainable if the rank of the linear system was 26 but we find that the rank is 23. Is this a purely geometric phenomenom or is it because of the photmetric constraints? We show in Chapter 6 that the degeneracy is geometric in origin and that it is connected to reconstruction from line correspondences.

A set of line correspondences among three views gives rise to a linear set of equations where the unknowns are the camera and motion parameters represented by the $3 \times 3 \times 3$ trilinear tensor. The image measurements are the coordinates of corresponding lines. The 27 coefficients of the trilinear tensor are not independent and there exist 9 non-linear constraints that reduce the tensor's degrees of freedom to 18.

In general this system of equations has a unique linear solution but for particular 3D line configurations in space this system of equations is degenerate. There are some *critical line configurations* from which a unique solution is not possible (see [44, 9, 50]). There are other configurations where a linear solution is degenerate but the additional non-linear constraints lead to a unique solution. When the sample of features is taken from a configuration of lines that have a common intersection, known as a *Linear Line Complex* (LLC), then the rank of the linear system reduces from 26 (in the general case) to 23 — yet, there exists a *unique* solution for the tensor when the non-linear admissibility constraints are accounted for.

The line configuration described by an LLC, rather than being some obscure case, is in fact quite typical. It arises, as a particular example, in the case of the tensor brightness constraint. In also include many urban scenes such as in Figure 1-3. In Figure 1-3a the common intersecting line is the edge of the building. All horizontal lines on the two faces of the building meet the edge in the image plane, and the vertical lines meet the edge at infinity. Note also that the vertical line representing the lamp-post also meets the edge of the building (at infinity) and is thereby included in the LLC configuration. This leaves very few lines (the sidewalk and the oblique line of the lamp-post) not part of the LLC. The common line in Figure 1-3b is the edge of the book-case leaning on the wall. All vertical lines in the scene intersect that

(a) (b)

Figure 1-3: *Typical urban indoor and outdoor scenes. Most of the 3D lines in each scene intersect a single line in space and therefore form a Linear Line Complex. (a) The all lines on the two faces of the building and the vertical lines of the lamp post intersect the edge (corner) of the building (possibly at infinity). (b) All vertical lines in the scene and all the horizontal lines on the bookcase and file cabinet intersect the edge of the bookcase leaning on the wall. Linear methods for 3D reconstruction from lines will fail for these scenes although unique solutions exist using non-linear methods.*

line at infinity and the horizontal lines of the file cabinet, bookcase and the arm chair meet it in the image plane. Again, very few lines in the scene do not belong to the LLC (the horizontal line segments of the box near the far end wall, the far arm of the armchair and two horizontal line segments attached to the book-case).

An investigation into these degeneracies and their remedy is important in the context of this thesis and the tensor brightness constraint. It is also important in the general context of 3D reconstruction from line correspondences.

1.4 Lens Distortion Calibration

The geometric constraints between two views (the epipolar constraint) and between three views (the trilinear tensors) assume a perfect pinhole camera. For medium and wide angle lenses, non-linear lens distortion and in particular radial lens distortion, can be a significant factor with errors of $10-100 pixels$ near the edges of the image [71]. These are typically the lenses used when performing image based 3D reconstruction of large objects or in a confined space, and which are used in the experiments throughout this thesis.

The investigation of the effects of lens distortion on the trilinear tensor and epipolar constraints leads to a new method for calibration of lens distortion. In this method only point correspondences between either two or three views are required. There is no need to know the 3D location of the points, the camera locations, or other internal parameters of the camera such as the focal length. In fact one can use the same point correspondence data which is to be used for subsequent vision tasks such as 3D

reconstruction. The method is therefore suitable as an online preprocessing stage.

The simulation results show that the geometric configuration of the cameras is important for the success of the calibration. The cameras must be far apart with large angles between the camera views (vergence angles). When the angles between the camera views are wide (20^o or more) then the lens distortion parameters can be recovered accurately. When the angle between views is small, as would typically be the case in nearby frames of a video sequence, the geometric constraints hold even with lens distortion. The recovered motion and structure however, will have some non-projective distortion. If we neglect the radial distortion, large planar surfaces do not appear flat in the reconstruction, and the 3D reconstructions from multiple stereo pairs do not align correctly.

1.5 An Example Application

A preview of some of the results from Chapter 5 will provide a flavor for some of the possible applications. In Chapter 5 we apply these techniques to the problem of recovering motion and extended 3D structure from a stereo image sequence. A sequence of 50 stereo pairs was used. Figures 1-4a,c show the first image pair from the sequence. Figure 1-4b shows the image from the same camera as Figure 1-4a but taken at the next time step. This sequence was taken in a laboratory but a similar set of images might be obtained from a low flying helicopter in a confined space.

From the three images (Figures 1-4a,b,c) we recover the *camera motion* and a *dense depth map* (Figure 1-4d). For a mobile robot or helicopter, the camera motion estimates can be used to augment an internal navigation system and the depth map can be used for obstacle avoidance. Edges in the depth map (Figure 1-4e) are places of *depth discontinuities*. In a traffic scene, these are places where a car or pedestrian might appear without warning. In a search-and-rescue operation these indicate places in the image where things are hidden from view. We can use this information to guide the search.

Figures 1-5a,b show 3D renderings of the reconstructed scene. The camera path shows that the camera traveled in a horizontal plane twice the height of the trees. A single image triplet can provide us with only a limited view of the scene. We can use the recovered camera motion as a way to combine multiple 3D views together to form an *extended 3D scene reconstruction* (Figure 1-6). This is an example of a *3D mosaic*. It provides a compact representation of all the information in the 50 input images and can thus be viewed as an *image compression scheme*.

1.6 Overview of the Thesis

The introduction in each chapter discusses its place in the context of the thesis and in the context of related and previous work. In the next chapter (Chapter 2) we summarize the common mathematical background needed in the rest of the thesis. We describe the projective camera model and its relationship to the perspective and

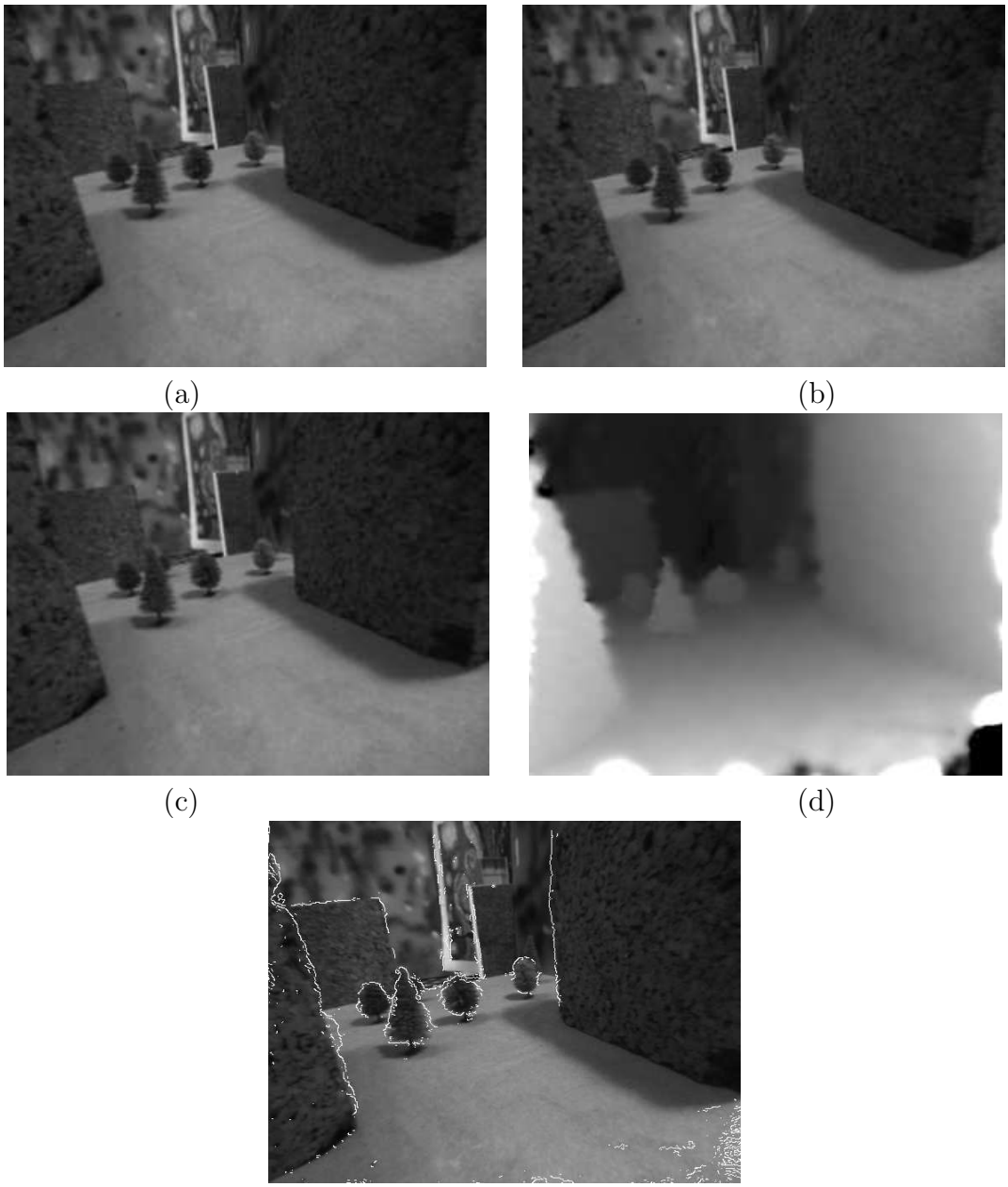


Figure 1-4: *The three input images (a,b,c). Image (a) and (c) are the first stereo pair in the sequence with the cameras displaced vertically. Image (b) is the image from camera (a) taken at the next time instance. (d) the estimated depth map. (e) Edges in the depth map identify occluding contours.*

(a) (b)

Figure 1-5: *3D renderings of the scene. Very distant surfaces have been removed. The spheres show the motion of the stereo rig in the whole sequence. The large sphere is the the location of the camera when this image was taken. Note in view (b) that the ground plane is correctly found to be flat and at right angles to the two cork walls.*

orthographic cameras. We then develop the constraint equations among point and line correspondences in two and three views. These are well known results based mainly on the work of Faugeras [16] for two views, and on Shashua's work on the trilinear tensor constraint for 3 views [57].

Chapters 3 through 5 should be read in sequence. Chapter 3 combines photogrammetric constraints in the form of the optical flow constraint equation [33] with the geometric constraint of the trilinear tensor to develop the tensor brightness constraint. We then develop simpler 'model-based brightness constraints' and describe a method for recovering structure and motion directly from image gradients. Section 3.2 derives the projective form of the constraint. Those uncomfortable with projective geometry can skim through that section and focus more on Section 3.4 and those that follow. As we have noted earlier, the optical flow constraint equation is only valid for infinitesimally small motions. In practice, this requires that the image motion be on the order of one pixel. In Section 3.7, among other details, we show how using a coarse-to-fine implementation, our method can handle displacements averaging up to 50 pixels for 640×480 resolution images. In Chapter 4 we show quantitative and qualitative results with real images.

In Chapter 5 we use the same three-view geometric constraints as a way of combining information from multiple stereo images obtained by a moving stereo rig. In addition to augmenting depth-from-stereo with additional depth information obtained from motion, this method greatly simplifies the process of bringing the depth maps from multiple stereo pairs into one coordinate frame. An example application is shown for direct estimation of motion and extended scene structure from a stereo image sequence.

In Chapter 6 we return to the trilinear tensor constraint and prove that for line correspondences the linear solution is degenerate when all the lines come from a *Linear Line Complex* (i.e. when all the lines intersect a single line in space). This

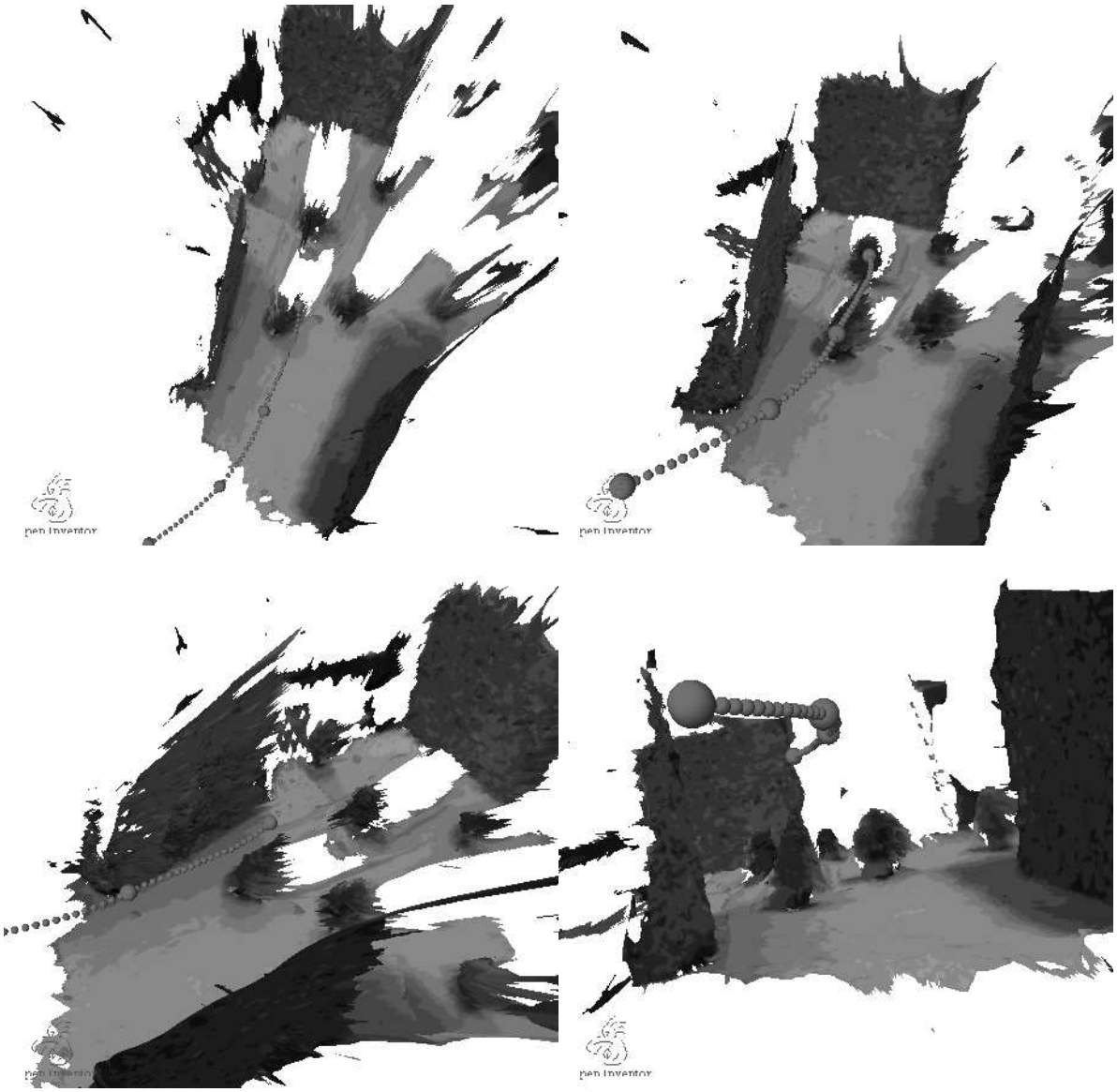


Figure 1-6: *Euclidean reconstruction of the extended scene created by aligning four separate reconstructions into one coordinate system using the recovered camera motion. The small spheres show the camera path. The large spheres show the camera positions of the four Euclidean reconstructions. There are still 'holes' in the 3D view because there are parts of the world which are not viewable from any image.*

configuration is shown to be quite common in practice, and in particular it arises in the context of the tensor brightness constraint (Chapter 3). We show a method for overcoming the degeneracy and experimental results.

The geometric constraints developed in Chapter 2 assume a perfect pinhole camera. Non-linear lens distortion and in particular radial lens distortion is a significant factor in medium and wide angel lenses and, can affect the accuracy of motion and structure estimates. Chapter 7 describes the non-linear lens distortion model and describes a new method for calibrating the lens distortion parameters using only point correspondences among multiple views. We verify the new method with simulations and experiments with real images.

Chapter 2

Mathematical Background

In this chapter we summarize the common mathematical background needed for the rest of the thesis. We first review the perspective camera model and the geometric constraint equations among point and line correspondences in two and three views. Then we review the photometric constraints, which take the form of the optical flow constraint equation.

This thesis uses results from projective geometry. A very readable introduction to the subject of projective geometry is given in the book by Young [89]. A modern book dealing more specifically with the application of projective geometry to computer vision is [16]. The appendix of [48] provides a concise summary of the basic material.

2.1 Notation

We will use uppercase bold to denote matrices (e.g. $\mathbf{A}$). Vectors representing 3D points which will be upper case and not bold. Other vectors and scalars will be in lower case.

We will use ψ to denote the first camera image or view. In the text it might also be called Image 1. The second and third views (camera images) will be called Image 2 and Image 3 respectively and denoted ψ' and ψ'' where the apostrophe $'$, and double apostrophe $''$, is used to denote variables associated with Image 2 and Image 3 respectively. Following this notation a point in Image 1, Image 2 and Image 3 will be denoted p, p' and p'' respectively. Translation from camera position of Image 1 to the camera position of Image 2 will be denoted t', translation from camera position of Image 1 to the camera position of Image 3 will be denoted t'', and so forth.

2.1.1 Points and Lines

Following [16] we use projective coordinates. A point in three-dimensional space is represented by a point in the projective space $\mathcal{P}^3$, $P \cong (X, Y, Z, W)^\top$. ($\cong$ indicates an equality up to a scale factor). A point in the image plane is represented by a point in $\mathcal{P}^2$, $p \cong (x, y, w)^\top$. The dual of a point in $\mathcal{P}^2$, the line s, is also represented by a set of three homogeneous coordinates: $s \cong (s_1, s_2, s_3)^\top$. A point p lies on a line s iff:

$$s^\top p = 0.$$

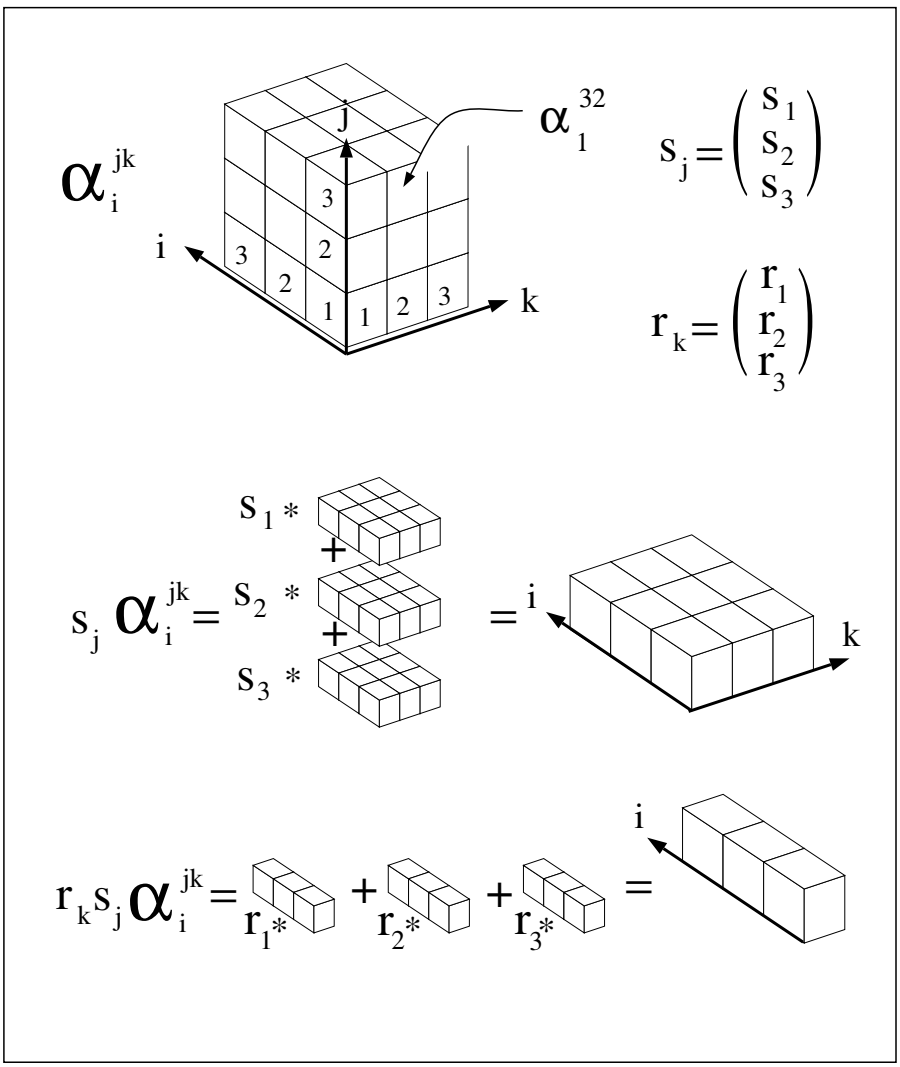


Figure 2-1: *Tensor Notation*

2.1.2 Tensor Notation

We will occasionally use tensorial notations. We use the covariant-contravariant summation convention: a point is an object whose coordinates are specified with superscripts, i.e., $p^i = (p^1, p^2, ...)$. These are called contravariant vectors. An element in the dual space (representing hyperplanes — lines in $\mathcal{P}^2$), is called a covariant vector and is represented by subscripts, i.e., $s_j = (s_1, s_2, ...)$. Indices repeated in covariant and contravariant forms are summed over, i.e., $p^i s_i = p^1 s_1 + p^2 s_2 + ... + p^n s_n$. This is known as a contraction. When using this notation we will use the indices i, j, k to denote features such as points and lines in Image 1, Image 2 and Image 3 respectively.

Figure 2-1 shows how the contraction of a 1-valence tensor (a vector) with a 3-valence tensor produces a 2-valence tensor (a matrix). A contraction of a vector with a matrix produces a vector and finally, the contraction of two 1-valence tensors is simply the familiar vector dot product.

An outer-product of two 1-valence tensors (vectors), $a_i b^j$, is a 2-valence tensor (matrix) c_i^j whose i, j entries are $a_i b^j$ — note that in matrix form $\mathbf{C} = ba^\top$.

2.2 Imaging Geometry and Motion Models

2.2.1 Perspective and Projective Cameras

We use the classic pinhole camera model, details of which can be found in [31]. A real lens has non-linear geometric distortion. We will defer the issue of lens distortion to Chapter 7 and for now assume the lens distortion, if significant, can be calibrated and compensated for.

Since the image plane is finite we can assign the value 1 to the third homogeneous coordinate thus: $p \cong (x, y, 1)^\top$. The perspective projection of the point P onto the image can be written as a linear projective equation:

$$p \cong \mathbf{M}\mathcal{P}\mathbf{K}P. \tag{2.1}$$

The 4×4 matrix $\mathbf{K}$ is the Euclidean transformation from the world coordinate system to a 3D coordinate system centered at the optical center of the camera:

$$\mathbf{K} = \begin{pmatrix} \mathbf{R} & t \\ 0_3^\top & 1 \end{pmatrix}. \tag{2.2}$$

The orthogonal rotation matrix $\mathbf{R}$ and the translation vector t are called the *extrinsic* camera parameters. The 3×4 matrix $\mathcal{P}$ is the projection matrix from the 3D camera coordinate system to the *normalized camera coordinate plane*:

$$\mathcal{P} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}. \tag{2.3}$$

The *normalized camera coordinate system* locates the image plane parallel to the retinal plane at a distance of one unit from the optical center. The upper triangular matrix $\mathbf{M}$ is the affine transformation from the normalized camera coordinate plane to the retinal coordinate system:

$$\mathbf{M} = \begin{pmatrix} -f_x & f_x \cot\theta & c_x \\ 0 & \frac{-f_y}{\sin\theta} & c_y \\ 0 & 0 & 1 \end{pmatrix} \tag{2.4}$$

The values of $\mathbf{M}$ are called the *intrinsic* camera parameters. Physically, f_x and f_y are the focal length (to be precise, the principle distance) where we allow for different focal lengths in the x and y directions to take into account different spatial sampling of the image plane, and (c_x, c_y) is the principle point (i.e. the point of intersection of the optical axis of the lens and the image plane). The angle θ is the angle between the image x and y axes. With CCD cameras $\theta = 90^o$, in which case:

$$\mathbf{M} = \begin{pmatrix} -f_x & 0 & c_x \\ 0 & -f_y & c_y \\ 0 & 0 & 1 \end{pmatrix} \tag{2.5}$$

If we know the values of the matrix $\mathbf{M}$ we refer to the camera as a *calibrated camera*. We can then move from image coordinates (p) to the normalized image coordinates (p_n) by premultiplying:

$$p_n \cong \mathbf{M}^{-1}p.$$

One often combines the matrices $\mathbf{M}\mathcal{P}\mathbf{K}$ into one 3×4 matrix:

$$p \cong \tilde{\mathbf{A}}P. \tag{2.6}$$

Note that not all 3×4 matrices are valid camera matrices but if we allow for pictures of pictures of pictures then there are no constraints on the projection matrix $\tilde{\mathbf{A}}$. This case, where there are no constraints on the matrix $\tilde{\mathbf{A}}$ is called the *projective camera*. It is clear that for any invertible matrix $\mathbf{T}$:

$$p \cong \tilde{\mathbf{A}}\mathbf{T}^{-1}\mathbf{T}P, \tag{2.7}$$

where the choice of $\mathbf{T}$ is the choice of the projective coordinate frame.

We will often split the matrix $\tilde{\mathbf{A}}$ into a 3×3 matrix (e.g. $\mathbf{A}$) and a 3×1 vector (e.g. t) which we will write as:

$$\tilde{\mathbf{A}} = [\mathbf{A}; t].$$

2.2.2 Motion Model for the Projective Case

We will present here some of the important results from [61]. Let p be the projection a 3D point P in Image 1. If we align the world coordinate system with the camera coordinate system then the (X, Y, Z) coordinates of the point $P = (X, Y, Z, 1)^\top$ are given by:

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = Z\mathbf{M}^{-1}p$$

where $\mathbf{M}$ is the matrix representing the internal parameters of Camera 1. Likewise, for Image 2:

$$\begin{pmatrix} X' \\ Y' \\ Z' \end{pmatrix} = Z'\mathbf{M'}^{-1}p$$

Let $\mathbf{R}$', and t' be the relative translation and rotation between Image 1 and Image 2. Then:

$$p' \cong \mathbf{M}'\mathbf{R}'\mathbf{M}^{-1}p + \frac{1}{Z}\mathbf{M}'t'. \tag{2.8}$$

If we parameterize the plane π by its normal vector n and distance its d_π to the origin of Camera 1 then the point P lies on the plane π iff:

$$\frac{n^\top (Z\mathbf{M}^{-1}p)}{d_\pi} = 1. \tag{2.9}$$

Multiplying the right hand term of (2.8) by (2.9) leads to:

$$\begin{aligned} p' &\cong \mathbf{M}'\mathbf{R}'\mathbf{M}^{-1}p + \frac{1}{Z}\frac{1}{d_\pi}\mathbf{M}'t'\left(n^\top Z\mathbf{M}^{-1}p\right) \\ &\cong \mathbf{M}'\left(\mathbf{R}' + \frac{1}{d_\pi}t'n^\top\right)\mathbf{M}^{-1}p = \mathbf{A}p \end{aligned} \tag{2.10}$$

where $\mathbf{A}$ is the homography matrix of the plane π. We note that given the corresponding images of 4 points on the plane π, or three points and the corresponding epipoles, it is possible to find the homography matrix $\mathbf{A}$ up to a scale factor.

Let $e' = \mathbf{M}'t'$, then for a general point P:

$$\begin{aligned} p' &\cong \mathbf{M}'\mathbf{R}'\mathbf{M}^{-1}p + \frac{1}{Z}e' \\ &\cong \mathbf{A}p - \frac{1}{d_\pi}e'n^\top\mathbf{M}^{-1}p + \frac{1}{Z}e' \\ &\cong \mathbf{A}p + \left[\frac{1}{Z} - \frac{n^\top(\mathbf{M}^{-1}p)}{d_\pi}\right]e' \\ &\cong \mathbf{A}p + \left[\frac{d_\pi - n^\top(Z\mathbf{M}^{-1}p)}{Zd_\pi}\right]e' \\ &\cong \mathbf{A}p + \left[\frac{d}{Zd_\pi}\right]e' \\ &\cong \mathbf{A}p + \rho e', \end{aligned} \tag{2.11}$$

where $d = d_\pi - n^\top(Z\mathbf{M}^{-1}p)$ is the perpendicular signed distance from the point P to the plane π.

The important thing to note in the above equation is that the term in square brackets, which we have denoted ρ, is invariant to the camera parameters of the second view (i.e. it does not include any of the terms $\mathbf{R}'$, t', and $\mathbf{M}'$). The homography matrix $\mathbf{A}$, and the vector e', are each given up to a scale factor. We can set the scale of e' such that for a given point P_0, not on the plane π,:

$$p'_0 \cong \mathbf{A}p + e'.$$

With this scaling of e', for any point P:

$$\rho = \frac{Z_0}{Z}\frac{d}{d_0}$$

where d_0 is the perpendicular signed distance from the point P_0 to the plane π. When the plane π is the plane at infinity then $d, d_0 \to \infty$ and therefore $\rho \to \frac{Z_0}{Z}$ is the inverse depth, and also, since $d_\pi \to \infty$, $\mathbf{A} \to \mathbf{M}'\mathbf{R}'\mathbf{M}^{-1}$.

We now show how equation (2.10) implies that any homography matrix $\mathbf{A}_2$ can be described by any other homography matrix $\mathbf{A}_\pi$, the vector e', a scalar d and a vector $\hat{n}$:

$$\mathbf{A}_2 = \mathbf{A}_\pi + \frac{1}{d}e'\hat{n}^\top. \tag{2.12}$$

If $\mathbf{A}_\pi$ is the homography due to the plane at infinity, i.e.:

$$\mathbf{A}_\infty \cong \mathbf{M}'\mathbf{R}'\mathbf{M}^{-1},$$

then:

$$\begin{aligned} \mathbf{A}_2 &\cong \mathbf{A}_\infty + \frac{1}{d_2} e' n_2^\top \mathbf{M}^{-1} \\ &\cong \mathbf{A}_\infty + \frac{1}{d_2} e' \hat{n}_2^\top . \end{aligned} \tag{2.13}$$

Geometrically $\hat{n}_2^\top = n_2^\top \mathbf{M}^{-1}$ is the normal vector to the plane *in the camera coordinate system*. Now let $\mathbf{A}_\pi$ be the homography due to an arbitrary plane:

$$\mathbf{A}_\pi \cong \mathbf{A}_\infty + \frac{1}{d_\pi} e' \hat{n}_\pi^\top$$

and since:

$$\mathbf{A}_2 \cong \mathbf{A}_\infty + \frac{1}{d_2} e' \hat{n}_2^\top$$

we have:

$$\mathbf{A}_2 \cong \mathbf{A}_\pi + \frac{1}{d} e' \hat{n}^\top \tag{2.14}$$

for some scalar d and some vector $\hat{n}$.

For the geometric interpretation of $\hat{n}$, let L be the projection of the intersection of the planes $\mathbf{A}_\pi$ and $\mathbf{A}_2$, onto Image 1. For all points $P \in L$:

$$p' \cong \mathbf{A}_2 p \cong \mathbf{A}_\pi p.$$

Therefore, from equation (2.14) $e'\hat{n}^\top p = 0$, and $\hat{n}$ must be the projection of L in Image 1.

The Epipole

For the case where we have chosen the projective frame such that:

$$p \cong [\mathbf{I}; 0]P,$$

the vector $e' = M't'$ is the *epipole*, the projection of the camera center of the first camera in Image 2. This can be derived directly from equation (2.1) by substituting $P = (0, 0, 0, 1)^\top$:

$$e' \cong \mathbf{M}'\mathcal{P} \begin{pmatrix} \mathbf{R}' & t' \\ 0_3^\top & 1 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} = \mathbf{M}'t'.$$

2.2.3 Motion Model for Calibrated Cameras and Small Rotation

Calibrated Cameras

In the case of calibrated cameras we know the internal camera parameters and we can move from image coordinates (p) to the normalized image coordinates (p_n) by premultiplying:

$$p_n \cong \mathbf{M}^{-1} p.$$

Notation: when we talk about calibrated cameras we will assume that we have moved to normalized image coordinates. We will drop the n subscript and refer to normalized image points in as p, p' and p''.

Thus for calibrated cameras the projections of a point $\hat{P}$, in some arbitrary world coordinate system, into Image 1, Image 2 and Image 3 are simply:

$$\begin{array}{rcccl} p & \cong & \mathcal{P}\mathbf{K}\hat{P} & \cong & [\hat{\mathbf{R}};\hat{t}]\hat{P} \\ p' & \cong & \mathcal{P}\mathbf{K}'\hat{P} & \cong & [\hat{\mathbf{R}}';\hat{t}']\hat{P} \\ p'' & \cong & \mathcal{P}\mathbf{K}''\hat{P} & \cong & [\hat{\mathbf{R}}'';\hat{t}'']\hat{P}. \end{array}$$

By using a rigid transformation of the world coordinates of the form:

$$P = \mathbf{T}^{-1}\hat{P}$$

where:

$$\mathbf{T} = \begin{pmatrix} \hat{\mathbf{R}}^{-1} & -\hat{\mathbf{R}}^{-1}\hat{t} \\ 0 & 1 \end{pmatrix},$$

we align the world coordinate system with the coordinate of the first camera. Then:

$$\begin{array}{rcl} p & \cong & [\hat{\mathbf{R}};\hat{t}]\hat{P} \\ & \cong & [\hat{\mathbf{R}};\hat{t}]\mathbf{T}\mathbf{T}^{-1}\hat{P} \\ & \cong & [\mathbf{I};0]P \end{array}$$

and

$$\begin{array}{rcl} p' & \cong & [\mathbf{R}';t']P \\ p'' & \cong & [\mathbf{R}'';t'']P, \end{array}$$

where

$$[\mathbf{R}';t'] = [\hat{\mathbf{R}}';\hat{t}']\mathbf{T}$$

and

$$[\mathbf{R}'';t''] = [\hat{\mathbf{R}}'';\hat{t}'']\mathbf{T}.$$

In this coordinate system:

$$p \cong \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \cong \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} \cong \begin{pmatrix} \frac{X}{Z} \\ \frac{Y}{Z} \\ 1 \end{pmatrix}$$

and

$$P \cong \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \cong \begin{pmatrix} \frac{X}{Z} \\ \frac{Y}{Z} \\ 1 \\ k \end{pmatrix} \cong \begin{pmatrix} x \\ y \\ 1 \\ k \end{pmatrix}$$

where $k = \frac{1}{Z}$ is the inverse depth. The motion model for calibrated cameras is therefore:

$$\begin{array}{rcl} p' & \cong & \mathbf{R}'p + kt' \\ p'' & \cong & \mathbf{R}''p + kt''. \end{array} \tag{2.15}$$

Small Rotation

If we also assume small angle rotations and can therefore make the approximations:

$$\cos(\theta) \approx 1, \quad \sin(\theta) \approx \theta,$$

the rotation matrices $\mathbf{R}'$ and $\mathbf{R}''$ become:

$$\begin{array}{rcl} \mathbf{R}' & \approx & [\mathbf{I} + [w']_x] \\ \mathbf{R}'' & \approx & [\mathbf{I} + [w'']_x], \end{array}$$

where w', w'' are the angular velocity vectors and $[\cdot]_x$ is the skew-symmetric matrix of vector products:

$$[w']_x = \begin{bmatrix} 0 & -w'_3 & w'_2 \\ w'_3 & 0 & -w'_1 \\ -w'_2 & w'_1 & 0 \end{bmatrix} \qquad [w'']_x = \begin{bmatrix} 0 & -w''_3 & w''_2 \\ w''_3 & 0 & -w''_1 \\ -w''_2 & w''_1 & 0 \end{bmatrix}.$$

The motion model for calibrated cameras and small rotation is therefore:

$$\begin{array}{rcl} p' & \cong & [\mathbf{I} + [w']_x]p + kt' \\ p'' & \cong & [\mathbf{I} + [w'']_x]p + kt''. \end{array} \tag{2.16}$$

The image motion is given by:

$$\begin{array}{rcl} p' - p & \cong & [w']_x p + kt' \\ p'' - p & \cong & [w'']_x p + kt''. \end{array} \tag{2.17}$$

2.2.4 Small Motion Model

We will now develop the equations of image motion due to the infinitesimal motion of a calibrated camera. In this section we will not be using homogeneous coordinates. Let $P = (X, Y, Z)^\top$ be the coordinates of a point in the 3D world. The motion of the point P due to a rotational velocity, $w = (w_1, w_2, w_3)^\top$, and translational velocity, $t = (t_1, t_2, t_3)^\top$, is given by:

$$\begin{pmatrix} \dot{X} \\ \dot{Y} \\ \dot{Z} \end{pmatrix} = \begin{pmatrix} w_1 \\ w_2 \\ w_3 \end{pmatrix} \times \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} + \begin{pmatrix} t_1 \\ t_2 \\ t_3 \end{pmatrix} \tag{2.18}$$

We will use $p = (x, y)$ to denote the image of the point $\tilde{P}$:

$$\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} \frac{X}{Y} \\ \frac{Y}{X} \end{pmatrix} \tag{2.19}$$

The motion of the image point p is given by:

$$\begin{aligned} \frac{dx}{dt} &= \frac{d}{dt}\left(\frac{X}{Z}\right) \\ &= \frac{\dot{X}}{Z} - \frac{X\dot{Z}}{Z^2} \\ &= \left(\frac{t_1}{Z} + w_2 - w_3\frac{Y}{Z}\right) - \frac{X}{Z}\left(\frac{t_3}{Z} + w_1\frac{Y}{Z} - w_2\frac{X}{Z}\right) \end{aligned} \tag{2.20}$$

$$\begin{aligned} \frac{dy}{dt} &= \frac{d}{dt}\left(\frac{Y}{Z}\right) \\ &= \frac{\dot{Y}}{Z} - \frac{Y\dot{Z}}{Z^2} \\ &= \left(\frac{t_2}{Z} - w_1 + w_3\frac{X}{Z}\right) - \frac{Y}{Z}\left(\frac{t_3}{Z} + w_1\frac{Y}{Z} - w_2\frac{X}{Z}\right) \end{aligned} \tag{2.21}$$

Noting that $x = \frac{X}{Z}$ and $y = \frac{Y}{Z}$, and rearranging the terms we get the infinitesimal motion equations of Longuet-Higgins & Prazdny [42]:

$$\begin{aligned} u &= \frac{1}{z}(t_1 - xt_3) - w_3 y + w_2(1 + x^2) - w_1' xy \\ v &= \frac{1}{z}(t_2 - yt_3) + w_3 x - w_1(1 + y^2) + w_2' xy \end{aligned} \tag{2.22}$$

where we use $u = \frac{dx}{dt}$ and $v = \frac{dy}{dt}$. For discrete time steps: $u' = x' - x$ and $v' = y' - y$.

2.2.5 Weak Perspective, Orthographic and Affine

Weak Perspective

We will consider the case of a simple set of internal camera parameters:

$$\mathbf{M} = \begin{pmatrix} -f & 0 & 0 \\ 0 & -f & 0 \\ 0 & 0 & 1 \end{pmatrix}. \tag{2.23}$$

In this case the camera matrix $\tilde{\mathbf{A}}$ is given by:

$$\tilde{\mathbf{A}} = \begin{pmatrix} -fR_1 & -ft_1 \\ -fR_2 & -ft_2 \\ R_3 & t_3 \end{pmatrix} \tag{2.24}$$

where R_i denotes the i'th row of the rotation matrix $\mathbf{R}$. A set of points $P_i = (X_i, Y_i, Z_i, 1)^\top \in \mathcal{P}^3$ project to the points $p_i = (x_i, y_i, 1)^\top \in \mathcal{P}^2$. Let $\tilde{P}_i = (X_i, Y_i, Z_i)^\top \in \mathcal{R}^3$ denote the coordinates of the points in Euclidean coordinates.

$$p_i \cong \begin{pmatrix} -fR_1\tilde{P}_i - ft_1 \\ -fR_2\tilde{P}_i - ft_2 \\ R_3\tilde{P}_i + t_3 \end{pmatrix} \tag{2.25}$$

Now let us assume that the distance to the centroid of set of points P_0 is large compared to the variation in depth among the points. If $P_i = P_0 + \delta_i$ this assumption translates to:

$$\frac{R_3\delta_i}{R_3P_0} << 1. \tag{2.26}$$

Using this assumption we can write equation (2.25) as:

$$p_i \cong \begin{pmatrix} -fR_1\tilde{P}_i - ft_1 \\ -fR_2\tilde{P}_i - ft_2 \\ R_3\tilde{P}_0 + t_3 \end{pmatrix}. \tag{2.27}$$

The third element of the projective image point p_i is the same for all the points. From equation (2.27) under this assumption $\tilde{\mathbf{A}}$ has the form:

$$\tilde{\mathbf{A}}_{\mathbf{wp}} = \begin{pmatrix} -fR_1 & -ft_1 \\ -fR_2 & -ft_2 \\ 0 & R_3\tilde{P}_0 + t_3 \end{pmatrix}. \tag{2.28}$$

This projection is called *weak perspective*. It preserves parallelism since a point $P_\infty = (X, Y, Z, 0)$ on the plane at infinity is projected to point $p_\infty = (x, y, 0)$ on the line at infinity on the image plane. We will now explore how the image of a set of points changes as the camera moves.

Translation: We can see from equation (2.28) that motion in the depth (translation along the optical axis) will amount to a uniform scaling of the whole image. The value of the scaling depends on the depth to the centroid of the points P_0. Translation along the X or Y axis will cause a uniform shift of the image. This is in contrast to perspective projection where the image motion of each point depended on the depth of that point. In weak perspective translation does not cause parallax and does not give information about relative depth.

Rotation: Rotation around the optical axis is simply a rotation of the image just like in the perspective case. For rotation around the Y axis the rotation matrix R is given by:

$$\mathbf{R} = \begin{pmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{pmatrix}$$

which results in the projection matrix:

$$\tilde{\mathbf{A}}_{\mathbf{wp}} = \begin{pmatrix} f\cos\theta & 0 & f\sin\theta & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 0 & -\sin\theta X_0 + \cos\theta Z_0 \end{pmatrix}. \tag{2.29}$$

The motion of point in the image due to rotation depends on the depth. This is very different from perspective projection where camera rotation gives no depth information.

Does this Contradict the Perspective Model? We have found that in weak perspective, translation appears to give no depth information while rotation does.

This appears to contradict the perspective model where translation provides depth information through parallax while rotation gives no depth information. If we look carefully at the weak perspective equation (2.29) we notice that the image motion due to rotation around the Y axis, neglecting the effects of the overall scale change, is given by:

$$u'_i = x'_i - x_i \approx \sin\theta \frac{Z_i}{Z_0} = \sin\theta(1 + \frac{\delta_i}{Z_0}).$$

But $Z_0 >> \delta_i$ and therefore the variation in image flow over the set of points will be small (and that is where the depth or shape information lies). To get a large variation we need to use large angles of rotation and then the term $\sin\theta(1 + \frac{\delta_i}{Z_0})$ will cause the set of points to disappear from inside the image. We must then translate the camera along the X axis to reposition the points inside the image. Therefore to get usable measurements we must both rotate and translate the camera. From the perspective point of view we translate the camera along the X axis to get depth from parallax but since the depth variation is very small (the weak perspective assumption) we must translate a lot to get accurate shape information.In order to keep the points in view we need to rotate the camera. So according to both models, in order to get shape information from scene with little depth variation, we must have a large translation and rotation and there is no contradiction.

The Affine Camera

The *affine camera* is a generalization of weak perspective where there is no restriction on the elements of the projection matrix other than that the first three element of the third row are zero:

$$\tilde{\mathbf{A}}_{\mathbf{affine}} = \begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ 0 & 0 & 0 & a_{34} \end{pmatrix}.$$

The affine projection also preserves the plane at infinity and therefore preserves parallelism. We can use this model if we do not know the internal camera parameters.

Orthographic Projection

Orthographic projection is a specialization of the weak perspective to the case where the centroid of the points is infinitely far away but the focal length (f) is infinitely large. Thus the scale factor is constant and does not change with motion along the optical axis:

$$\tilde{\mathbf{A}}_{\mathbf{ortho}} = \begin{pmatrix} R_1 & t_1 \\ R_2 & t_2 \\ 0 & 1 \end{pmatrix}. \tag{2.30}$$

2.3 The Geometry of Two Views: the Epipolar Constraint

We will consider two perspective views ψ and ψ' of a 3D scene in Camera 1 and Camera 2 respectively. The centers of projection of the two cameras will be labeled O_1 and O_2 respectively. Let P be a 3D point in $\mathcal{R}^3$. Let P_1 and P_2 be the coordinates of the point P in the coordinate systems of Camera 1 and Camera 2 . The two camera coordinates frames are related through the Euclidean transformation:

$$P_1 = \mathbf{R}P_2 + T$$

where $\vec{T} = \vec{O_1O_2}$ is the coordinate vector of O_2 in the coordinate frame of Camera 1.

The vectors $\vec{O_1P}$, $\vec{O_2P}$ and $\vec{T}$ are coplanar. Therefore in coordinate frame of Camera 1:

$$\vec{O_1P} \cdot \left(\vec{T} \times \vec{O_2P}\right) = 0. \tag{2.31}$$

The cross product by T can be written using the skew symmetric matrix:

$$[t]_x = \begin{bmatrix} 0 & -t_z & t_y \\ t_z & 0 & -t_x \\ -t_y & t_x & 0 \end{bmatrix}.$$

Substituting $\vec{O_1P} = P_1$, $\vec{O_2P} = \mathbf{R}P_2 + T$ in equation (2.31) we get:

$$P_1^\top \mathbf{E} P_2 = 0 \tag{2.32}$$

where $\mathbf{E} = [t]_x\mathbf{R}$ is called the *essential matrix* [41]. The normalized image coordinates p_1 and p_2 are proportional to the 3D Cartesian point coordinates P_1 and P_2 thus:

$$p_1^\top \mathbf{E} p_2 = 0. \tag{2.33}$$

The above equation was developed for the Euclidean case. The projective case can be obtained by applying a 3×3 planar projective transformation to the normalized image coordinates of each camera:

$$p_1 = \mathbf{H}_1 p_1'$$

and

$$p_2 = \mathbf{H}_2 p_2'.$$

Equation (2.33) can then be written in the form:

$$p_1'^\top \tilde{\mathbf{H}}_1 \mathbf{E} \mathbf{H}_2 p_2' = p_1'^\top \mathbf{F} p_2' = 0 \tag{2.34}$$

where $\tilde{\mathbf{H}}_1 = \mathbf{H}_1^{(-1)}$, and $\mathbf{F} = \tilde{\mathbf{H}}_1 \mathbf{E} \mathbf{H}_2$ is called the *fundamental matrix* [16].

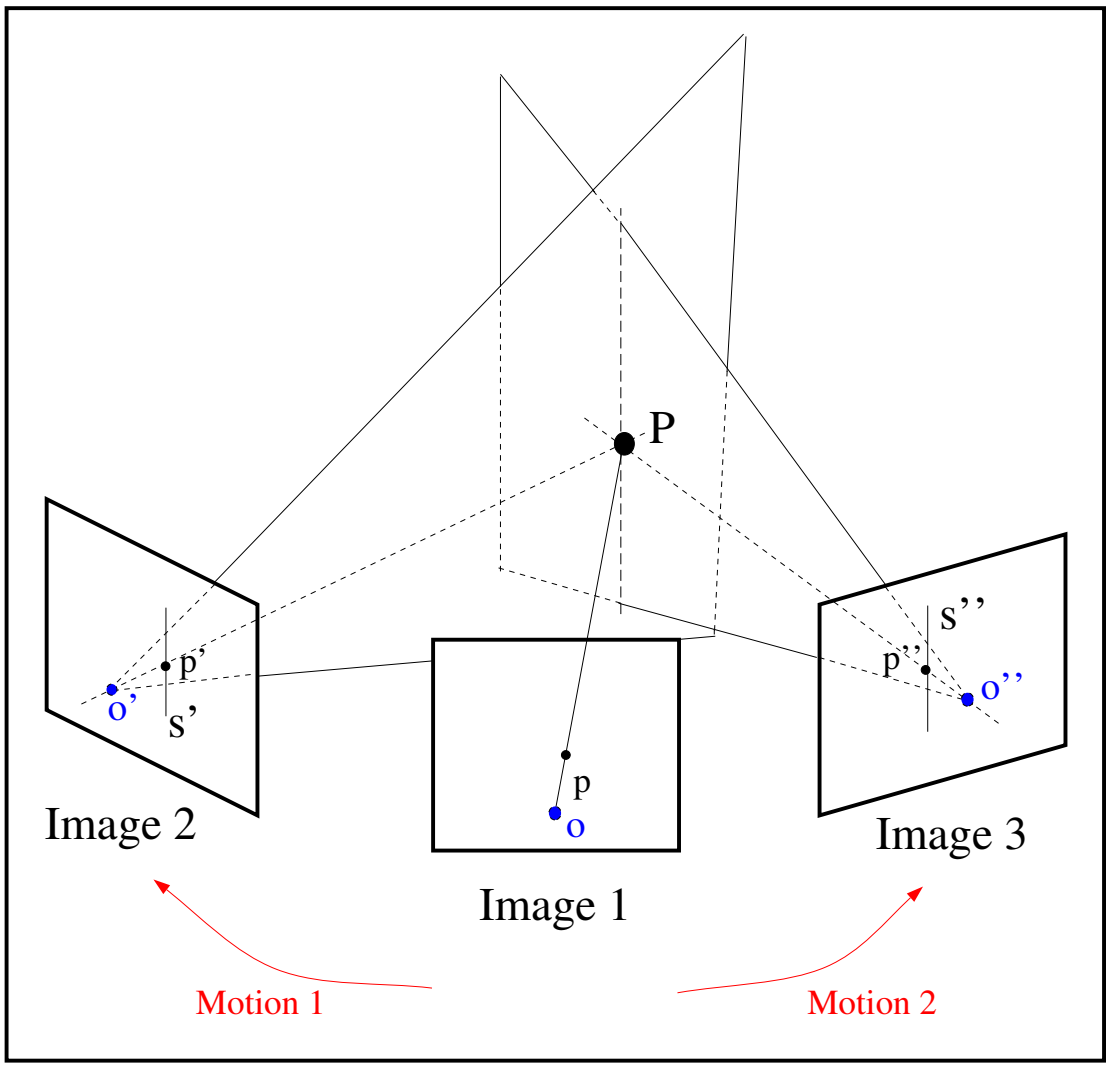


Figure 2-2: *The Geometry of Three Views: Image 1, Image 2 and Image 3 are three views obtained from cameras centered at o, o' and o'' respectively. Point P in the scene projects to image points p, p' and p''. s' and s'' are any image lines passing through the points p' and p'' respectively.*

2.4 The Geometry of Three Views - the Trilinear Tensor

We will now consider three perspective views ψ, ψ' and ψ'' of a 3D scene. Figure 2-2 shows a 3D point $P \in \mathcal{P}^3$ and its image in the three views $p \in \psi$, $p' \in \psi'$ and $p'' \in \psi''$.

Without loss of generality we can align the 3D world coordinate system with the coordinate system of the first camera:

$$p \;\cong\; \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \;\cong\; [\mathbf{I}; 0]\, P \tag{2.35}$$

Thus $P \cong (x, y, 1, \rho)^\top$. After we have set the first camera coordinate system, the other two camera coordinate systems are relative to the first. From equation (2.11):

$$\begin{array}{rcccccccc} p' & \cong & \begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} & \cong & \tilde{\mathbf{A}}P & \cong & [\mathbf{A}; t']\, P & \cong & \mathbf{A}p + \rho t' \\ p'' & \cong & \begin{pmatrix} x'' \\ y'' \\ 1 \end{pmatrix} & \cong & \tilde{\mathbf{B}}P & \cong & [\mathbf{B}; t'']\, P & \cong & \mathbf{B}p + \rho t'' \end{array} \tag{2.36}$$

The matrices $\mathbf{A}$ and $\mathbf{B}$ are homography matrices from Image 1 to Image 2 and to Image 3 respectively *due to the same plane* π. The vectors t' and t'' are the epipoles, the projection center of camera 1 projected onto the image planes of the 2nd and 3rd cameras respectively. In a calibrated setting, (i.e. if the intrinsic parameters are known), the matrices $\mathbf{A}$ and $\mathbf{B}$ are rotations and the vectors t' and t'' are the translations. Finally, in the calibrated setting ρ is replaced by $k = \frac{1}{z}$.

It has been shown [57] that the image coordinates of a point in three views are related through 4 trilinear equations. The basic idea can be neatly summarized in the form of the *trilinear tensor* in the following proposition.

Proposition 1 (The trilinear tensor) *Let P be a point in the projective space $\mathcal{P}^3$ and let its image in three views be p, p', p''. Let s' and s'' be any lines in the image plane passing through the image points p' and p'' respectively. p, s' and s'' are related through the following equation:*

$$p^i s''_k s'_j \mathcal{T}_i^{jk} = 0$$

Proof:

Let $P \cong (x, y, 1, \rho)^\top$ be a point in the projective space $\mathcal{P}^3$ and its image in 3 views, p, p', p''. Let s' and s'' be lines through points p' and p'' respectively:

$$s'^\top p' = 0 \qquad\qquad s''^\top p'' = 0 \tag{2.37}$$

As we recall from equation (2.36):

$$\begin{array}{rcl} p' & \cong & \mathbf{A}p + \rho t' \\ p'' & \cong & \mathbf{B}p + \rho t'' \end{array}$$

Premultiplying the left and right hand sides of the first equation by $s'^{\top}$ and the second equation by $s''^{\top}$ we get:

$$\begin{array}{rcl} s'^{\top}\mathbf{A}p + \rho s'^{\top}t' & = & 0 \\ s''^{\top}\mathbf{B}p + \rho s''^{\top}t'' & = & 0 \end{array} \tag{2.38}$$

Eliminating ρ from the above equations results in the equation:

$$s'^{\top}t's''^{\top}\mathbf{B}p - s''^{\top}t''s'^{\top}\mathbf{A}p = 0 \tag{2.39}$$

This can be written compactly using tensor notation:

$$p^i s''_k s'_j \mathcal{T}_i^{jk} = 0 \tag{2.40}$$

where $\mathcal{T}$ is the tensor representing a bilinear function of the camera matrices:

$$\mathcal{T}_i^{jk} = t'^j b_i^k - t''^k a_i^j \tag{2.41}$$

□

Equation (2.40) relates a point p in Image 1 and lines s' and s'' passing through the corresponding points p' and p'' in Image 2 and Image 3 respectively. It is important to note that the lines s' and s'' do not have to correspond to any physical line in space or the image. Taking any two distinct lines through p' and two distinct lines through p'' gives rise to four distinct trilinear equations. Any others lines through p' or p'' are simply linear combinations of the previous ones and therefore give rise to no new constraints. We can write equations relating point correspondences between three views if we choose the *standard* form where s' (and s'') represent either the vertical or horizontal scan lines, i.e.,

$$s' = \begin{pmatrix} -1 \\ 0 \\ x' \end{pmatrix} \quad or \quad s' = \begin{pmatrix} 0 \\ -1 \\ y' \end{pmatrix}$$

and

$$s'' = \begin{pmatrix} -1 \\ 0 \\ x'' \end{pmatrix} \quad or \quad s'' = \begin{pmatrix} 0 \\ -1 \\ y''. \end{pmatrix}$$

Then the four trilinear forms, referred to as *trilinearities* [57], have the following explicit form:

$$\begin{array}{l} x''\mathcal{T}_i^{13}p^i - x''x'\mathcal{T}_i^{33}p^i + x'\mathcal{T}_i^{31}p^i - \mathcal{T}_i^{11}p^i = 0, \\ y''\mathcal{T}_i^{13}p^i - y''x'\mathcal{T}_i^{33}p^i + x'\mathcal{T}_i^{32}p^i - \mathcal{T}_i^{12}p^i = 0, \\ x''\mathcal{T}_i^{23}p^i - x''y'\mathcal{T}_i^{33}p^i + y'\mathcal{T}_i^{31}p^i - \mathcal{T}_i^{21}p^i = 0, \\ y''\mathcal{T}_i^{23}p^i - y''y'\mathcal{T}_i^{33}p^i + y'\mathcal{T}_i^{32}p^i - \mathcal{T}_i^{22}p^i = 0. \end{array} \tag{2.42}$$

These constraints were first derived in [57]; the tensorial derivation leading to equations (2.41) and (2.40) was first derived in [63]. The trilinear tensor has been well

known in disguise in the context of Euclidean line correspondences and was not identified at the time as a tensor but as a collection of three matrices (a particular contraction of the tensor, correlation contractions, as explained next) [68, 70, 84]. The link between the two and the generalization to projective space was identified later by Hartley [24, 25]. Additional work in this area can be found in [62, 17, 80, 30, 59, 2], and applications in [3, 74].

2.4.1 Tensor Slices and Contractions

From Figure 2-1 it is clear that the tensor can be sliced in three principle ways. This can be generalized into performing a contraction operation on the tensor with a vector. Contracting the tensor along each of the three coordinates i, j, k produces matrices each with its own distinct geometric properties.

We will mostly focus on the contraction:

$$\delta^i \mathcal{T}_i^{jk} \tag{2.43}$$

where δ^i is an *arbitrary* contravariant vector and the results of the contraction is a 3×3 matrix, which we denote by $\mathbf{T}$.

The general meaning of the matrix $\mathbf{T}$ can be found in [58] where we have the proposition:

Proposition 2 (Correlation Contractions) *The contraction $\delta^i \mathcal{T}_i^{jk}$ for some arbitrary δ^i is a rank 2 correlation matrix from Image 2 onto Image 3, that maps the dual image plane (the space of lines in Image 2) onto a set of collinear points in Image 3 that form the epipolar line corresponding to the point δ^i in Image 1. The null space of the correlation matrix is the epipolar line of δ^i in Image 2. Similarly, the transpose of $\mathbf{T}$ is a correlation from Image 3 onto Image 2 with the null space being the epipolar line in Image 3 corresponding to the point δ^i in Image 1.*

For a full proof see [58]. We will provide here a simple proof of the rank 2 constraint:

Proposition 3 *Rank($\delta^i \mathcal{T}_i^{jk}$)=2 for all choices of δ^i.*

Proof: We recall that:

$$\mathcal{T}_i^{jk} = t'^j b_i^k - t''^k a_i^j$$

Therefore:

$$\delta^i \mathcal{T}_i^{jk} = \sum_{i=1}^{3} \delta^i (t' b_i^\top - t'' a_i^\top) \tag{2.44}$$

where a_i and b_i are the i'th columns of the matrices $\mathbf{A}$ and $\mathbf{B}$ respectively. This can be rewritten as:

$$\delta^i \mathcal{T}_i^{jk} = t' \sum_{i=1}^{3} (\delta^i b_i^\top) - t'' \sum_{i=1}^{3} (\delta^i a_i^\top) \tag{2.45}$$

The outer product of two vectors is a matrix of $rank = 1$ and the matrix $\delta^i \mathcal{T}_i^{jk}$ is seen to be a linear combination of two matrices of $rank = 1$ and is therefore in general of $rank = 2$. □

We can define the *Standard Correlation Slicing* as the correlation contractions associated with selecting δ be $(1, 0, 0)$ or $(0, 1, 0)$ or $(0, 0, 1)$, thus the three standard correlation slices are $\mathcal{T}_1^{jk}, \mathcal{T}_2^{jk}$ and $\mathcal{T}_3^{jk}$, and we denote them by $\mathbf{T}_1, \mathbf{T}_2, \mathbf{T}_3$, respectively. The three standard correlations date back to the work on structure from motion of lines across three views [68, 84] where these matrices were first introduced.

We note also the contraction:

$$\mathbf{E} = \delta_k \mathcal{T}_i^{jk} = \delta_1 \mathcal{T}_i^{j1} + \delta_2 \mathcal{T}_i^{j2} + \delta_3 \mathcal{T}_i^{j3} \tag{2.46}$$

which results in the 3×3 matrix $\mathbf{E}$, and δ_k is an *arbitrary* covariant vector. The matrix $\mathbf{E}$ has a general meaning introduced in [62]:

Proposition 4 (Homography Contractions) *The contraction $\delta_k \mathcal{T}_i^{jk}$ for some arbitrary δ_k is a homography matrix from Image 1 onto Image 2 determined by the plane containing the third camera center C'' and the line δ_k in the third image plane. Generally, the rank of $\mathbf{E}$ is 3. Likewise, the contraction $\delta_j \mathcal{T}_i^{jk}$ is a homography matrix from Image 1 onto Image 3, denoted $\mathbf{W}$.*

For proof and applications see [62].

2.4.2 Tensor Admissibility Constraints

The 27 coefficients $\mathcal{T}_i^{jk}$ are not independent. It has been shown in [15] and [58] that the tensor is determined by only 18 parameters. We will first present the explanation from [58]. We will then show how from the contraction properties discussed above, one can come up with the constraints among the 27 coefficients, referred to as *admissibility constraints*. These can are grouped into three classes. (For further details see [58]).

18 Parameters

The tensor

$$\mathcal{T}_i^{jk} = t'^j b_i^k - t''^k a_i^j$$

is determined by 24 parameters given by the two camera matrices, each has 12 parameters. Two additional parameters drop out because we can scale t' and accordingly b_i^k without changing the tensor, and likewise scale t'' and accordingly a_i^j. An additional parameter drops out because of the global scale factor (tensor is determined up to overall scale). Thus, we readily see there can be at most 21 parameters defining the tensor. We can drop out three more parameters by noticing that the matrices a_i^j and b_i^k belong to a family of homography matrices that leaves the tensor unchanged (uniqueness proof in [57]), as detailed below:

$$\mathcal{T}_i^{jk} \quad = \quad t'^j b_i^k - t''^k a_i^j \tag{2.47}$$

$$
\begin{aligned}
&= t'^j(b_i^k + \alpha_i t''^k) - t''^k(a_i^j + \alpha_i t'^j) && (2.48)\\
&= \mathcal{T}_i^{jk} + \alpha_i t'^j t''^k - \alpha_i t'^j t''^k && (2.49)\\
&= \mathcal{T}_i^{jk} && (2.50)
\end{aligned}
$$

hence, we have three free parameters α_i (in geometric terms there is a free choice of reference plane in space). We can select α_i such that the matrix b_i^k will have a vanishing column (this corresponds to selecting a reference plane coplanar with the center of projection of the third view). Therefore, the new matrices a_i^j and b_i^k have only 15 non-vanishing entries, and we have reduced the number of parameters from 21 to 18.

Admissibility Constraints

We may deduce from the Correlation Contractions discussed above the following three groups of constraints that the 27 coefficients must satisfy:

1. Rank($\delta^i \mathcal{T}_i^{jk}$)=2 for all choices of δ. The three standard correlation slices $\mathbf{T}_1, \mathbf{T}_2, \mathbf{T}_3$ are of rank 2 each and this property is *closed* under all linear combinations.

2. Rank(null($\mathbf{T}_1$),null($\mathbf{T}_2$),null($\mathbf{T}_3$))=2. This follows from the fact the the null space of $\delta^i \mathcal{T}_i^{jk}$ is the epipolar line in the second image corresponding to the point δ in the first image — since all epipolar lines are concurrent, their rank is 2.

3. Rank(null($\mathbf{T}_1^\top$),null($\mathbf{T}_2^\top$),null($\mathbf{T}_3^\top$))=2. These are epipolar lines in third image, thus their rank is 2 as well.

One can easily show that no subset of these constraints is sufficient to describe an admissible tensor of the form of equation (2.41). For example, the tensor:

$$\mathcal{T}_i^{jk} = t'^j b_i^k - t''^j a_i^k$$

where we have switched the position of the j and k indices in the $t''^j a_i^k$ term, will satisfy the first constraint but not the second.

In practice, in the presence of errors in image measurements one often starts with the linear solution (that might not satisfy the admissibility constraints) and improves it further by employing a numerical Gauss-Newton style iterative procedure until a solution that satisfies the admissibility constraints is obtained (for example, [15]).

2.5 Photometric Constraints - the Optical Flow Constraint Equation

We now move on to the photometric constraints between two images. We will derive the *optical flow constraint equation* following [31]. The basic assumption of the optical flow equation is that the irradiance (brightness) of an image point (x, y) at time t (i.e. $I(x, y, t)$), is equal to the irradiance of the corresponding image point (x', y') in

the second image taken at time t' (i.e. $I(x', y', t')$). (In this section t and t' are used to denote time not translations.) This is known as the *constant brightness constraint*. It can be written as:

$$I(x + \delta x, y + \delta y, t + \delta t) = I(x, y, t) \tag{2.51}$$

where $\delta t = t' - t$, $\delta x = x' - x$ and $\delta y = y' - y$. Assuming the image brightness is smooth we can expand the left-hand-side of the equation as a Taylor series:

$$I(x, y, t) + \delta x I_x + \delta y I_y + \delta t I'_t + e = I(x, y, t) \tag{2.52}$$

where I_x, I_y, I'_t are the spatial and temporal derivatives of image brightness and e contains the higher order terms. Cancelling the terms $I(x, y, t)$ from both sides, dividing by δt and taking the limit as $\delta t \to 0$ results in the *optical flow constraint equation* of Horn & Schunk [33]:

$$u' I_x + v' I_y + I'_t = 0 \tag{2.53}$$

where:

$$u'(x, y) = \lim_{\delta t \to 0} \frac{\delta x}{\delta t} = \frac{dx}{dt}$$

and

$$v'(x, y) = \lim_{\delta t \to 0} \frac{\delta y}{\delta t} = \frac{dy}{dt}$$

are the *optical flow* values at (x, y) (i.e. the motion of the image point in the x and y directions between frames: $u' = x' - x$ and $v' = y' - y$). In practice, I_x, I_y are the components of the image gradient at location (x, y), and I'_t is the discrete temporal derivative at (x, y), i.e., $I_2(x, y) - I_1(x, y)$, where I_2 and I_1 are the image intensity values of the second and first images, respectively.

Equation (2.53) provides a single constraint on the values (u, v) given I_x, I_y and I'_t. Values of (u, v) that satisfy this equation are constrained to lie on a straight line. The component of the flow in the direction of the brightness gradient is given by:

$$\frac{I'_t}{\sqrt{I_x^2 + I_y^2}}.$$

This is also called the *normal flow*. We cannot find the component of the flow at right angles to this direction (i.e. tangent to the iso-brightness contour). This leads to the *aperture problem* described next.

2.5.1 The Aperture Problem

Imagine a black bar moving in front of a white background and suppose we can only observe the scene through a finite window or aperture (Figure 2-3). From our limited view we can tell how much the bar has moved in the direction perpendicular to the edge (i.e. in the direction of the gradient) but we cannot tell how much motion there was, if any, along the edge of the bar. Thus our interpretation of the scene is

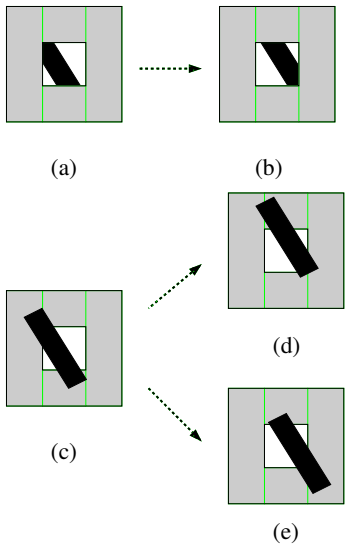


Figure 2-3: *The aperture problem: Due to the finite aperture the motion of the black bar between images (a) and (b) has an ambiguous interpretation. Two possible interpretations are the motions (c → d) and (c → e). If the window (aperture) was larger and the ends of the finite bar could be seen, this ambiguity disappears.*

ambiguous. If the window was larger and we could see the ends of the finite bar this ambiguity disappears.

Various attempts have been made to overcome the aperture problem by adding constraints on the brightness derivatives:

$$\frac{d\left(\nabla I\right)}{dt} = 0 \tag{2.54}$$

or

$$\nabla\left(\frac{dI}{dt}\right) = 0. \tag{2.55}$$

A good survey of these approaches appears in [49]. These approaches lead to additional constraint equations on the optical flow which involve second partial derivatives of the image brightness. In order to compute the optical flow it is required that the Hessian of the image brightness at time t is non singular, a condition that is not met, for example, along straight edges. Another problem with these approach is that the constraint (2.54) is only correct for surfaces that are on planes parallel to the image plane. The direction of the iso-brightness contour in the image of lines that vary in depth, does in fact change subtly. This subtlety can be important, as we will see in the 'future work' section of Chapter 3 (Section 3.8.1).

2.5.2 The Optical Flow Constraint as a Point-Line Correspondence

We now present a new interpretation of the optical flow constraint equation as a point-line correspondence between the two images (to a first order approximation).

In projective coordinates the optical flow constraint equation can be written in the form:

$$\begin{pmatrix} I_x \\ I_y \\ I'_t \end{pmatrix}^\top \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = 0 \tag{2.56}$$

The line $s = (I_x, I_y, -xI_x - yI_y)^\top$ in the projective plane passes through the point $p = (x, y, 1)^\top$ since:

$$\begin{pmatrix} I_x \\ I_y \\ -xI_x - yI_y \end{pmatrix}^\top \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = 0. \tag{2.57}$$

Combining equations (2.56) and (2.57):

$$\begin{pmatrix} I_x \\ I_y \\ -xI_x - yI_y \end{pmatrix}^\top \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} + \begin{pmatrix} I_x \\ I_y \\ I'_t \end{pmatrix}^\top \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = 0. \tag{2.58}$$

Noting that $x' = x + u$ and $y' = y + v$, and rearranging the terms, equation (2.58) becomes:

$$\begin{pmatrix} I_x \\ I_y \\ I'_t - xI_x - yI_y \end{pmatrix}^\top \begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = 0, \tag{2.59}$$

and therefore, the line:

$$s' = \begin{pmatrix} I_x \\ I_y \\ I'_t - xI_x - yI_y \end{pmatrix}^\top \tag{2.60}$$

passes through the point $p' = (x', y', 1)^\top$.

In summary: Given the coordinates of a point $p = (x, y, 1)^\top$ and the spatial and temporal derivatives (I_x, I_y, I'_t), the optical flow constraint equation gives us the equation of a line s' that passes through the corresponding point $p' = (x', y', 1)^\top$ (equation 2.60). Thus, we have a correspondence between a point in one image a line passing through the corresponding point in the second image. We do not need to know the coordinates of the point in the second image.

2.6 Summary

Geometric Constraints

The main result from this section which will be used throughout the rest of the thesis is the trilinear tensor equation:

$$p^i s''_k s'_j \mathcal{T}_i^{jk} = 0$$

where p is a point in Image 1 and s', s'' are lines going through the corresponding points p', p'' in Image 1 and Image 2 respectively. The $3\times3\times3$ tensor $\mathcal{T}_i^{jk}$ encapsulates the camera motions and internal camera parameters:

$$\mathcal{T}_i^{jk} = t'^j b_i^k - t''^k a_i^j.$$

The 27 coefficients of the tensor are not algebraically independent and there are in fact only 18 algebraically independent parameters. For a given set of 27 coefficients to be a valid tensor they must obey a set admissibility constraints.

In general, 7 corresponding points or 13 corresponding lines, in 3 views, give enough equations in form of equation (2.40) or equation (2.42) to solve for the tensor coefficients using linear methods. One can then refine the tensor coefficients to enforce the algebraic constraints [15]. As we show in Chapter 6 there are certain line configurations where the linear solution is degenerate but the admissibility constraints enable us to arrive at a unique solution.

Photometric Constraints

Given the coordinates of a point p and the spatial temporal derivatives (I_x, I_y, I'_t), the optical flow constraint equation provides a first order approximation of the equation of a line s' passing through the corresponding point p'. The line s' is given by:

$$s' = \begin{pmatrix} I_x \\ I_y \\ I'_t - xI_x - yI_y \end{pmatrix}^\top \tag{2.61}$$

Chapter 3

Model-based Brightness Constraints: Direct Estimation of Motion and Structure

3.1 Introduction

In this chapter we combine the geometric constraints of the motion models from Chapter 2 with the photometric constraints to form *model-based brightness constraints*. The result is a new method for computing the camera motion (ego-motion) between three views in an image sequence. The motion estimates are then used to compute dense structure. This method can be viewed as an extension of the 'direct methods' of Horn & Weldon [34] from two views (one motion) to three views (two motions). These methods are dubbed 'direct methods' because they do not require prior computation of optical flow.

Why three views (two motions) and not two views (one motion)? As we have seen in Section 2.5, the photometric constraints, in the form of the optical flow constraint equation, provide a matching constraint between each point in one image and a line passing through the corresponding point in the second image. In other words, for every point in one image it gives us the equation of a line along which the corresponding point must lie in the second image. There are no geometric constraints on point-line correspondences between two views. We must therefore use three view geometry, where a point in one view, and lines through the corresponding points in two other views, provide a constraint, which can be written in the form of the *trilinear tensor* (Section 2.4).

We can also use a counting argument. A single image pair provides one equation for each point in the image for a total of N equations, where N is the number of points in the image. There are $N + 5$ unknowns: the N unknown depths and the 3 translation and 3 rotation parameters with one unknown dropping out because translation and depth can only be found up to a common scale factor. The problem is therefore ill-posed. Adding another motion provides an additional N equations while adding only 6 unknowns, the second translation and rotation parameters.

By combining the optical flow constraint (Section 2.5) with the trilinear tensor model (Section 2.4) we obtain a model-based brightness constraint. The constraint describes the relationship between the spatio-temporal derivatives at each pixel in

the image with the unknown ego-motion parameters modeled by the 27 coefficients of the trilinear tensor. This constraint, the *tensor brightness constraint*, provides a linear constraint per pixel in the image which results in a highly over-constrained set of equations. We then proceed through a hierarchy of reduced motion models first by assuming calibrated cameras and small rotations, and then by assuming the Longuet-Higgins & Prazdny small motion model [42], resulting in reduced model-based brightness constraints for those motion models. Since these methods are based on the image gradients and the 'constant brightness constraint', they are only exact for infinitesimal image motion. By using a coarse-to-fine implementation our method can handle displacements averaging up to 50 pixels for 640×480 resolution images.

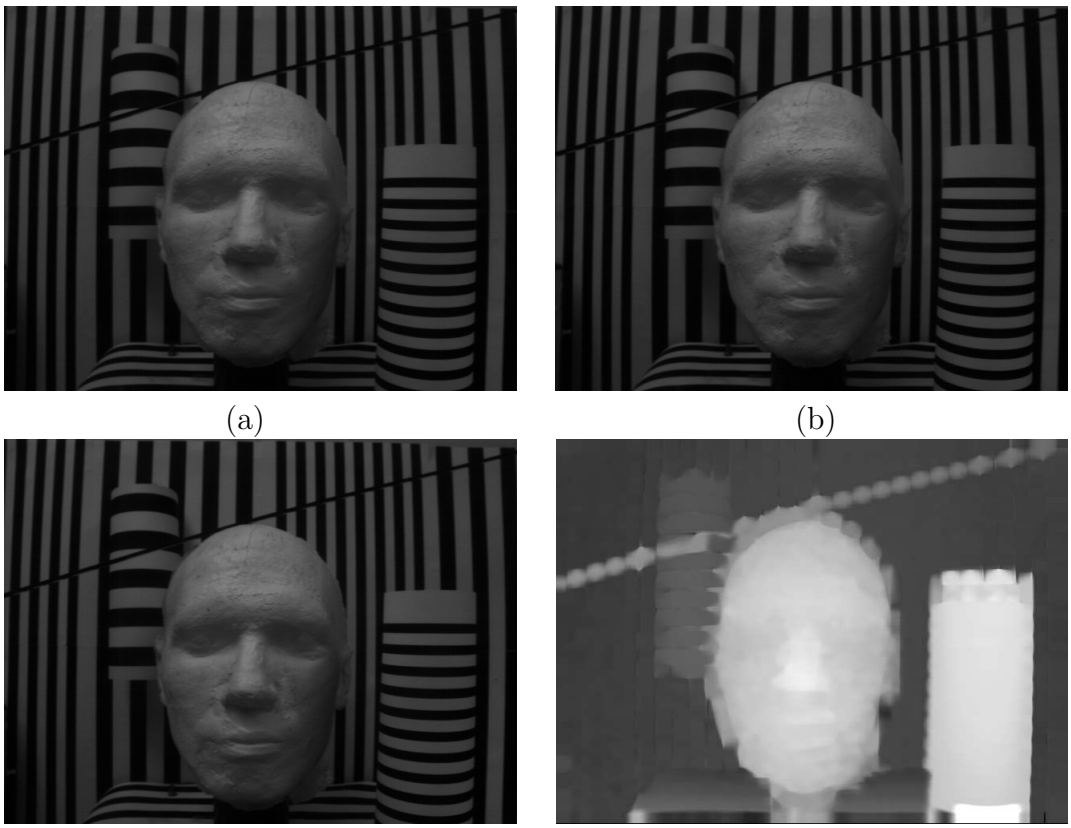
3.1.1 Related Work

Correspondence Based Methods

The standard approach to the problem of structure from motion is to first compute correspondences. These might be either dense correspondence in the form of optical flow [33, 43, 49] or feature correspondence [11, 83, 16]. Then the correspondences are used to compute the camera motion and scene structure. The advantage of our method over both optical flow methods [42, 43, 28, 55, 77] and feature based methods [79, 16, 65, 67, 4] is that no prior computation of correspondences is needed, a computation which in itself is error prone. Since we obtain a linear system of equations that combines the information from all the pixels in the image, we avoid the aperture problem. By avoiding the need to explicitly detect feature points we can use information from areas where gradient information is weak such as shadow edges.

These advantages are highlighted in a scene such as in Figure 3-1. The scene contains a background of straight bars, a plaster bust in the foreground, and an oblique line cutting the bars. The background bars provide an unreliable source of information for point-to-point correspondence because of the aperture problem. Even large regions of the image have no information which constrains the flow in the Y direction. Many feature points in the image do not correspond to real feature points in the 3D world. For example, the intersection of the bars and the line in the image do not correspond to physical features in 3D since the line does not lie on the background plane. Figure 3-2 shows a detail of the optical flow field in the upper left of the scene where these two problems occur. The flow was computed using a widely available, 'industrial strength', optical flow program [7, 6]. (For more examples see Section (4.2.3).)

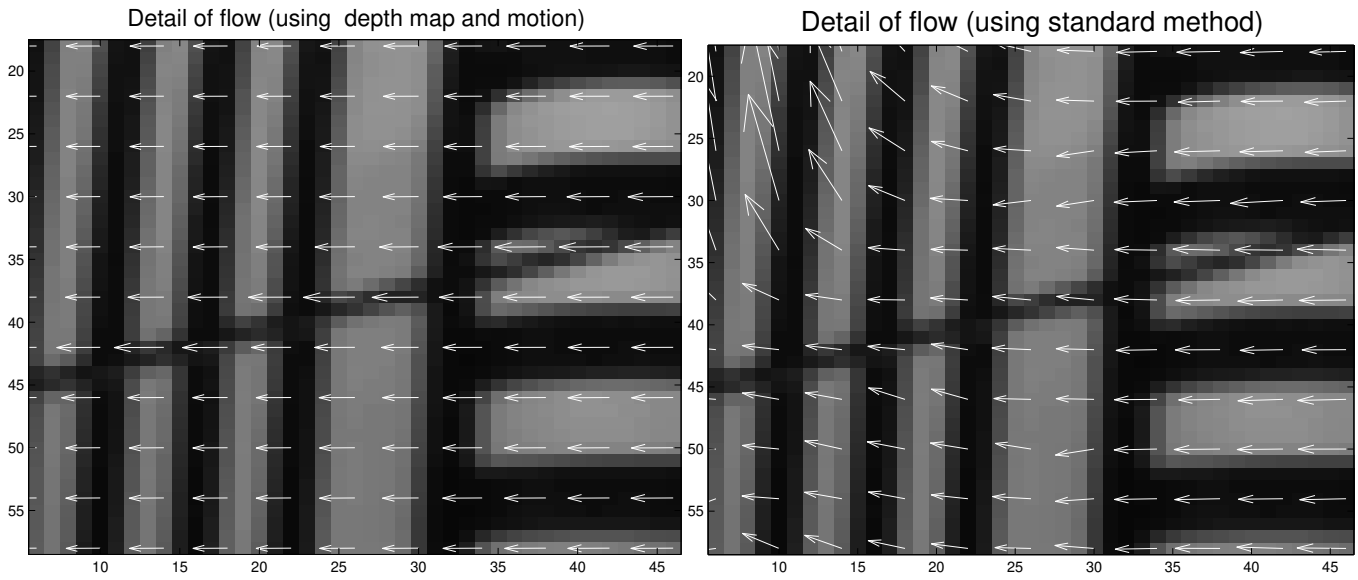
Finally, the texture on the plaster bust varies smoothly and is low contrast thereby giving rise only to a relatively small number of reliable features to track. Yet, our method produces a full 3D model of the scene as demonstrated in Figure 3-3*a,b*. Many natural scenes such as tree branches or man made objects such as window frames, lamp posts and fences, often give rise to these problems.



(a) (b)

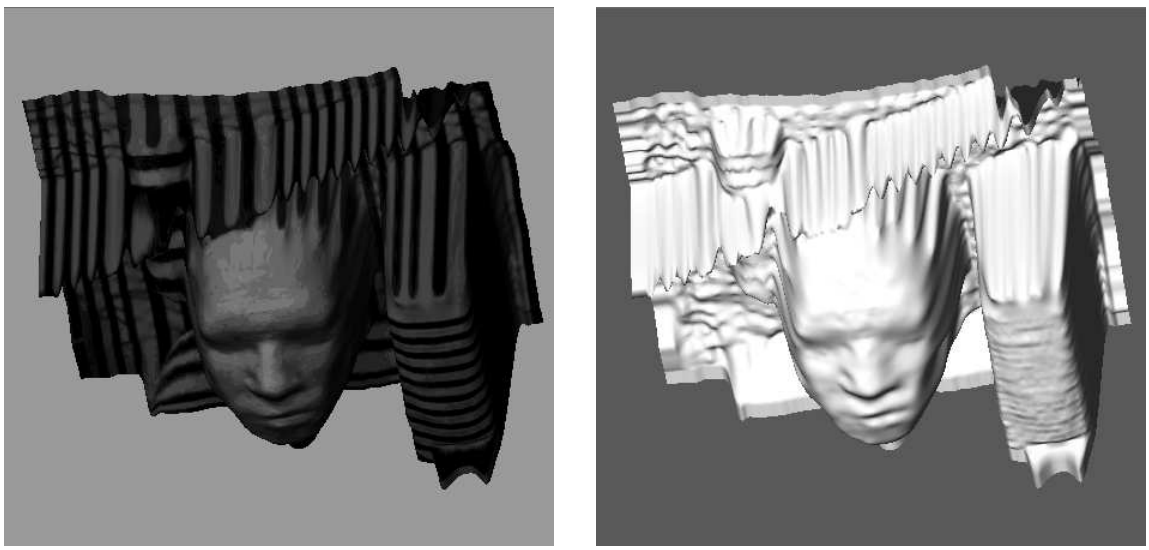
(c) (d)

Figure 3-1: *The three input images (a,b,c) and the estimated depth map (d). The motion between images (a) and (b) is horizontal. The motion is small but can be seen in the width of the rightmost vertical stripe and in the parallax between the cylinder on the right and the vertical stripes behind it. The motion of between images (a) and (c) was vertical.*



(a) (b)

Figure 3-2: *Detail of optical flow computed for Figure 3-1. (a) Correct optical flow computed using depth map and recovered motion. The camera motion was horizontal parallel to the image plane. The flow vectors have a zero Y component. (b) Optical flow computed using Bergen & Hingorani optical flow program. Note in the upper left corner the flow vectors have a strong Y component. This is due to the aperture problem where even a large window (aperture) will see image gradients in only one direction giving no constraint on the Y component of the flow. Near the intersections (in the image) of the diagonal line and the vertical bars the flow has a small Y component. This error occurs because the program 'tracks' the intersection point as if it were a real 'feature' point, but the lines do not intersect in space. Although smaller in magnitude this is the more significant error because the program will also give a high confidence to this value.*



(a) (b)

Figure 3-3: *3D rendering of the estimated depth map from images in Figure 3-1. These images show the inverse depth, $k = \frac{1}{z}$, which is the natural value to compute. In (b) the texture map was removed to show the detail and the flaws.*

Direct Methods

The 'direct methods' were pioneered by Horn & Weldon in [34]. Using only a single image pair they ended up with N equations in $N+5$ unknowns, where N is the number of points in the image. The unknowns are the N unknown depths and the 3 translation and 3 rotation parameters with one unknown dropping out because translation and depth can only be found up to a common scale factor. The problem is therefore ill posed and additional constraints are needed. Negahdaripour & Horn [52] present a closed form solution assuming a planar or quadratic surface. Szeliski & Kang [76] describe an iterative solution using splines to enforce a smoothness constraint on the depth. McQuirk [45] shows that in a pure translation model the subset of the image points with a nonzero spatial derivative but a zero time derivative gives the direction of motion, thus the *focus of expansion* (FOE) is on a line perpendicular to the gradient at these points. Since this uses only a subset of the image points, one ends up discarding most of the image data.

Heel [29] constructs a Kalman filter to build up a structure model from more than one image pair but the core computation is fundamentally the same single image pair computation. The basic idea is that the Horn & Weldon equation is linear in depth if motion is known and linear in motion if depth is known. Heel initially assumes a depth (constant depth surface) and estimates the motion and then uses the motion to estimate the depth. The depth map is then warped using the computed motion and used as the initial depth estimate for the next image pair in the sequence. The question of convergence is not answered and failures are not reported. Results are only shown for pure translation. Heel limits the motion to image motions of less than 3 pixels which is one of the limiting factors in the accuracy of the surface

reconstructions. A coarse-to-fine framework (Section 3.7.4) enables us to use much larger image motions, thus increasing the dynamic range by an order of magnitude.

Michaels [46] uses three frames (two motions). Each motion gives N equations totaling $2N$ equations. The unknowns are the N unknown depths and the 2×6 translation and rotation parameters. Again, one parameter drops out because of the translation and depth scale ambiguity. He solves the $2N$ equations with $N + 11$ unknowns as a large nonlinear optimization problem using the Levenberg-Marquart algorithm [47]. Since it took a very long time to converge results are shown for very low resolution images only. He also shows theoretically that a large field of view ($\approx 120^o$) is required for accurate direct estimation of motion given structure. Since estimation of motion given structure is a key stage in the process he suggests that a wide field of view is required for the whole process of estimation of structure and motion. Kumar et al [40] first align a dominant plane in the images. The residual image motion is epipolar motion (i.e. pure translation). The epipole of the residual motion is found using iterative techniques starting with an initial guess. Others who have explored planar alignment include [36, 56]. The theoretical framework for planar alignment can also be found in [60].

More recently, Fermuller & Aloimonos [18] describe global geometric properties of the flow-field that give rise to direct relationships between the measurement of normal flow and the ego-motion parameters solved by means of search techniques.

The tensor brightness constraint was first presented in [64] and a practical implementation with results was first described in [74]. During the course of implementation various subtleties and limitations were discovered.

3.1.2 Overview of this Chapter

In Section 3.2 we derive the the tensor brightness constraint, the model-based brightness constraint for the projective camera model. The constraint provides one linear equation per pixel in the 27 unknown elements of the trilinear tensor. This results in a highly over-constrained set of equations but we show that these equations are linearly dependent. A unique solution is obtainable by applying further admissibility constraints as shown in Chapter 6 but this added complexity in finding the unique solution leads us to search for simpler models.

We next consider the case of a calibrated camera with the rotation limited to small rotation angles (Section 3.3). This results in a 20-parameter model but again the linear solution is degenerate. Therefore we develop a simpler model in Section 3.4 using the small motion approximations of Longuet-Higgins & Prazdny [42]. This results in a 15-parameter model-based brightness constraint. Before showing how to solve for the ego-motion parameters for the general case of rotation and translation, we show how to solve for the motion parameters in the simpler example of pure translation in Section 3.5. This example is useful because it shows the basic method we employ to solve the linear homogeneous equations. In Section 3.6 we show how to solve the 15-parameter model for both translation and rotation.

In Section 3.7 we discuss the implementation issues and we provide the details needed to build a program that actually works.

3.2 The Tensor Brightness Constraint: the Projective Model

As we have seen in Chapter 2, a geometric trilinear matching constraint is produced by contracting the tensor with the point p of Image 1, *any* line coincident with p' in Image 2, and *any* line coincident with p'' in Image 3 (equation 2.41). In particular, we may use as the lines the tangents to the iso-brightness contours at p' and p'', respectively, as suggested by [69]. However, there still remains the problem of finding those matching tangents in the first place.

In order to avoid the need to first find correspondences we use the results from Section 2.5.2. The lines

$$s' = \begin{pmatrix} I_x \\ I_y \\ I'_t - xI_x - yI_y \end{pmatrix} \qquad s'' = \begin{pmatrix} I_x \\ I_y \\ I''_t - xI_x - yI_y \end{pmatrix} \tag{3.1}$$

are lines coincident with p' and p'' respectively and parallel to the iso brightness contour at (x, y). It is important to note that equations of the lines s' and s'' (equation 3.1) can be computed from the coordinates (x, y) and the derivatives I_x, I_y, I'_t, I''_t. I_x and I_y are the spatial derivatives in Image 1 at point (x, y). I'_t and I''_t are the temporal derivatives computed at the point (x, y). In practice $I'_t = I_2(x, y) - I_1(x, y)$ and $I''_t = I_3(x, y) - I_1(x, y)$. The coordinates of the corresponding points (x', y') and (x'', y'') are not required.

Substituting equations (3.1) into the tensor equation (2.41) results in the *tensor brightness constraint*:

$$\boxed{s''_k s'_j p^i \mathcal{T}_i^{jk} = 0.} \tag{3.2}$$

We have one such equation for each point on the image where s'_j and s''_k can be computed from the image gradients and $p = (x, y, 1)^T$ are the (projective) image coordinates of the point in Image 1. We wish to solve for $\mathcal{T}_i^{jk}$ the elements of which are the bilinear combinations of the motion and camera parameters.

Every pixel with a non-vanishing gradient contributes one linear equation to the 27 unknown parameters comprising $\mathcal{T}_i^{jk}$. However, a configuration of a point in the first image and two *parallel* lines in Images 2 and 3 is a particular instance of a degenerate line configuration called a Linear Line Complex. The general solution for the LLC case is explored in Chapter 6. In this particular case the system of equations can provide a linear solution to 21 of the parameters and the remaining 6 parameters of $\mathcal{T}_i^{jk}$ can be determined using quadratic admissibility constraints. We provide here a proof for this special case:

From equation (3.2) the coefficients of the terms $\mathcal{T}_i^{12}$ and the terms $\mathcal{T}_i^{21}$ are both $p^i I_x I_y$. Therefore the linear equations cannot be solved for the six terms ($\mathcal{T}_i^{12}$ and $\mathcal{T}_i^{21}$) individually but only for the three sums $\mathcal{T}_i^{12} + \mathcal{T}_i^{21}, i = 1...3$.

The sum $\mathcal{T}_i^{12} + \mathcal{T}_i^{21}$ provides one linear equation in $\mathcal{T}_i^{12}$ and $\mathcal{T}_i^{21}$. We will use the admissibility constraint on the Standard Correlation Slices, T_i. The constraint states

that the matrix T_i is of $rank = 2$. This leads to a quadratic equation in $\mathcal{T}_i^{12}$ and $\mathcal{T}_i^{21}$. The quadratic and linear equations together result in two solutions for each pair $\mathcal{T}_i^{12}, \mathcal{T}_i^{21}, i = 1...3$ for a total of 8 discrete solutions. A unique solution is obtainable by applying further admissibility constraints as shown in Chapter 6.

This added complexity in finding the unique solution leads us to search for simpler models. We next consider the case of a calibrated camera with the rotation limited to small rotation angles.

3.3 The Small Rotation Model with Calibrated Cameras

The next model is defined for small-angle rotations with calibrated cameras. Assuming calibrated cameras equations (2.36) become:

$$\begin{array}{rcccl} p' & \cong & [\mathbf{R}'; t']\, P & \cong & \mathbf{R}'p + kt' \\ p'' & \cong & [\mathbf{R}''; t'']\, P & \cong & \mathbf{R}''p + kt'' \end{array} \tag{3.3}$$

i.e., $\tilde{A} = [\mathbf{R}'; t']$ and $\tilde{B} = [\mathbf{R}''; t'']$ and $k = \frac{1}{z}$ replaces ρ in equation (2.36).

If we also assume small angle rotations and can therefore make the approximations:

$$\cos(\theta) \approx 1, \quad \sin(\theta) \approx \theta$$

then:

$$\begin{array}{rcl} p' & \cong & [I + [w']_x; t']\, P \\ p'' & \cong & [I + [w'']_x; t'']\, P \end{array} \tag{3.4}$$

where w', w'' are the angular velocity vectors and $[\cdot]_x$ is the skew-symmetric matrix of vector products. (i.e., $\tilde{A} = [I + [w']_x; t']$ and $\tilde{B} = [I + [w'']_x; t'']$).

We can now proceed along two paths. The different results will give us some further insight into the problem. First we will substitute $\tilde{A} = [I + [w']_x; t']$ and $\tilde{B} = [I + [w'']_x; t'']$ directly into equation (2.41) and then write out the tensor explicitly:

$$T_1 = \begin{bmatrix} t'_1 - t''_1 & t'_1 w''_3 - t''_2 & -t'_1 w''_2 - t''_3 \\ t'_2 - t''_1 w'_3 & t'_2 w''_3 - t''_2 w'_3 & -t'_2 w''_2 - t''_3 w'_3 \\ t'_3 + t''_1 w'_2 & t'_3 w''_3 + t''_2 w'_2 & -t'_3 w''_2 + t''_3 w'_2 \end{bmatrix} \tag{3.5}$$

$$T_2 = \begin{bmatrix} -t'_1 w''_3 + t''_1 w'_3 & t'_1 + t''_2 w'_3 & t'_1 w''_1 + t''_3 w'_3 \\ -t'_2 w''_3 - t''_1 & t'_2 - t''_2 & t'_2 w''_1 - t''_3 \\ -t'_3 w''_3 - t''_1 w'_1 & t'_3 - t''_2 w'_1 & t'_3 w''_1 - t''_3 w'_1 \end{bmatrix} \tag{3.6}$$

$$T_3 = \begin{bmatrix} t'_1 w''_2 - t''_1 w'_2 & -t'_1 w''_1 - t''_2 w'_2 & t'_1 - t''_3 w'_2 \\ t'_2 w''_2 + t''_1 w'_1 & -t'_2 w''_1 + t''_2 w'_1 & t'_2 + t''_3 w'_1 \\ t'_3 w''_2 - t''_1 & -t'_3 w''_1 - t''_2 & t'_3 - t''_3 \end{bmatrix} \tag{3.7}$$

where T_1, T_2 and T_3 are the Standard Correlation Slices (see Section 2.4.1). Studying the tensor terms we notice that the terms are not linearly independent and the

following seven linear equations hold:

$$\begin{array}{l}
\mathcal{T}_1^{1,2} + \mathcal{T}_2^{1,1} + \mathcal{T}_1^{2,1} - \mathcal{T}_2^{2,2} = 0 \\
\mathcal{T}_2^{2,1} + \mathcal{T}_1^{2,2} + \mathcal{T}_2^{1,2} - \mathcal{T}_1^{1,1} = 0 \\
\mathcal{T}_3^{1,1} + \mathcal{T}_1^{1,3} + \mathcal{T}_1^{3,1} - \mathcal{T}_3^{3,3} = 0 \\
\mathcal{T}_1^{3,3} + \mathcal{T}_3^{3,1} + \mathcal{T}_3^{1,3} - \mathcal{T}_1^{1,1} = 0 \\
\mathcal{T}_2^{3,3} + \mathcal{T}_3^{3,2} + \mathcal{T}_3^{2,3} - \mathcal{T}_2^{2,2} = 0 \\
\mathcal{T}_3^{2,2} + \mathcal{T}_2^{2,3} + \mathcal{T}_2^{3,2} - \mathcal{T}_3^{3,3} = 0 \\
\mathcal{T}_2^{1,3} + \mathcal{T}_3^{2,1} + \mathcal{T}_1^{2,3} + \mathcal{T}_3^{1,2} + \mathcal{T}_1^{3,2} + \mathcal{T}_2^{3,1} = 0
\end{array} \tag{3.8}$$

Therefore, for the calibrated small rotation model, the 27-parameter Tensor Brightness Constraint reduces to a *20-parameter* constraint equation. Each point in the image gives one homogeneous equation resulting in a set of N homogeneous equation, where N is the number of image points. This can be written in matrix form as:

$$\mathbf{A}x = 0$$

where $\mathbf{A}$ is the $N \times 20$ estimation matrix and x is the 20 parameter vector of the unknown intermediate parameters. There is a unique non-trivial solution if the estimation matrix is of $rank = 19$. But as in the projective case, the system of homogeneous equations is degenerate and in this case the rank of the estimation matrix is 16 not 19.

In [64, 74] a model-based brightness constraint for the calibrated, small rotation model was derived in a different way. Expanding equation (3.4):

$$p' \;\cong\; [I + [w']_x]\, p + kt' \tag{3.9}$$

and taking the dot product of the left-hand-side and right-hand-side terms with s' (equations 3.1) results in the equation:

$$0 \;=\; s'^{\top} p + s'^{\top} [w']_x + k s'^{\top} t' \tag{3.10}$$

This can be simplified to the form:

$$k s'^{\top} t' + v'^{\top} w' + I'_t \;=\; 0 \tag{3.11}$$

where:

$$v' = p \times s' = \begin{pmatrix} -I_y + y(I'_t - xI_x - yI_y) \\ I_x - x(I'_t - xI_x - yI_y) \\ xI_y - yI_x \end{pmatrix}.$$

In a similar manner, for the second motion:

$$k s''^{\top} t'' + v''^{\top} w'' + I''_t \;=\; 0 \tag{3.12}$$

where

$$v'' = p \times s'' = \begin{pmatrix} -I_y + y(I''_t - xI_x - yI_y) \\ I_x - x(I''_t - xI_x - yI_y) \\ xI_y - yI_x \end{pmatrix}.$$

Multiplying equations (3.11) by $s''^{\top}t''$ and equations (3.12) by $s'^{\top}t'$ and subtracting we obtain the following *24-parameter* model-based brightness constraint as a reduction of the tensor brightness constraint:

$$I''_t s'^{\top}t' - I'_t s''^{\top}t'' + s'^{\top}[t'w''^{\top}]v'' - s''^{\top}[t''w'^{\top}]v' = 0. \tag{3.13}$$

The resulting 24-parameter model (equation 3.13) is not as compact as the 20-parameter model we obtained by direct substitution of the small rotation approximation into the tensor brightness constraint equation. We find that the $24 \times N$ estimation matrix obtained using the 24-parameter model is again of rank=16. Due to the high degeneracy of the linear equations neither the 24-parameter model nor the 20-parameter model are convenient to work with.

3.4 The 15-Parameter, Small Motion Model

A unique solution is obtained when we reduce the motion model further to include infinitesimal motion using the model introduced by Longuet-Higgins & Prazdny [42] (see Section 2.2.4). This model assumes in addition to small rotation that $\frac{t'_z}{Z} \ll 1$. The motion field equations for the first camera motion are then:

$$\begin{aligned} u' &= \frac{1}{z}(t'_1 - xt'_3) - w'_3 y + w'_2(1 + x^2) - w'_1 xy \\ v' &= \frac{1}{z}(t'_2 - yt'_3) + w'_3 x - w'_1(1 + y^2) + w'_2 xy \end{aligned} \tag{3.14}$$

By substituting equation (3.14) into the optical flow constraint equation (equation2.53, Section 2.5):

$$u'I_x + v'I_y + I'_t = 0 \tag{3.15}$$

and rearranging the terms we obtain:

$$ks^{\top}t' + v^{\top}w' + I'_t = 0 \tag{3.16}$$

where $k = \frac{1}{z}$ denotes the inverse depth at each pixel location and where s, v are defined below:

$$s = \begin{pmatrix} I_x \\ I_y \\ -xI_x - yI_y \end{pmatrix} \tag{3.17}$$

and

$$v = p \times s = \begin{pmatrix} -I_y - y(xI_x + yI_y) \\ I_x + x(xI_x + yI_y) \\ xI_y - yI_x \end{pmatrix} \tag{3.18}$$

Note that v is a vector that can be computed from image brightness, and should not be confused with v' which is the y component of the image flow between Image 1 and Image 2. (Equation (3.16) which was first derived in [52, 34] can also be derived

directly from equation (3.11) by making the LH&P assumptions: $\frac{t'_z}{Z} \ll 1$, $\frac{w'_y x}{f} \ll 1$ and $\frac{w'_x y}{f} \ll 1$.)

Similarly for the second motion:

$$ks^\top t'' + v^\top w'' + I''_t = 0 \tag{3.19}$$

By eliminating k from equations (3.16) and (3.19) we obtain the *15-parameter model-based brightness constraint*:

$$\boxed{I''_t s^T t' - I'_t s^T t'' + s^T [t' w''^T - t'' w'^T] v = 0} \tag{3.20}$$

Equation (3.20) is a key equation. There is one such equation for every point in the image. The unknowns are the motion parameters t', t'', w' and w''. The values I'_t, I''_t, s and v, can all be computed from image derivatives and the coordinates of the point in Image 1. Given the solution to the ego-motion parameters (to be described later), one can recover the dense depth map from equations (3.16) and (3.19).

Before showing how to solve equation (3.20) for both rotation and translation we will start with the simple case of pure translation. This simple will be used to explain the basic method we employ to find the motion parameters. The pure translation case arises in practice when we can either assure pure translation or we have previously rectified the images (perhaps by registering a common plane in the images as in Section 3.6.4). It is also an important case from a theoretical point of view. We will show that there are problems in recovering the motion parameters in the case of collinear motion. If collinear motion causes a problem for the pure translation case, it will also create a problem for the more general translation and rotation case.

3.5 Pure Translation

The pure translational small-motion model takes the simplest form:

$$I''_t s^T t' - I'_t s^T t'' = 0 \tag{3.21}$$

We have one such equation for each image point and we can write it out in the matrix form:

$$\mathbf{A}t = 0$$

where:

$$t = \begin{pmatrix} t'_x & t'_y & t'_z & t''_x & t''_y & t''_z \end{pmatrix}^T$$

and $\mathbf{A}$ is an $N \times 6$ matrix with the $i'th$ row (corresponding to the $i'th$ pixel) given by:

$$\begin{pmatrix} I''_t s_{i1} & I''_t s_{i2} & I''_t s_{i3} & I'_t s_{i1} & I'_t s_{i2} & I'_t s_{i3} \end{pmatrix}$$

To avoid the trivial solution $t = 0$ we add the constraint $\|t\| = 1$. The least squares problem now maps to the problem of finding $\|t\| = 1$ that minimizes:

$$t^T \mathbf{A}^T \mathbf{A} t = 0$$

The solution is the eigenvector of $\mathbf{A}^T\mathbf{A}$ corresponding to the smallest eigenvalue.

3.5.1 The Singular Case of Collinear Motion

This method fails when the two motions are in the same (or opposite) directions. This is obvious in the pure translation case where:

$$I''_t s^\top t' - I'_t s^\top t'' = 0 \tag{3.22}$$

and

$$k s^\top t' + I'_t = 0 \tag{3.23}$$
$$k s^\top t'' + I''_t = 0 \tag{3.24}$$

If the second translation vector t'' is proportional to the translation vector t' then equation (3.24) is simply a scaled version of equation (3.23) adding no new information. The solution will therefore be ill conditioned when $t' = \beta t''$ for a scalar β. This is a drawback in many applications (e.g. 3D reconstruction from a monocular image sequence). We present in Section 3.8.1 some early research on ways to overcome this problem. The initial results look promising.

3.6 Solving for Translation and Rotation

In equation (3.20) the ego-motion parameters are embedded in a 15 parameter model: 6 translation parameters and the 9 outer-product terms $[t'w''^T - t''w'^T]$. Each pixel provides one linear equation, thus N pixels provide a system $\mathbf{A}c = 0$ where $\mathbf{A}$ is an $N \times 15$ measurement matrix and c is the 15-vector of unknown parameters. The least squares solution is the eigenvector of $\mathbf{A}^\top\mathbf{A}$ corresponding to the smallest eigenvalue. Without noise the eigenvalue will be zero and the solution is in the null space of $\mathbf{A}^\top\mathbf{A}$. The solution is unique if the null space is one dimensional and the rank of $\mathbf{A}$ is 14. We will prove however, that the null space is at least two dimensional.

Proposition 5 *The rank of* $\mathbf{A}$ *is 13 not 14*

Let $\mathbf{A}$ *be the* $N \times 15$ *measurement matrix associated with the homogeneous equation (3.20), where* $N \gg 15$*. Then the null space of the matrix* $\mathbf{A}^\top\mathbf{A}$ *is of dimension greater or equal to 2 and* $Rank(\mathbf{A}) \leq 13$*.*

Proof:
Since

$$s^\top v = s^\top (p \times s) = 0$$

the vector

$$c_0 = (0,0,0,0,0,0,1,0,0,0,1,0,0,0,1)^T$$

is in the null space of $\mathbf{A}$.
The vector c_0 corresponds to:

$$t' = 0$$
$$t'' = 0$$
$$\left[t'w''^T - t''w'^T\right] = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Therefore c_0 is clearly not an admissible solution, and therefore the null space of $\mathbf{A}$ includes two vectors: the vector c_0 and the true solution. It follows that the rank of $\mathbf{A}$ is at most 13. □

The method of solution is based on the following observation. Let b_0 be another vector of the null-space (say c_0 and b_0 are the two eigenvectors of $\mathbf{A}^\top\mathbf{A}$ corresponding to the two smallest eigenvalues), then the desired solution vector b is a linear combination of the two:

$$b = b_0 + \alpha c_0 \tag{3.25}$$

We note that c_0 is in the null space of $\mathbf{A}$ regardless of noise in the data and will therefore correspond to an eigenvalue of zero (up to numerical precision). Theoretically, b_0 is also in the null space of $\mathbf{A}$ but only in the case of noise free data and exact motion model. Therefore b_0 is the eigenvector corresponding to the second smallest eigenvalue. In order to find α given c_0 and b_0 we enforce the constraint that $Rank[t'w''^T - t''w'^T] = 2$.

3.6.1 Solution Using the Rank=2 Constraint.

Clearly the choice of α will have no affect the first 6 elements of the vector b. Let us arrange the last 9 elements of b, b_0 and c_0 into the corresponding 3×3 matrices $\mathbf{B}$, $\mathbf{B_0}$ and $\mathbf{C_0}$. We are now looking for an α such that:

$$Rank(\mathbf{B_0} - \alpha\mathbf{C_0}) = 2. \tag{3.26}$$

In our case $\mathbf{C_0}$ is the identity matrix so the solution for α is given simply by the eigenvalues of $\mathbf{B_0}$. Since $\mathbf{B_0}$ is a 3×3 matrix this results in up to three discrete solutions. We now prove that only one of those can be a valid solution.

Theorem 1 *The 15-parameter model has a unique solution*
Let:

$$\mathbf{B} = \left[t'w''^T - t''w'^T\right]. \tag{3.27}$$

Let:

$$\tilde{\mathbf{B}} = \left[t'w''^T - t''w'^T\right] + \alpha\mathbf{I}. \tag{3.28}$$

Then the equation:

$$\tilde{\mathbf{B}} = \left[t'\tilde{w}''^T - t''\tilde{w}'^T\right] \tag{3.29}$$

has a solution for $\tilde{w}'', \tilde{w}'$ given t', t'' iff $\alpha = 0$.

Proof:

The *if* part of the proof is trivial. For $\alpha = 0$, $\tilde{w}'' = w''$ and $\tilde{w}' = w'$ is a solution for (3.29). To prove the *only if* part, suppose there exist $\tilde{w}'', \tilde{w}'$ that is a solution for (3.29). Then:

$$\left[t'w''^T - t''w'^T\right] + \alpha\mathbf{I} = \left[t'\tilde{w}''^T - t''\tilde{w}'^T\right]. \tag{3.30}$$

Let q be a vector such that $q^Tt' = 0$, $q^Tt'' = 0$ and $q \neq \vec{0}$. If t' and t'' are not parallel we can choose:

$$q = t' \times t''$$

Multiply left and right side of (3.30) by q^T to get:

$$q^T \left[t' w''^T\right] - q^T \left[t'' w'^T\right] + q^T \alpha \mathbf{I} = q^T \left[t' \tilde{w}''^T\right] - q^T \left[t'' \tilde{w}'^T\right]. \tag{3.31}$$

Since $q^T t' = 0$ and $q^T t'' = 0$ most of terms in (3.31) become zero and we are left with:

$$q^T \alpha \mathbf{I} = \vec{0}. \tag{3.32}$$

Since $q \neq \vec{0}$ then $\alpha = 0$. □

3.6.2 The Algorithm for Finding Motion Parameters

Based on the previous arguments the algorithm for finding the motion parameters is as follows:

1. Compute the $N \times 15$ matrix $\mathbf{A}$ from equation (3.20).

2. Find the eigenvector corresponding to the second smallest eigenvalue of the matrix $\mathbf{A}^T\mathbf{A}$. This is b_0. (The vector c_0 corresponds to the smallest.)

3. The first 6 elements of b_0 are the translations, t',t''.

4. Arrange the last 9 elements of b_0 into a 3×3 matrix $\mathbf{B_0}$.

5. Construct three possible rank 2 matrices $\mathbf{B}_{i,i=1..3}$ from $\mathbf{B}_0$:

$$\mathbf{B}_i = \mathbf{B}_0 - \alpha_i \mathbf{I} \tag{3.33}$$

   where $\alpha_{i,i=1..3}$ are the three eigenvalues of $\mathbf{B}_0$.

6. For each $\alpha_{i,i=1..3}$ solve:

$$[t' w''^T - t'' w'^T] = \mathbf{B},$$

   for w' and w'', given t' and t'' from step (3). This a set of 9 linear equations in the six unknowns (w',w'') and is solved using least squares.

7. From Theorem 1 only one of the three solutions is correct. For each of the three solutions use the 12 parameters t', t'', w' and w'' to form the 15 intermediate parameter vector b_i and then compute the error:

$$E_i = b_i^T \mathbf{A}^T \mathbf{A} b_i. \tag{3.34}$$

   Of the three solutions select the solution which gives the smallest error.

3.6.3 Solve as a Nonlinear Optimization

In Section 3.6.2 we use linear methods to solve for the motion parameters. In order to do so we have treated the 15 intermediate parameters as linearly independent parameters while they are in fact bilinear combinations of 12 independent parameters. It is often possible to improve the motion estimates by using the linear solution as a starting point for non-linear optimization techniques [15]. In our experiments we did not find that a nonlinear optimization stage improved the results. This is possibly because the iterative framework (Section 3.7.2) is in itself a form of nonlinear optimization.

The nonlinear optimization procedure is brought here for completeness. Our general problem is to find motion parameters t', t'', w', w'' that minimize the cost function:

$$E(t', t'', w', w'') = b^\top \mathbf{A}^\top \mathbf{A} b \tag{3.35}$$

where $\mathbf{A}$ is the estimation matrix derived from equation (3.20) and $b(t', t'', w', w'')$ is the fifteen element vector of bilinear functions of the motion parameters.

One method would be to find motion parameters t', t'', w', w'' that minimize equation (3.35). In order to avoid the trivial solution we need to add a constraint such as: $|t'|^2 + |t''|^2 = 1$. This becomes a 12 parameter nonlinear constrained optimization problem.

The process can be simplified by noting that equation (3.20) can be rewritten in the form:

$$\left(I_t'' - w''^\top v''\right) s'^\top t' - \left(I_t' - w'^\top v'\right) s''^\top t'' = 0. \tag{3.36}$$

given a set of rotation values w' and w'', one can compute the least squares estimate of t' and t'' using linear methods and also compute the least squares error (equation 3.35). One can then use nonlinear search techniques to find w' and w'' which minimize this least squares error. This becomes a six parameter unconstrained optimization problem.

3.6.4 Reduce to Pure Translation by Planar Alignment

As an alternative to solving the set of bilinear equations we can follow the ideas suggested by [36, 56, 40] (see also [60]) and first align a dominant plane in the image. After planar alignment the resulting image motion is due to pure translation, and we can use pure translation version of the tensor brightness constraint.

The resulting 3D geometry will be relative to the aligned plane. Therefore it is critical that the same image region (planar surface) is used to align both image motions. The resulting reconstruction will be a projective transformation of the true Euclidean shape and the motion field information will be correct. For some purposes such as finding correspondences or depth discontinuities this is sufficient. If needed the true (Euclidean) translation and rotation can be found as a second step [36].

3.7 Implementation Details

In this section we will provide the details needed to build a program that actually works.

3.7.1 Computing the Depth, Smoothing and Interpolation

After recovering the camera motion (Section 3.6.2) we use equations (3.16) and (3.19) to compute depth at every point. Information is combined from both image pairs by minimizing the least squares error:

$$E = \min_k \arg \sum_{j=1}^{2} \left(k s^T t^j + v^T w^j + I_t^j \right)^2 .$$

Here $j = 1$ and $j = 2$ denote values from the first and second image pairs respectively.

There are points in the image where the brightness gradients are close to zero, (and therefore $s^T t^j \simeq 0$) and the estimation of k will be ill-conditioned at those points. In order to overcome this problem we use a local region of support around the point of interest. It is simplest form we assume the depth is constant in the region and minimize:

$$E = \min_k \arg \sum_{x,y \in R} \sum_j \beta(x,y) |s^T t^j|^p \left(k s^T t^j + v^T w^j + I_t^j \right)^2 \tag{3.37}$$

where the windowing function $\beta(x, y)$ allows one to increase the weights of the points closer to the center of the region. The $|s^T t^j|^p$ term reduces the weight of points which have a small gradient or where the gradient is perpendicular to the camera motion since these cases are highly affected by noise. We used $p = 1$.

During the iteration process we typically used a region R of 7×7 to 11×11. For 'prettier' results the last iteration, we typically reduced the region R to 1×1 but added a very weak global smoothness term and performed multi-grid membrane interpolation[53]. The smoothness term stabilizes regions where there is no image gradient so very small regions of support can be used.

3.7.2 Iterative Refinement

The constant brightness constraint is a linearized form of the Sum Square Difference (SSD) criteria. The linear solution can be thought of as a single iteration of Newton's method applied to the problem. Iterative refinement is performed as follows:

1. Calculate motion (using equation 3.20).
2. Compute depth (using equation 3.37).
3. Using the depth and motion, warp Images 2 and 3 towards Image 1 (Section 3.7.3).

4. Compute new time derivatives I'_t and I''_t.

5. Compute a new motion and depth estimate.

In the ideal case, as the final result, the warped images should appear nearly identical to Image 1. One must be careful and not simply compute the incremental changes in the translation $\delta t'$ and $\delta t''$. As the images are warped closer together and the translation estimate approaches zero, the system of equations (3.20) will become ill conditioned. Furthermore, since the equations are homogeneous we must enforce a constraint such as $|t'|^2 + |t''|^2 = 1$ to avoid the trivial solution. We would not wish to apply such a constraint to $\delta t'$ and $\delta t''$. Thus, one must compute the full translation model (previous iteration plus the incremental change to the translation). This problem does not arise for the rotations: w' and w''. It is in fact convenient to warp the images using the best rotation estimate and then compute only the incremental improvement in the rotations: $\delta w'$ and $\delta w''$. For example we might chose to warp the images using a more exact rotation model than the linearized small rotation model.

The method for computing the full translation model and incremental rotation is now described. Let Ψ_1, Ψ_2 and Ψ_3 be the three images. Assume we have $\hat{k}$, $\hat{t}^j$, $\hat{w}^j$, from the previous iteration. (Here we use j to stand for either $'$ or $''$.) The translation components of image motion $\hat{u}_t^j$ and $\hat{v}_t^j$, and the rotational components $\hat{u}_r^j$ and $\hat{v}_r^j$ can be computed using $\hat{k}$, $\hat{t}^j$ and $\hat{w}^j$ in equation (3.14). These are then used to warp images Ψ_2 to $\hat{\Psi}_2$ and Ψ_3 to $\hat{\Psi}_3$. After warping, the images satisfy the brightness constraint equation:

$$\begin{aligned} I_x du' + I_y dv' + \hat{I}'_t = 0 \\ I_x du'' + I_y dv'' + \hat{I}''_t = 0 \end{aligned} \tag{3.38}$$

where the temporal derivatives at each pixel are given by:

$$\begin{aligned} \hat{I}'_t = \hat{\Psi}_2 - \Psi_1 \\ \hat{I}''_t = \hat{\Psi}_3 - \Psi_1 \end{aligned} \tag{3.39}$$

and du^j, dv^j are the (still unknown) differences between computed image motions and the real image motions:

$$\begin{aligned} du^j = u^j - \hat{u}_t^j - \hat{u}_r^j \\ dv^j = v^j - \hat{v}_t^j - \hat{v}_r^j \end{aligned} \tag{3.40}$$

Let:

$$\alpha^j = I_x \hat{u}_t^j + I_y \hat{v}_t^j \tag{3.41}$$

which can also be written as:

$$\alpha^j = \hat{k} s^T \hat{t}^j \tag{3.42}$$

Substituting equations (3.41) and (3.40) in equation (3.38) we get:

$$\begin{aligned} I_x(u' - \hat{u}_r^{'}) + I_y(v' - \hat{v}_r^{'}) + (\hat{I}_t' - \alpha') = 0 \\ I_x(u'' - \hat{u}_r^{''}) + I_y(v'' - \hat{v}_r^{''}) + (\hat{I}_t'' - \alpha'') = 0 \end{aligned} \tag{3.43}$$

Substituting equation (3.14) in equation (3.43) we get modified versions of the equations (3.16) and (3.19)

$$\begin{aligned} ks^T t' + v^T \delta w' + (I_t' - \alpha') = 0 \\ ks^T t'' + v^T \delta w'' + (I_t'' - \alpha'') = 0 \end{aligned} \tag{3.44}$$

We start our first iteration with $\hat{k}$, $\hat{t}^j$, $\hat{w}^j$ all zero and therefore $\alpha = 0$ as well.

3.7.3 Image Warping

Given an estimate of the camera motion and the depth at every point, we can warp Image 2 towards Image 1. Image warping in general is described in [87]. Let I'_{old} be the original Image 2 and let I'_{new} be the warped image we are trying to create. We must first define functions $x_{old}(x, y, ...)$ and $y_{old}(x, y, ...)$ which, given image coordinates in the new image (x, y) and possibly some extra parameters (symbolized by , ...), return the coordinates of the corresponding point in the old image (x_{old}, y_{old}).

In our case the *extra parameters* are the camera motions t' and w', and the depth map at every point $k(x, y)$. The functions x_{old} and y_{old} depend on the motion model. In our implementation we used the LH&P small motion model (equations (3.14)) but we could have also used the exact motion for calibrated cameras given in equations (2.15). Therefore:

$$x_{old} = x + u' = x + \tfrac{1}{z}(t'_1 - xt'_3) - w'_3 y + w'_2(1 + x^2) - w'_1 xy \tag{3.45}$$

and

$$y_{old} = y + v' = y + \tfrac{1}{z}(t'_2 - yt'_3) + w'_3 x - w'_1(1 + y^2) + w'_2 xy. \tag{3.46}$$

After defining x_{old} and y_{old} we can compute the value for every pixel in I'_{new} according to the formula:

$$I'_{new}(x, y) = \begin{cases} I'_{old}(x_{old}(x, y, ...), y_{old}(x, y, ...)), & if (x_{old}, y_{old}) \in I'_{old} \\ 0, & otherwise. \end{cases} \tag{3.47}$$

There are two points to note:

- The values of x_{old} and y_{old} are in general non-integer, so we use bilinear interpolation to compute the appropriate pixel values. Bilinear interpolation works better than nearest neighbor. The use of the more complex bicubic interpolation caused no noticeable improvement over bilinear interpolation.

- If the coordinates (x_{old},y_{old}) are outside the coordinates of the image then we use the value 0. It would make sense to mark these pixels as invalid and not use them in future motion and depth computation. This has not been implemented.

3.7.4 Coarse-to-Fine Processing

In order to deal with image motions larger than 1 pixel we use a Gaussian pyramid for coarse to fine processing [5][10]. Each of the three images is filtered by an approximation to a Gaussian filter and subsampled to create an image of half the size (in each dimension). This operation is performed recursively to create a pyramid with four or five levels depending on the size of the original image. The following 5×5 kernel was used as the filter:

$$\frac{1}{256} \times \begin{bmatrix} 1 & 4 & 6 & 4 & 1 \\ 4 & 16 & 24 & 16 & 4 \\ 6 & 24 & 36 & 24 & 6 \\ 4 & 16 & 24 & 16 & 4 \\ 1 & 4 & 6 & 4 & 1 \end{bmatrix}.$$

This filter is separable and can be implemented by convolving the rows and columns of the image, twice each, with the filter $f = (0.25,\ 0.5,\ 0.25)$.

Starting from the coarsest level of we perform a few iterations to compute motion and depth according to the scheme described in Section 3.7.2. The brightness derivatives (I_x, I_y, I'_t, I''_t) are computed using the subsampled images. After computing the motion and depth, we create a finer depth map from the coarse depth map using bilinear interpolation. We use the interpolated depth map, and the motion estimates, as the starting values for the iterations at the finer level.

As we reduce the resolution of the image we must also scale the focal length accordingly (divide by 2). As we move to a finer level of the depth map, the depth values and motion estimates do not need scaling.

3.7.5 Field of View

In this section we will first demonstrate how in the general problem of determining camera motion from motion fields, when the field of view is narrow, there exists an ambiguity between rotation and translation. We will also note the errors in the estimation of the rotation if we use the wrong focal length value. Then we show how this ambiguity is observed in the shape of the cost function (equation 3.35) from three views. Finally we describe how to achieve stable and accurate motion estimates in the presence of rotation even without very wide fields of view.

Rotation-Translation Ambiguity from Motion Fields

It is well known that for a medium to narrow field of view, the motion field due to camera rotation around the Y axis, can be indistinguishable from the motion field due to some translation along the X axis and an appropriate depth surface. A similar ambiguity exists between rotations around the X axis and translations along the Y axis.

In Figure 3-4 we see that for medium-narrow fields of view (30^o) the flow due to rotation around the Y axis is parallel to the X axis. The length of the flow vectors due

to rotation is slightly shorter, closer to the Y axis. A similar flow field can be produced by camera translation along the X axis with a surface that is curved so that points in the center of the image are more distant. There are motion flows that unambiguously indicate a translational component to the motion: motions fields that include a focus-of-expansion or show parallax effects. Nevertheless even these fields are ambiguous since a rotational component could be added to the motion, compensated for by a change in the translation and depth to produce the same motion field. In some case the resulting depth map will include negative depths indicating some obvious error. This is often formulated as the *depth positive constraint* which can be used to limit the range of motion ambiguity.

With wider fields of view (i.e. shorter focal lengths) the ambiguity disappears. The motion fields due to rotation cannot be modeled by translational motion. Figure 3-4 also shows the motion fields due to rotation with fields of view of 53^o and 90^o. The distribution of the Y component of the flow has the unique characteristics of rotational flow. For wide fields of view the Y component points towards the X axis on the right of the image and away from the X axis on the left of the image (for this particular sign of the rotatation).

The motion field for a wide field of view is unambiguous if we know the focal length, or equivalently, the field of view (i.e. the camera's internal parameters are known). If our estimate of the focal length is incorrect then the motion and structure estimates which best fit the flow field will be incorrect. If the focal length estimate is smaller than the true focal length then estimate angle of rotation estimate will be smaller than the true rotation angle (in absolute values). Figure 3-5a shows the X and Y components of the motion field of a rotation of -4.5^o around the Y axis using a focal length of 400_{pixels}. Figure 3-5b shows the same measurements for a focal length of 200_{pixels} and a rotation of -2.5^o. The Y components are similar. The X components are very different but these can be accounted for by adjusting the translation and depth estimates.

Rotation-Translation Ambiguity in the 15-parameter Model

The rotation translation ambiguity can also be observed while using the 15-parameter model. Figure 3-6 show the shape of the cost function (equation 3.35) near the global minimum. These are simulation results for focal lengths of 50, 100 and 200. The image size was 320×240. It is not possible to plot the full 12 dimensional surface so the figures show only the shape of the function as we vary two motion parameters at a time. We show the effect of varying w'_x and t'_y where we expect to see ambiguity and the effect of varying w'_x and t'_x where no ambiguity is expected. In Figure 3-6f, the long diagonal valley shape of the cost function indicates the ambiguity of the motion estimate for narrower fields of view.

The Solution

As we have seen, in order to get reasonable rotation and translation estimates a wide field of view is required. The theoretical results of [46] and simulation results indicate

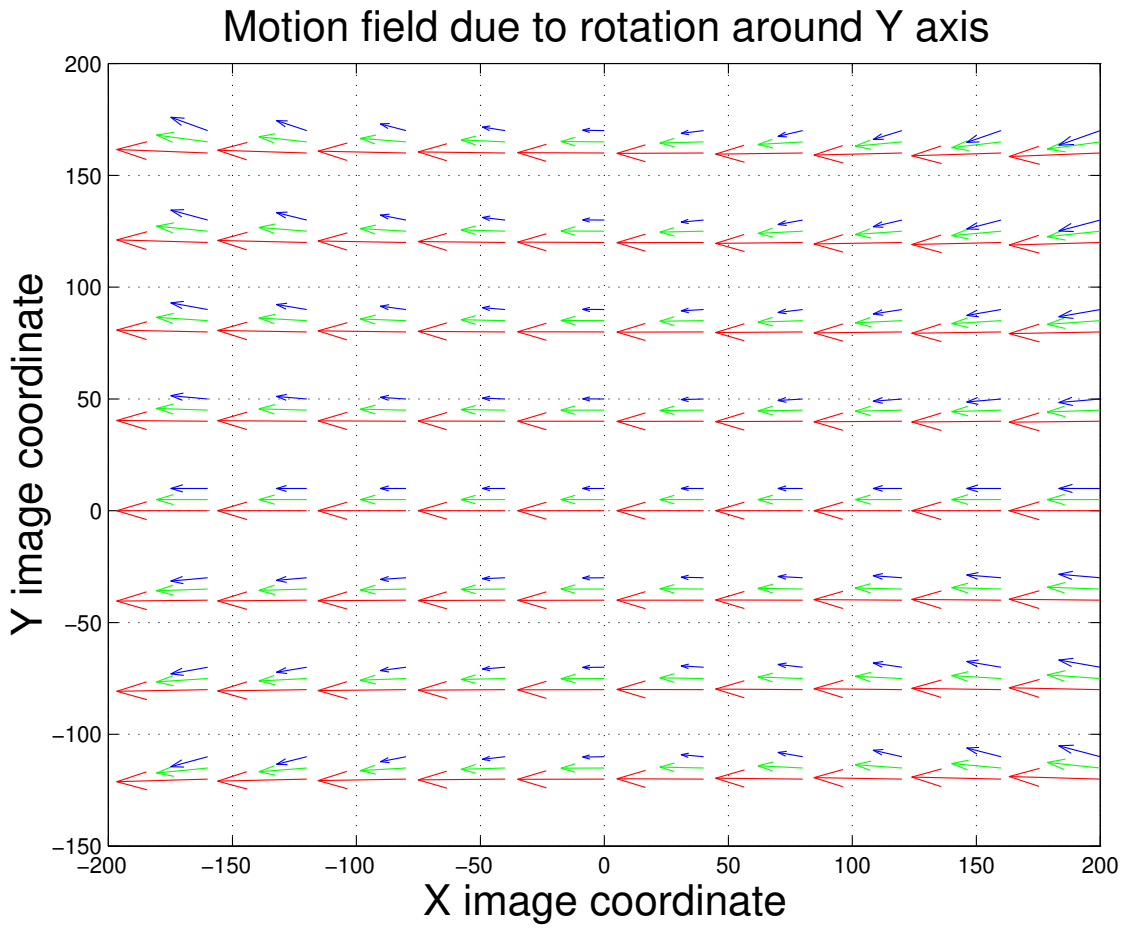
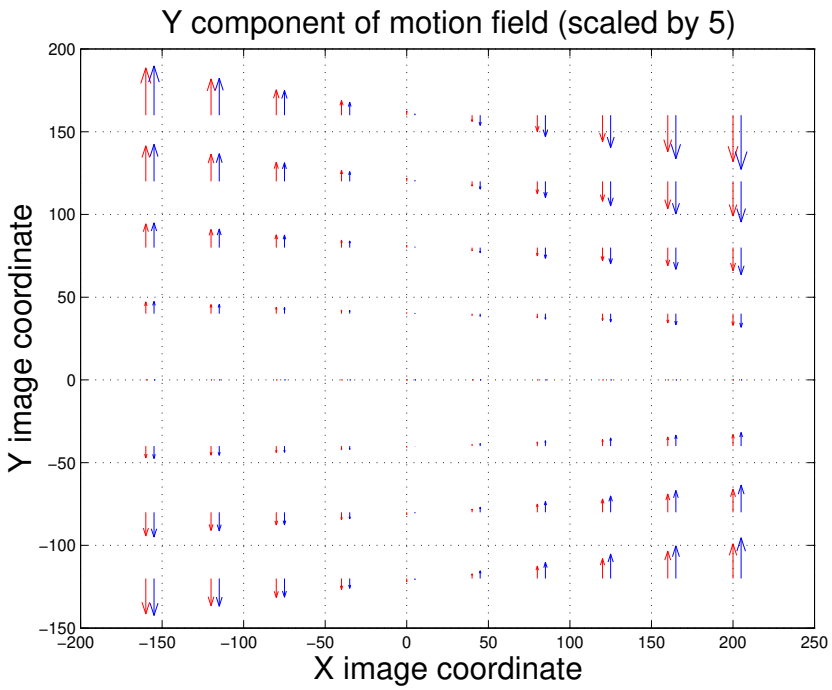
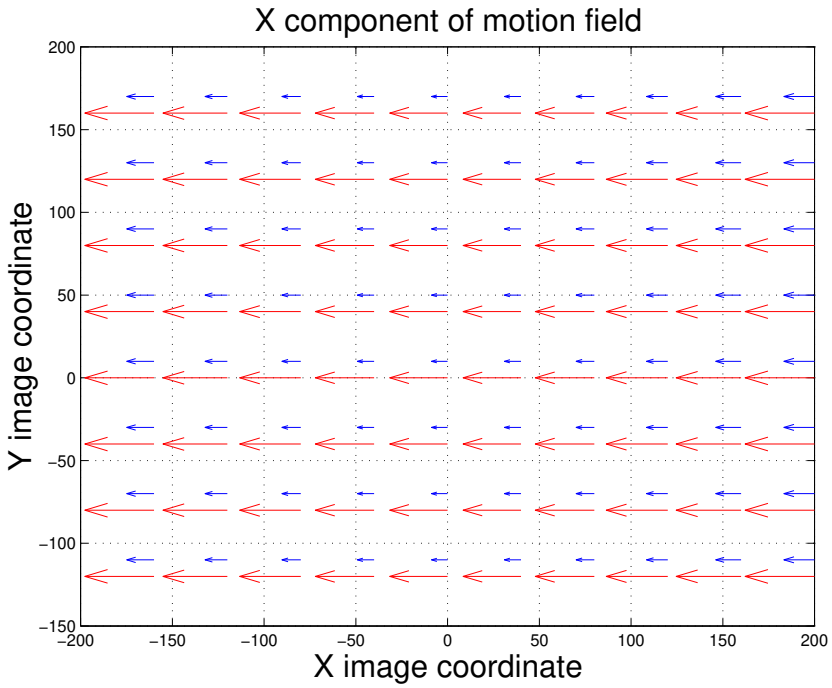


Figure 3-4: *Motion flow field due to rotation around the Y axis for wide, medium and narrow fields of view. The arrows for the medium and wide have been offset in the Y direction for clarity. For the narrow field of view ($f = 800_{pixels}$, FOV=30^o) the flow field is large and almost parallel to the X axis. For wider fields of view ($f = 400$ and $f = 200$) the X component of the flow is smaller but there is a noticeable Y component.*



(a)



(b)

Figure 3-5: *The X and Y components of the motion field due a rotation around the Y axis of -4.5^o with a focal length of 400_{pixels} for a rotation of -2.5^o with a focal length of 200_{pixels}. The arrows for the 200_{pixels} camera have been plotted offset right (a) and up (b). Note that the Y flow components are very similar.*

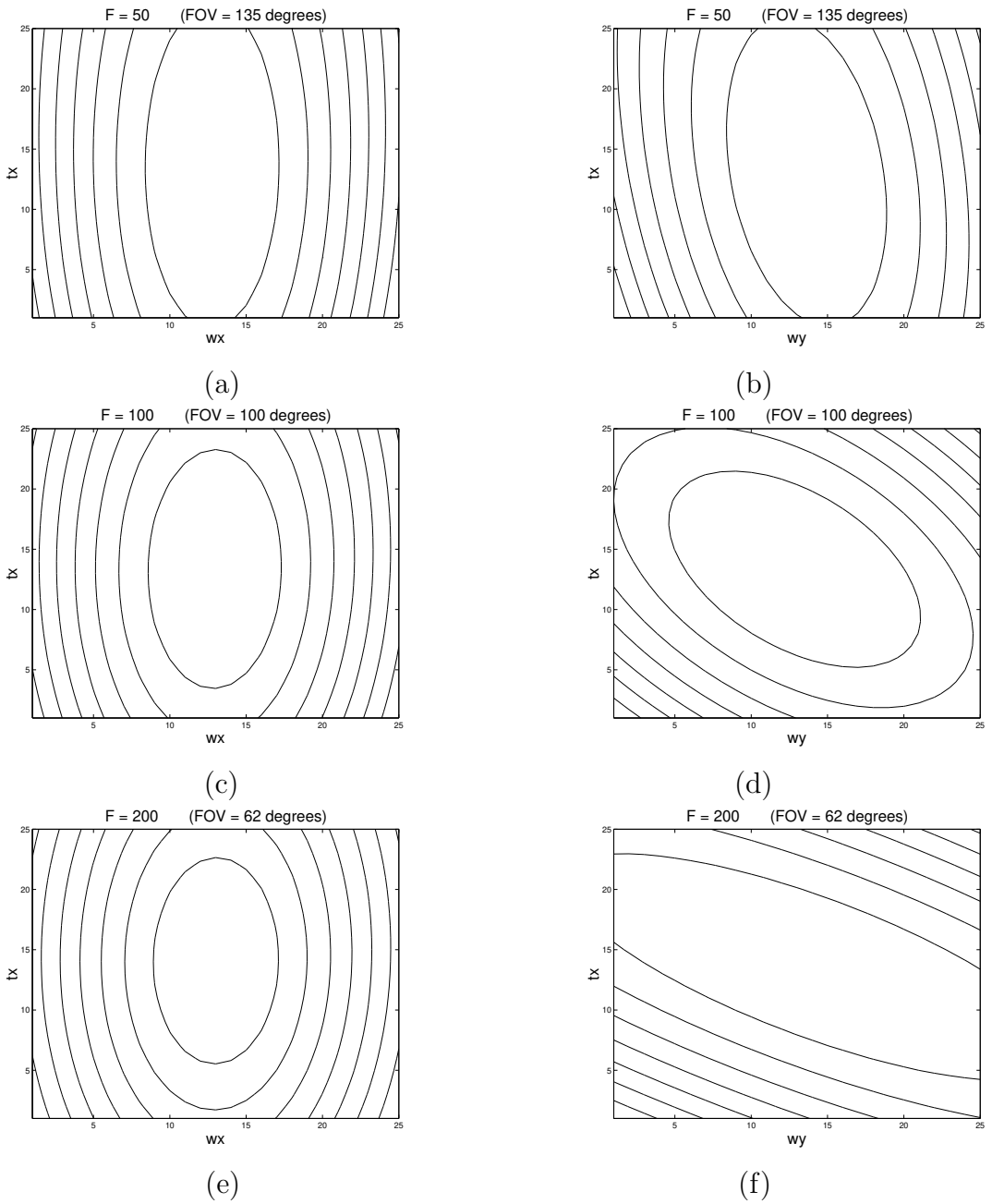


Figure 3-6: *The shape of the cost function around the minimum for various fields of view. These simulation results were computed after the program converge to the final motion and depth estimates. For visualization only two of the 12 parameters were changed at one time. Note that for a narrower field of view (f) the cost function has the shape of a narrow diagonal valley. This indicates an ambiguity between translation along the X axis and rotations around the Y axis.*

that very wide fields of view are required (120^o or greater). These are hard to achieve with standard lenses. With narrower fields of view the iterative scheme described in Section 3.7.2 fails to converge. In previously published work [74] we managed to stabilize the results for small rotation angles by unknowingly biasing the results towards small rotation. This was done by setting the focal length parameter to 50 instead of around 600. For a 640×480 size image this implied a field of view of 150^o instead of the correct 55^o. As a result the rotation estimates were considerably smaller that the true values. This also resulted in errors in the translation and depth estimates.

It is important to note that while the magnitude of the angle estimates in [74] were too small, the estimated axis of rotation was correct. In fact the results show a linear relationship between true angle and estimated angle. That the rotation values were always smaller than the true values is consistent with the simulation results shown in Figure 3-5. This leads us to the following simple modification to the iterative scheme (Section 3.7.2):

1. For the estimation of motion use a focal length parameter $f << f_{true}$.

2. During the image warping stage use the correct focal length to rotate Images 2 and Image 3 towards Image 1.

This modification required no changes to the structure of the algorithm since the coarse-to-fine and iterative mechanisms were already in place. The results shown in Section 4.3.1 show that the modified algorithm gives accurate rotation estimates even for medium fields of view (55^o).

How does this affect the translation estimates ?

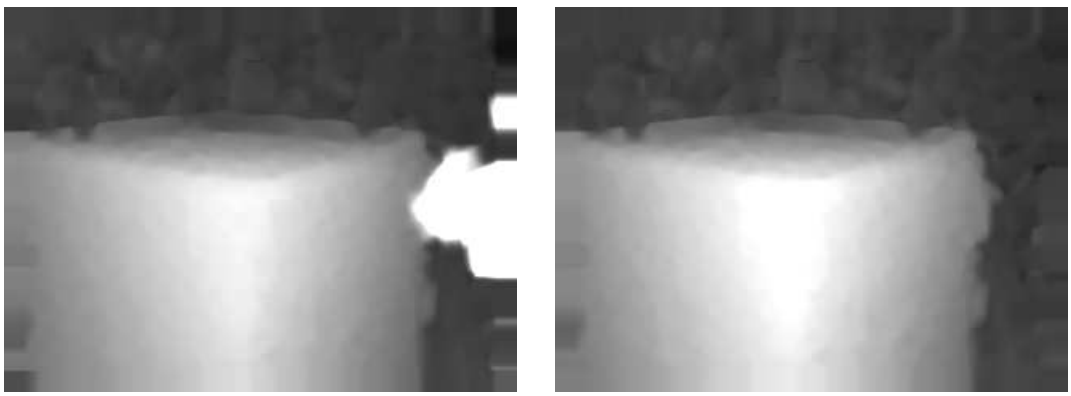
How does using the wrong focal length affect the translation estimates ? Looking at equation (3.20) (and also (3.16), (3.19) and (3.37)) we see that the translations t' and t'' always appear in a dot product with s where:

$$s = \begin{pmatrix} I_x \\ I_y \\ -xI_x - yI_y \end{pmatrix}.$$

The first and second terms of s are not affected by the focal length estimate. Therefore there is no effect on the translation estimate in the X and Y direction, t_x and t_y. In the third term, x and y are supposed to be the normalized image coordinates. If we use the wrong focal length then the estimated Z translation (t_z) will be scaled accordingly.

$$\frac{t_Z}{t_{Ztrue}} = \frac{f}{f_{true}}.$$

Using this simple relationship we can recover the true translation direction from the estimated one even if we use a focal length value which is much smaller than the true value.



(a) (b)

Figure 3-7: *An example of a local depth error (a) fixed by a second pass down the pyramid performing 'depth-only' computation (b). (see text)*

3.7.6 Final Touches - Recompute the Depth Keeping the Motion Constant

Initially, during the first iterations and at the coarsest levels of the pyramid, the motion estimates are not accurate and cause errors in the depth estimates. Sometimes, particularly near the borders of the image, these errors are large enough to affect the new derivatives, I'_t and I''_t in such a way that at the next iteration even when the motion estimates are better the depth values are still badly in error. This tends to be a local problem and does not affect the global motion estimates but it does create local 'holes' in the depth map.

To fix this problem, after computing the motion at the finest level, we go back down the pyramid and recompute the depth while keeping the motion estimate constant. This is in effect becomes a three camera stereo computation. Figure 3-7 shows an example of the depth estimates before and after the second pass.

3.8 Summary of Theoretical Results and Future Work

We have presented a general relationship between the spatio-temporal derivatives of three frames and the ego-motion parameters of the two motions. This relationship was derived first for the general case where the ego-motion model comprises of the 27 coefficients of the trilinear tensor, and then for the small-motion model of Longuet-Higgins & Prazdny.

On practical grounds, these relationships are model-based brightness constraints which provide a linear constraint per pixel in the image — thereby providing a direct structure and motion estimation that cuts through the aperture problem and without prior detection of feature points. The linear constraints of the 27-parameter projective model are degenerate and cannot lead directly to a unique solution for the tensor

coefficients although a unique solution can be found using quadratic admissibility constraints. This added complexity lead us to implement in practice the simpler small-motion model of Longuet-Higgins & Prazdny.

The implementation details of these model-based brightness constraints are important and include four critical elements: (i) embedding of the computations within a coarse-to-fine (Gaussian pyramid) framework, (ii) Newton iterations over the brightness constraint equation, (iii) post-processing of smoothness and surface interpolation for obtaining visually pleasing results, and (iv) obtaining both stability and accuracy by assuming a very short focal length for the motion estimation, but using the correct focal length for the image warping.

In the following chapter we test the algorithm on a set of challenging real image situations, some of which contain very few "good features" (local regions with significant variability of gradient direction) that are necessary for optical flow and discrete point matching algorithms. Yet, we obtained a faithful dense depth map of the scene due to the fact that the algorithm is not hindered by the presence of aperture effects.

3.8.1 Future Work - Collinear Motion

The method, as described so far, fails when the two motions are in the same (or opposite) directions. This is a drawback in many applications (e.g. 3D reconstruction from a monocular image sequence). We present here some early research on ways to overcome this problem. The initial results look promising (Section 4.2.2). We will investigate the pure translation case because if it fails for pure translation, it will also fail if there is some rotation.

For pure translation:

$$I''_t s^\top t' - I'_t s^\top t'' = 0 \tag{3.48}$$

and

$$k s^\top t'' + I''_t = 0 \tag{3.49}$$

$$k s^\top t' + I'_t = 0 \tag{3.50}$$

If the second translation vector t'' is proportional to the translation vector t' then equation (3.49) is simply a scaled version of equation (3.50) adding no new information and the solution is ill conditioned. Another way of interpreting equations (3.50) and (3.49) is that a motion t' in one direction will create a change in the image I'_t which exactly the opposite of the change I''_t induced by a motion t'' of equal magnitude and opposite direction to t'.

The LH&P model which assumes ($\frac{t_z}{Z} << 1$) and was used to derive equations (3.50) and (3.49) is of course not exact unless the motion in the Z direction is zero, but in any case the contribution to the optical flow due to translation in the Z direction is small unless the field of view is very wide. Therefore, for small motion, the LH&P model will be quite accuarte and the equations will be ill-conditioned.

This situation is closely related to the case of reconstruction from line correspondences. There is a critical combination of a line configuration and camera motions which can be stated in the following theorem:

Theorem 2 *Critical configuration of lines and motion*

Let $\mathcal{S}$ be a set of lines in 3D which have a common intersecting line L (i.e., $S \wedge L = 0$ for all $S \in \mathcal{S}$). Let s,s′,s″ be the projections of the line $S \in \mathcal{S}$ in the three views: Image 1, Image 2 and Image 3. Let o, o′ and o″ be the corresponding camera centers. If o, o′ and o″ all lie on a line $S_i \in \mathcal{S}$ the corresponding lines s,s′,s″ do not provide a unique solution to the camera motion.

We currently have no proof, but simulation experiments confirm the theorem. This line configuration is the Linear Line Complex (LLC) which is examined in Chapter 6. In Chapter 6 (Theorem 4) we show that for general motion, the tensor has a linear degeneracy in the case of an LLC, but a unique solution is obtained by taking into account nonlinear constraints. Here, for the specific case where the camera motion is along a line belonging to the LLC, there is a whole family of solutions. The epipole could lie anywhere on the projection of the line L onto the image. Thus, there is one degree of uncertainty in addition to the scale factor ambiguity.

If we take the common to be the line at infinity on the XY plane we get the particular case where all the lines lie on planes which are parallel to the image plane, and the translations are collinear and also parallel to the image plane. This is a critical condition according to Theorem 2 and the direction of translation in the XY plane cannot be recovered. In such a configuration, lines in Image 1 will be parallel to corresponding lines in Image 2 and Image 3 (assuming no rotation).

How does this relate to our case? By using the first order approximation of the optical flow constraint equation to get the point-line-line correspondences, we have created, as an artifact, a situation where all corresponding lines are parallel. In reality this is not the case, unless, all the lines came from planes parallel to the image plane.

A Possible Solution

The Optical Flow Constraint Equation (Section 2.5) was based on the first order Taylor expansion of $I(x, y, t) = I(x', y', t + \delta t)$ where t is used to denote time (not translation). Keeping the second order terms of the Taylor expansion results in:

$$I(x + \delta x, y + \delta y, t + \delta t) = I(x, y, t) + \delta x I_x + \delta y I_y + \delta t I_t + \frac{1}{2!}\left(\delta x^2 E_{xx} + \delta y^2 E_{yy} + \delta t^2 E_{tt} + 2\delta x \delta y E_{xy} + 2\delta x \delta t E_{xt} + 2\delta y \delta t E_{yt}\right) + e \quad (3.51)$$

Replacing $\delta x = u'\delta t$ and $\delta y = v'\delta t$ yields:

$$I(x + u\delta t, y + v\delta t, t + \delta t) = I(x, y, t) + u'\delta t I_x + v'\delta t I_y + \delta t I_t + \frac{1}{2!}\left((u\delta t)^2 E_{xx} + (v'\delta t)^2 E_{yy} + \delta t^2 E_{tt} + 2u'v'\delta t^2 E_{xy} + 2u'\delta t^2 E_{xt} + 2v'\delta t^2 E_{yt}\right) \quad (3.52)$$

We notice that two of the second order terms are also linear with u and v. If we keep those two terms, (at this point, apart from convenience, we have no formal justification why we should keep those two terms and not all the 2nd order terms) and apply the constant brightness assumption:

$$I(x, y, t) = I(x + \delta x, y + \delta y, t + \delta t)$$

we get for the first motion:

$$u'(I_x + I_{xt}\delta t) + v'(I_y + I_{yt}\delta t) + I'_t = 0 \tag{3.53}$$

Since $I'_x = I_x + I_{xt}\delta t$ and $I'_y = I_y + I_{yt}\delta t$ this becomes:

$$u'I'_x + v'I'_y + I'_t = 0 \tag{3.54}$$

(The original work by Horn & Schunk used: $u'\frac{(I'_x+I_x)}{2} + v'\frac{(I'_y+I_y)}{2} + I'_t = 0$). Note that I'_x and I'_y are computed at coordinates (x, y) not (x', y'). Similarly for the second motion:

$$u''I''_x + v''I''_y + I''_t = 0 \tag{3.55}$$

Likewise I''_x, I''_y are the brightness gradients in Image 3 at coordinates (x, y). That leads us to the equation:

$$I''_t s'^{\top} t' - I'_t s''^{\top} t'' = 0 \tag{3.56}$$

where:

$$s' = \begin{pmatrix} I'_x \\ I'_y \\ -xI'_x - yI'_y \end{pmatrix} \qquad d'' = \begin{pmatrix} I''_x \\ I''_y \\ -xI''_x - yI''_y \end{pmatrix} \tag{3.57}$$

Since now $s' \neq s''$ the equations are better conditioned. This depends on how much change there was in the images gradients. As we have noted, this depends on whether the surface is parallel to the image plane or not.

A modification is required to the coarse-to-fine framework and the iterative refinement which are described in Section 3.7.4 and Section3.7.2 respectively. For motion estimation, Image 1 is warped (using forward warping) towards Image 2 and Image 3. For depth estimation Image 2 and Image 3 are warped towards Image 1.

We have shown an initial solution to the case of collinear motion with pure translation and have successfully implemented these ideas. It is not clear how to transfer this to the general motion case and this remains an open area of study.

Chapter 4

Direct Estimation of Motion and Structure - Experiments

4.1 Overview

In this section we perform experiments which test various aspects of direct estimation of motion and structure. Experiments 1,2 and 3 deal with the pure translation case. In experiment 1, we test the accuracy of heading estimation for non-collinear motion. Experiment 2 tests the accuracy heading estimates obtained from collinear motion. Experiment 3 tests a standard optical flow program on some of the same input images and compares the results with the 'direct methods'.

We then move on to case where the motion involves both translation and rotation. Experiment 4 revisits the image sequence used in Stein & Shashua [74]. We show that the improvements described in Section 3.7.5 greatly improve the rotation estimates. In Experiment 5 we repeat the heading estimation experiments but this time the motion includes some rotation. We also show how not taking into account even small rotations ($\sim 0.5^o$) leads to large errors in heading estimates. In Experiment 6 we test the accuracy of Euclidean reconstruction using the simple image of a cube. We measure whether the right-angles between the cube faces are correctly recovered.

In all the preceding experiments the camera was mounted on a motion stage so that ground truth motion could be known accurately. In Experiment 7 we repeat the Euclidean reconstruction experiments but this time with a hand-held camera. We show that if we neglect the rotation the shape recovery is noisy. We also show that if we neglect radial lens distortion then the reconstruction is qualitatively good but the Euclidean measurements, such as angles, are less accurate. This agrees with the observations regarding lens distortion and the trilinear tensor in Chapter 7.

4.2 Pure Translation

4.2.1 Experiment 1: Accuracy of Heading Estimates for Pure Translation

In this experiment we measure the accuracy of the estimation of the camera motion direction (heading direction) in the case of pure translation. We also gauge the effect

of the non-linear lens distortion.

Experimental Procedure:

The camera was mounted on a motion stage with three degrees of computer controlled motion (Figure 4-1): horizontal translation, vertical translation and rotation around the camera's Y axis. The axes of motion are highlighted in Figure 4-1a.

The camera Y axis was aligned with vertical axis of the stage. Initially the camera was positioned so that the optical axis was aligned with the horizontal axis of the translation stage. The accuracy of the alignment was $\pm 1^o$. The camera was rotated α^o and an image captured. Then the camera was translated vertically $12.5mm$ and an image captured. Finally the camera was translated horizontally $10mm$ and a third image captured. Thus we captured an image triplet where the camera motion was pure translation with one of the motions vertical and the second at a heading of α^o. This was repeated for angles $\alpha = 0^o, 10^o, \ ... \ , 90^o$.

Figures (4-2a,b,c) show the image triplet for a heading angle of 40^o. Motion and depth estimates were performed with and without lens distortion correction. The depth map estimate is shown in figure (4-2d). The camera lens was (4.9mm lens / 82^o FOV). The experiment was then repeated with a medium-wide angle lens ($8.5mm$ / 52^o FOV) for the scene shown in figure (4-15).

Results:

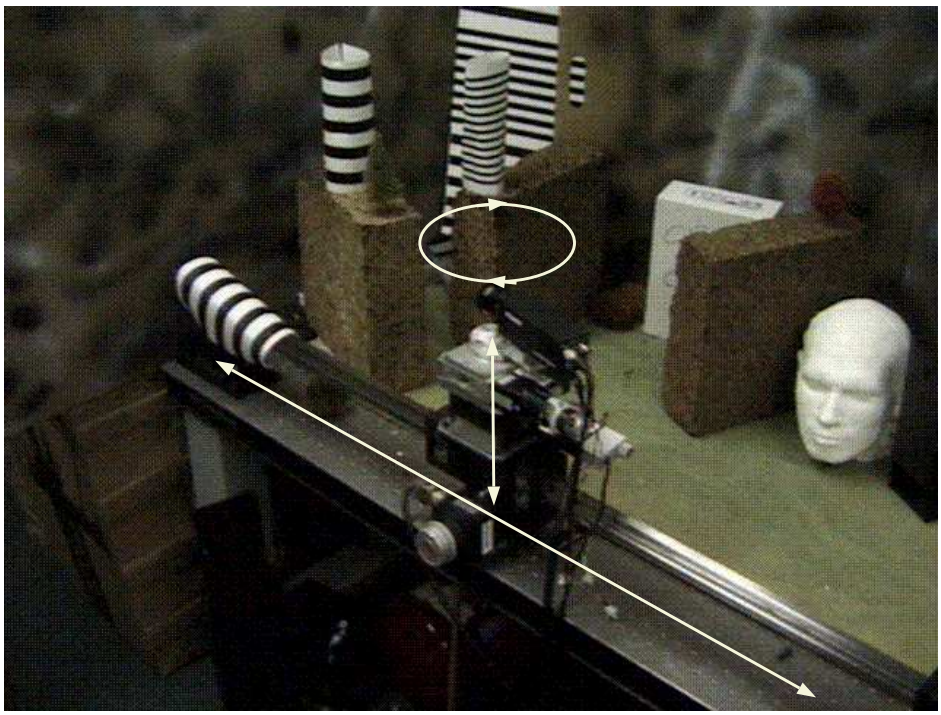
Figure 4-3 and Figure 4-4 show the heading estimates for the wide and medium-wide angle lens respectively. The heading estimates are within a few degrees. Notice that when the true heading was inside the FOV the errors were less than 1^o. When the true heading was outside the FOV the errors increased to $1 - 2^o$. In the case of the wide angle lens the errors were considerably larger when lens distortion was not taken into account (dashed line in Figure 4-3b).

4.2.2 Experiment 2: Collinear Motion with Pure Translation

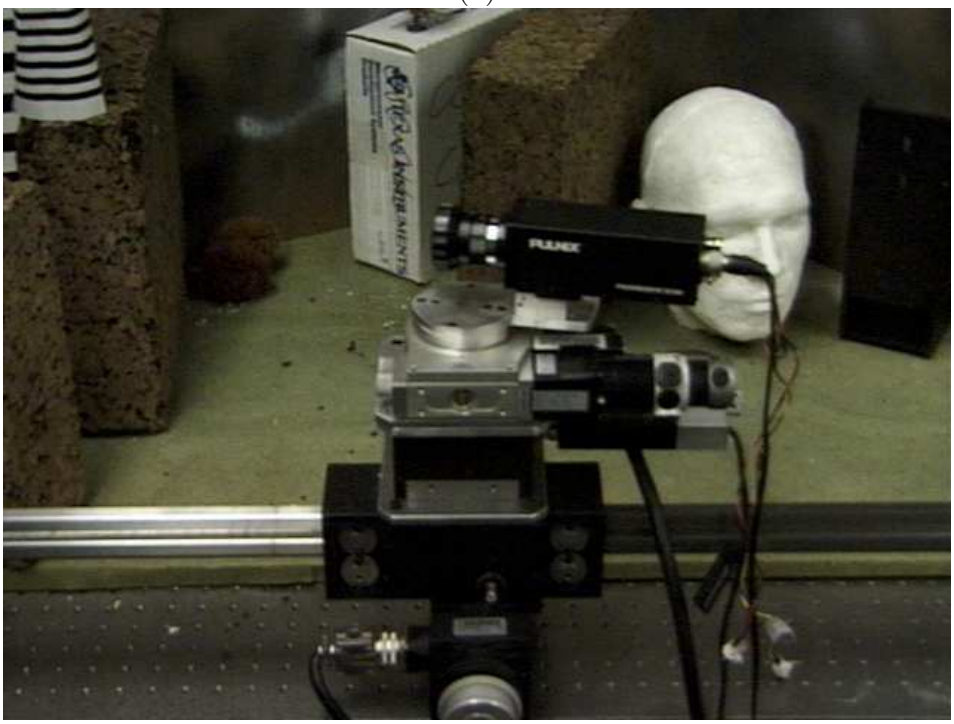
This experiment aims to test the accuracy of motion heading estimation using collinear motion as opposed to non-collinear motion. We also qualitatively compare how the two types of motion affect the structure estimates and convergence of the algorithm. The algorithm is tested against a scene which is challenging for optical flow based techniques because it lacks corner-like feature points.

Experimental Procedure:

The camera was mounted on the motion stage as described in Experiment 1 (Section 4.2.1). The vertical axis of the camera coordinate system was aligned with the vertical axis of the motion stage. The camera was rotated to a given camera heading and was then moved with three translational motions as shown in Figure 4-5. The first motion was vertical, the other two were equal horizontal motions in opposite directions. By choosing the first and second motions we have non-collinear motion.

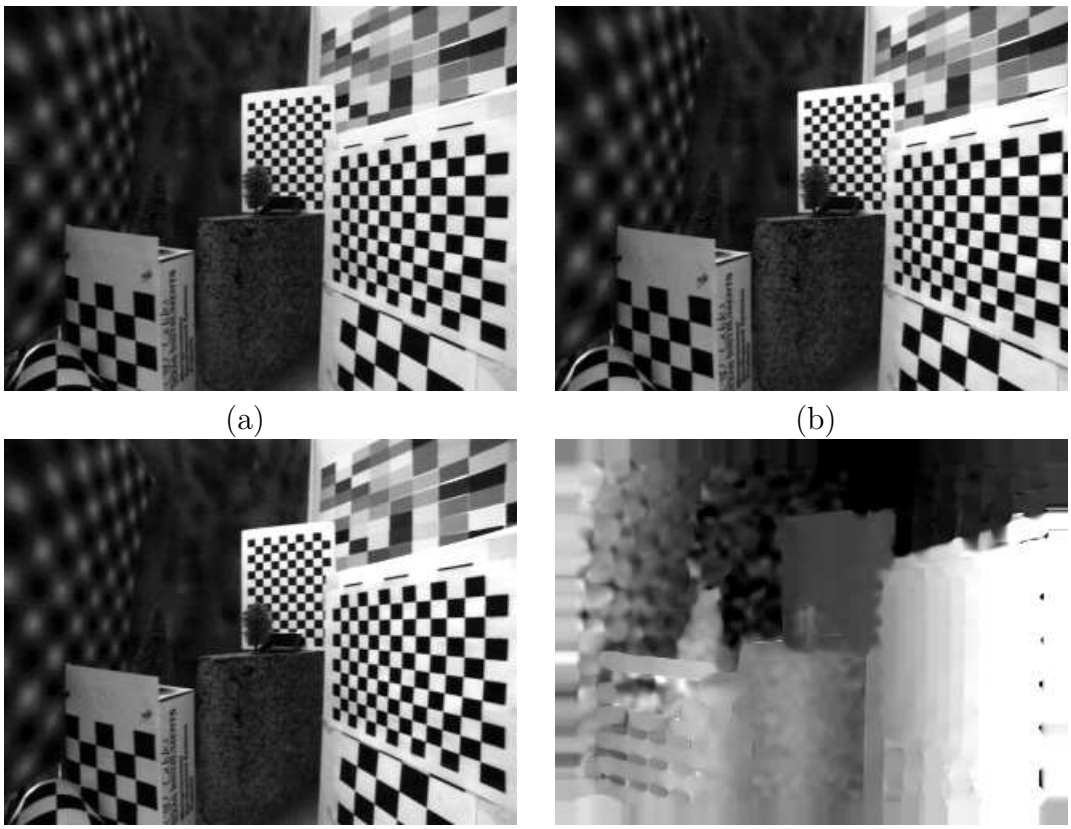


(a)



(b)

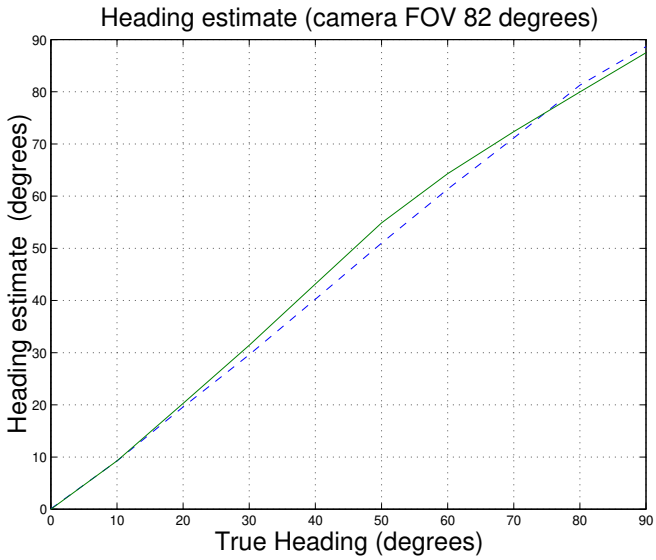
Figure 4-1: *Two views of the optical bench and the camera mounted on the motion stage. The three axes of motion are highlighted in (a). The objects on the table are those that appear in Figure 4-6.*



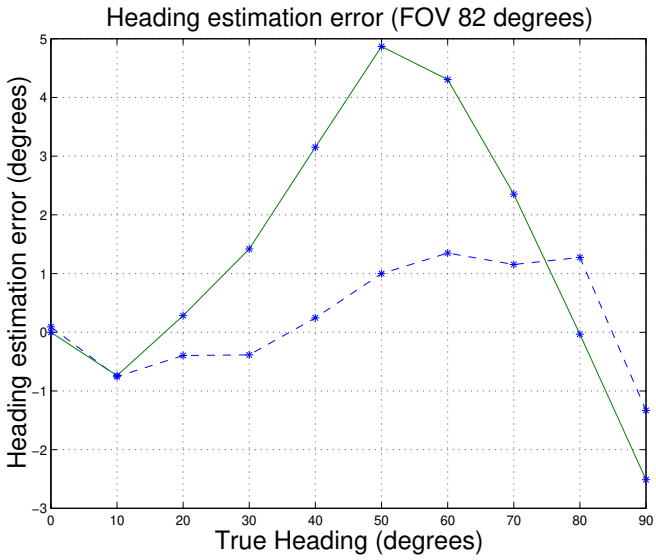
(a) (b)

(c) (d)

Figure 4-2: *The three input images (a,b,c) and the estimated depth map (d). The motion between images (a) and (b) is horizontal, forward and* 40^o *to the left. The motion between images (a) and (c) is vertical. (*4.9*mm lens /* 82^o *FOV)*



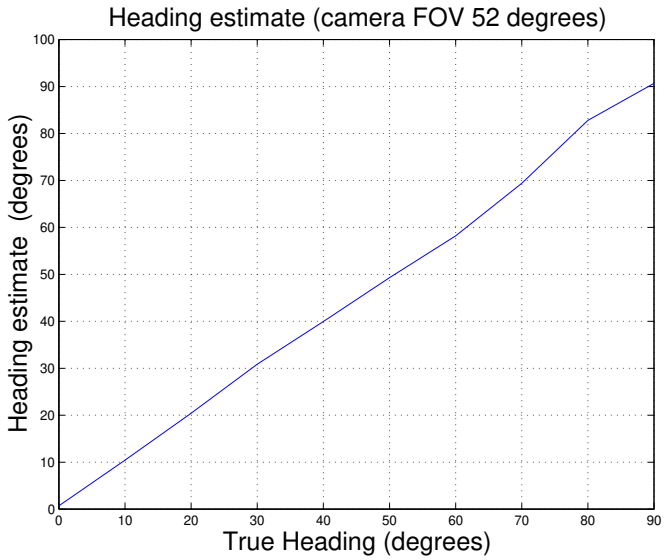
(a)



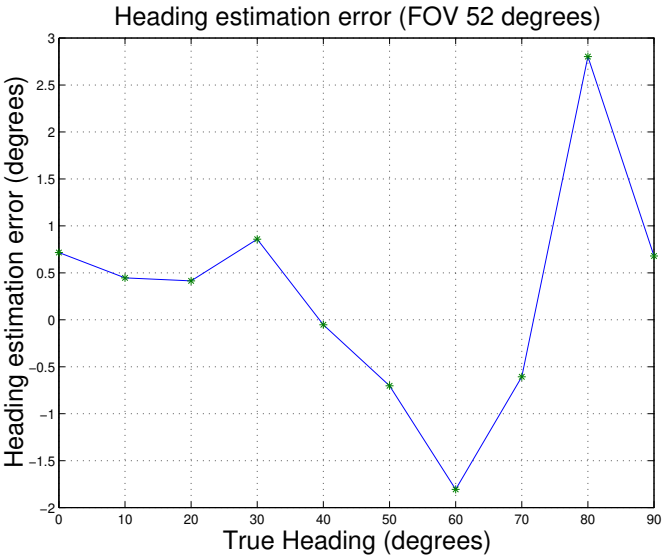
(b)

Legend:	
——	No radial distortion correction
- - -	Images corrected for radial distortion

Figure 4-3: *(a) Heading estimate from a translating camera. Pure translation was assumed. The* 4.9_{mm} *lens gives a* 82^o *FOV along the X axis. (b) Difference between heading estimate and motion stage reading ("True Heading").*



(a)



(b)

Figure 4-4: *(a) Heading estimate from a translating camera. Pure translation was assumed. The* 8.5_{mm} *lens gives a* 52^o *FOV along the X axis. (b) Difference between heading estimate and motion stage reading ("True Heading").*

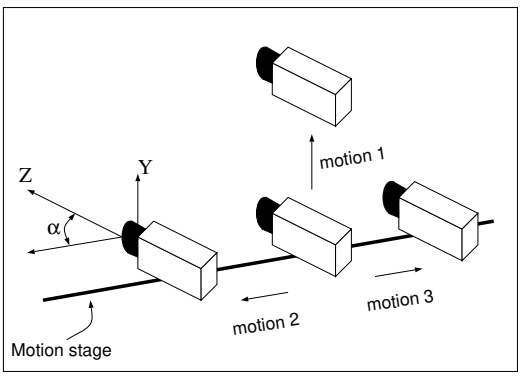


Figure 4-5: *Schematic diagram of the pure translation experiment. Camera heading is set by the angle α. Motion 2 and Motion 3 are collinear.*

By choosing the second and third motion we have collinear motion. A typical set of four images is shown in Figure 4-6 for a heading direction of 30^o.

Results:

Figure 4-7 shows the motion direction estimates using the non-collinear image triplets or the collinear triplets. In both cases the heading estimate is within a few degrees of the true stage motion.

Figure 4-6e and Figure 4-6f show the depth maps estimates for the non-collinear and collinear cases. The results are shown for the horizontal motion forward and to the left at an angle of 30^o to the camera optical axis. The FOE in this case was inside the image. As expected, at in the case of collinear motion the depth map around the epipole is not accurate since the image motion is very small. There are also errors in the depth estimates on the cylinders where the tangents to the iso-brightness contours pass through the epipole. These errors do not appear when the two motions are non-collinear because each motion can give depth estimates where the other motion fails.

The algorithm is very robust for non-collinear motion and will converge for large and small motions (for the whole tested range of $5mm$ to $40mm$) and a variety of parameter settings. For the collinear motion the algorithm would only converge for a smaller motion range ($5mm$ to $10mm$).

Figure 4-8 shows another example of 3D reconstruction using collinear motion. In this example the motion was vertical. There are no areas in the image where the edges are predominantly vertical so the depth estimates are quite good throughout the image.

4.2.3 Experiment 3: Comparison with Optical Flow Techniques.

After computing the motion and depth from an image triplet we can compute the motion flow using the motion equations (equation 3.14). In this experiment we compare

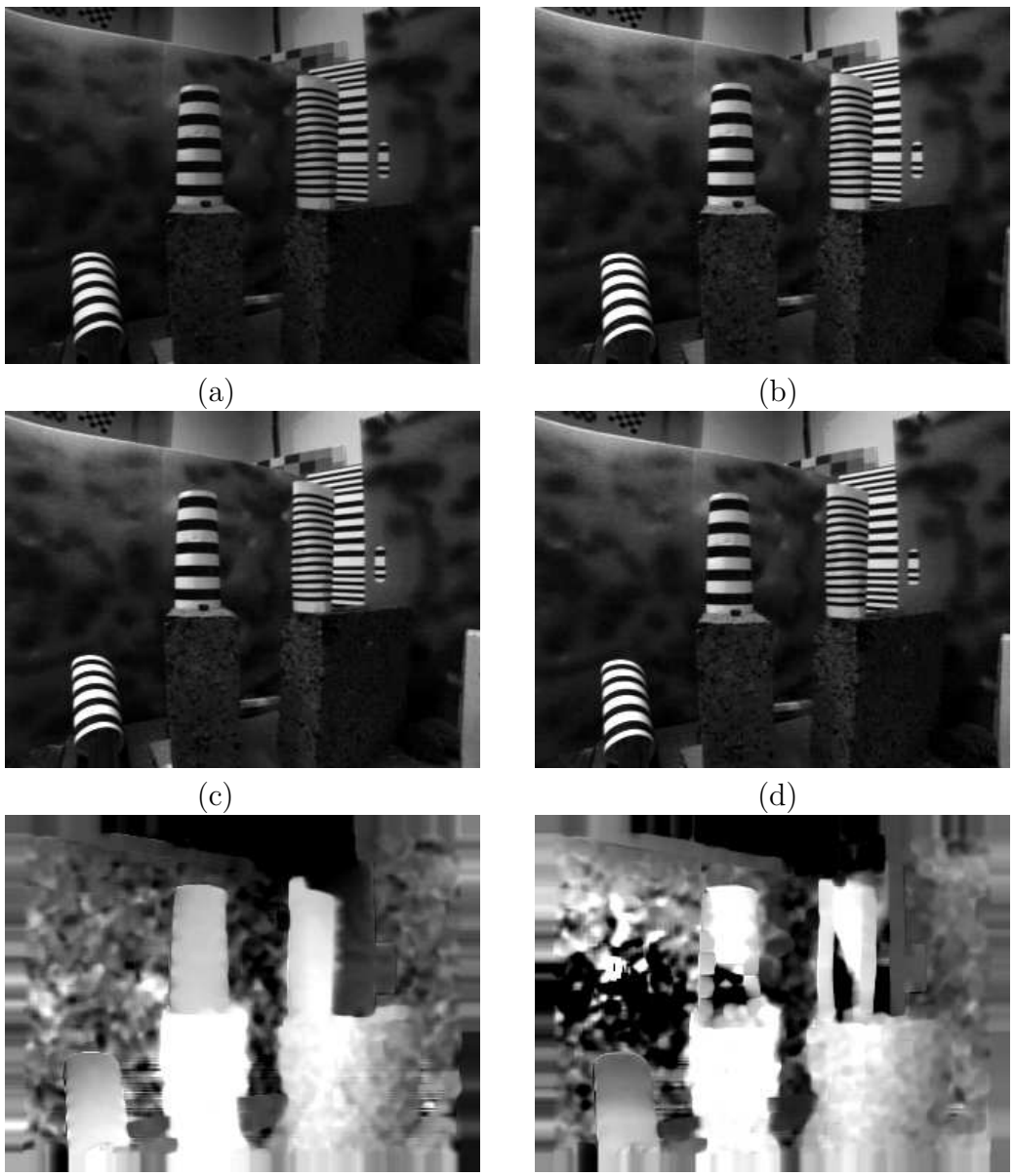
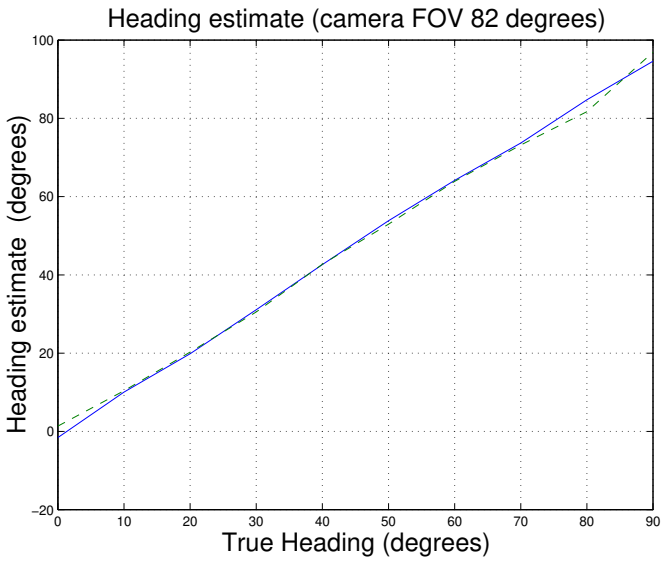
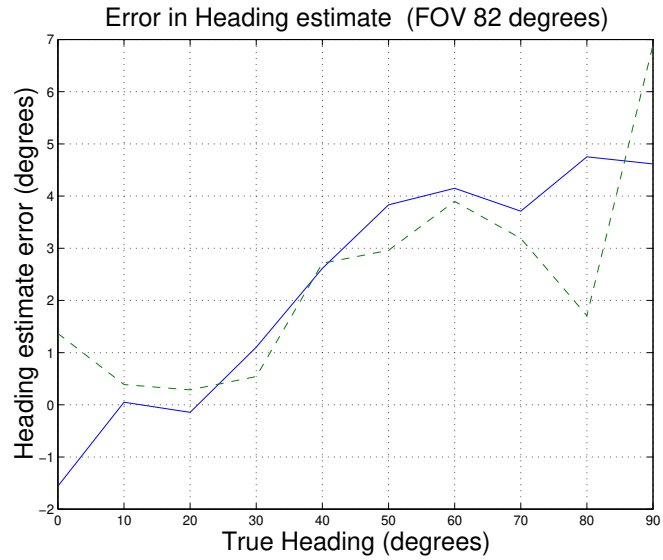


Figure 4-6: *(a) (b) (d) (c) Four input images (4.9mm lens, 82^o FOV). Motion between (a) and (d) is vertical, Motion between (a) and (b) is horizontal, forward and 30^o to the left. Motion between (a) and (c) is collinear with the motion $a \rightarrow b$ in the opposite direction. (e) Depth map recovered from images (a) (b) (d). (f) Depth map recovered from collinear motion, images (a) (b) (c). Note the depth map has errors (dark areas) around the epipole and also on the lines on the cylinders which pass through the epipole.*



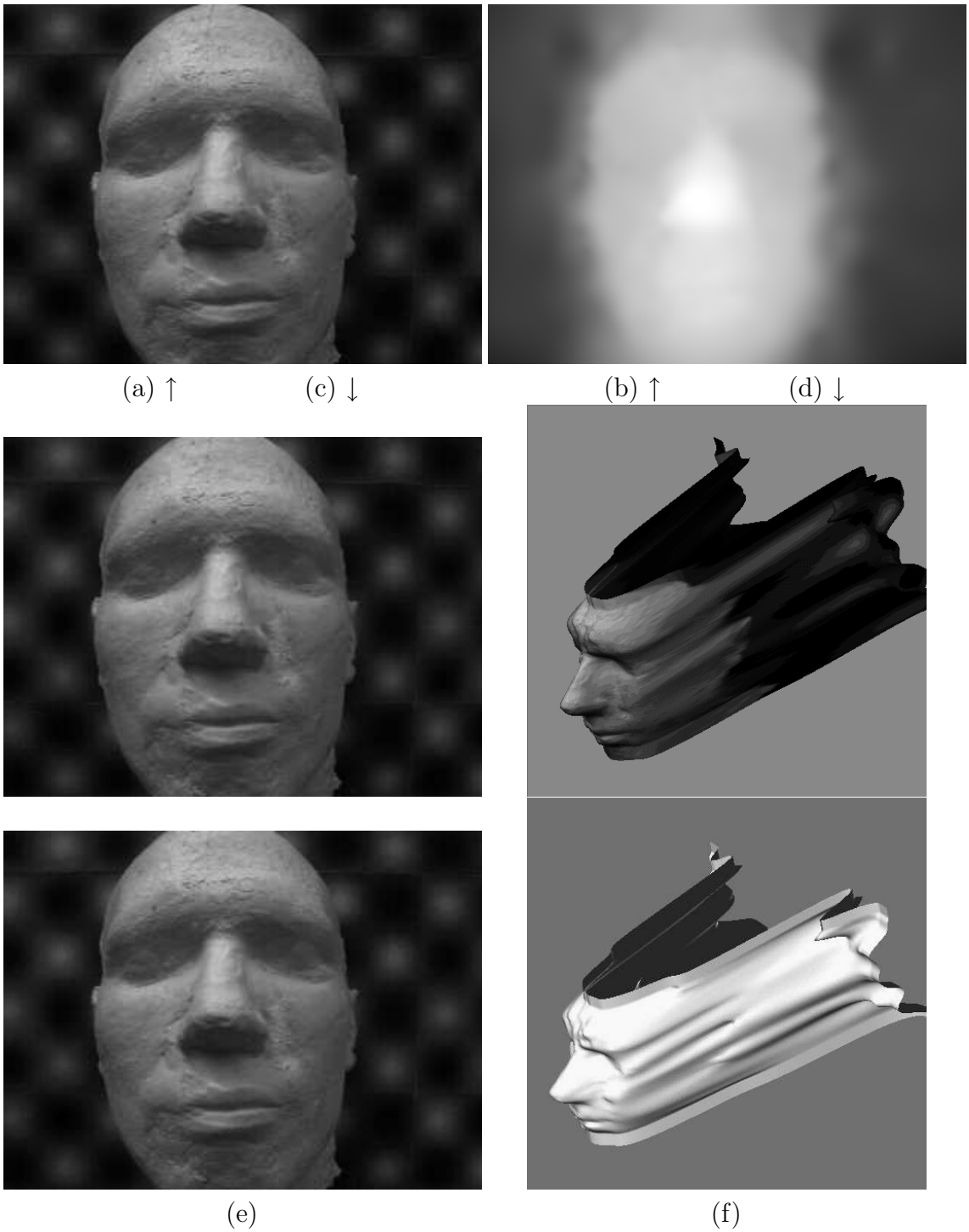
(a)



(b)

Legend:	
——	Non-collinear motion
- - -	Collinear motion

Figure 4-7: *(a) Heading estimate from a translating camera - collinear and non-collinear image triplets were used. Pure translation was assumed. (4.9_{mm} lens / 82^o FOV). (b) Difference between heading estimate and motion stage reading ("True Heading").*



(a) ↑ (c) ↓ (b) ↑ (d) ↓

(e) (f)

Figure 4-8: *The three input images (a,c,e) with collinear vertical translation. The estimated depth map (b) and 3D rendering of the surface (d,f).*

the optical flow estimates obtained in this manner with the flow estimates computed directly from two of the images using an 'industry standard' optical flow program. The program is based on code by Bergen & Hingorani of the Sarnoff Corp [7, 6]. We use two sets of images which have been chosen to be particularly difficult for optical flow programs.

Experimental Procedure

Figure 3-3 shows three images of a real scene: two paper cylinders, a plaster bust and a black metal bar in front of a background of vertical stripes. The two motions are pure translation, vertical and horizontal, parallel to the image plane. Figure 3-3a shows the recovered depth map for the sequence. The results are qualitatively correct. In Figure 3-3b the texture mapping was removed for clarity. The difficulties in the scene are explained in the introduction (Section 3.1) and expanded upon in the results section below.

From the depth map and motion the motion flow was computed using equations (equation 3.14). The flow field was also computed using the optical flow program and the results compared.

This experiment was repeated for a second set of images shown in Figure 4-12a,b,c. The main difficulty here is that most of the gradients are smooth a low contrast. The strongest edges appear at the occluding contour of the bust and these edges do not represent real features.

Results

Figure 4-9a shows the estimated flow computed from the recovered depth and motion. The camera motion was horizontal parallel to the image plane. The flow vectors have a zero Y component. Figure 4-9b shows the magnitude of the flow vectors. Figure 4-9c shows the flow estimated from two images using an optical flow program. Near the vertical bars on the left the flow vectors have a strong Y component. This is due to the aperture problem. Even a large window (aperture) will see image gradients in only one direction giving no constraint on the Y component of the flow.

On the cylinder to the lower right of the bust the magnitude of the flow vectors is too small. This is due to the fact that the edges were horizontal and parallel to the horizontal epipolar lines. This means that the depth estimates for theses points are unreliable (Figure 4-9d). This problem does not occur in Figure 4-9b because two motions one horizontal and one vertical were use to compute the original depth map.

An enlarged detail of one of the more difficult regions is shown Figure 4-10. Near the intersections (in the image) of the diagonal line and the vertical bars the flow has a small Y component. This error occurs because the program 'tracks' the intersection point as if it were a real 'feature' point, but the lines do not intersect in space. Although smaller in magnitude this is the more significant error because the program will also give a high confidence to this value.

The results for the second set of images are shown in Figure 4-11. Errors in flow direction can be seen near the edges of the bust. The most significant errors are errors

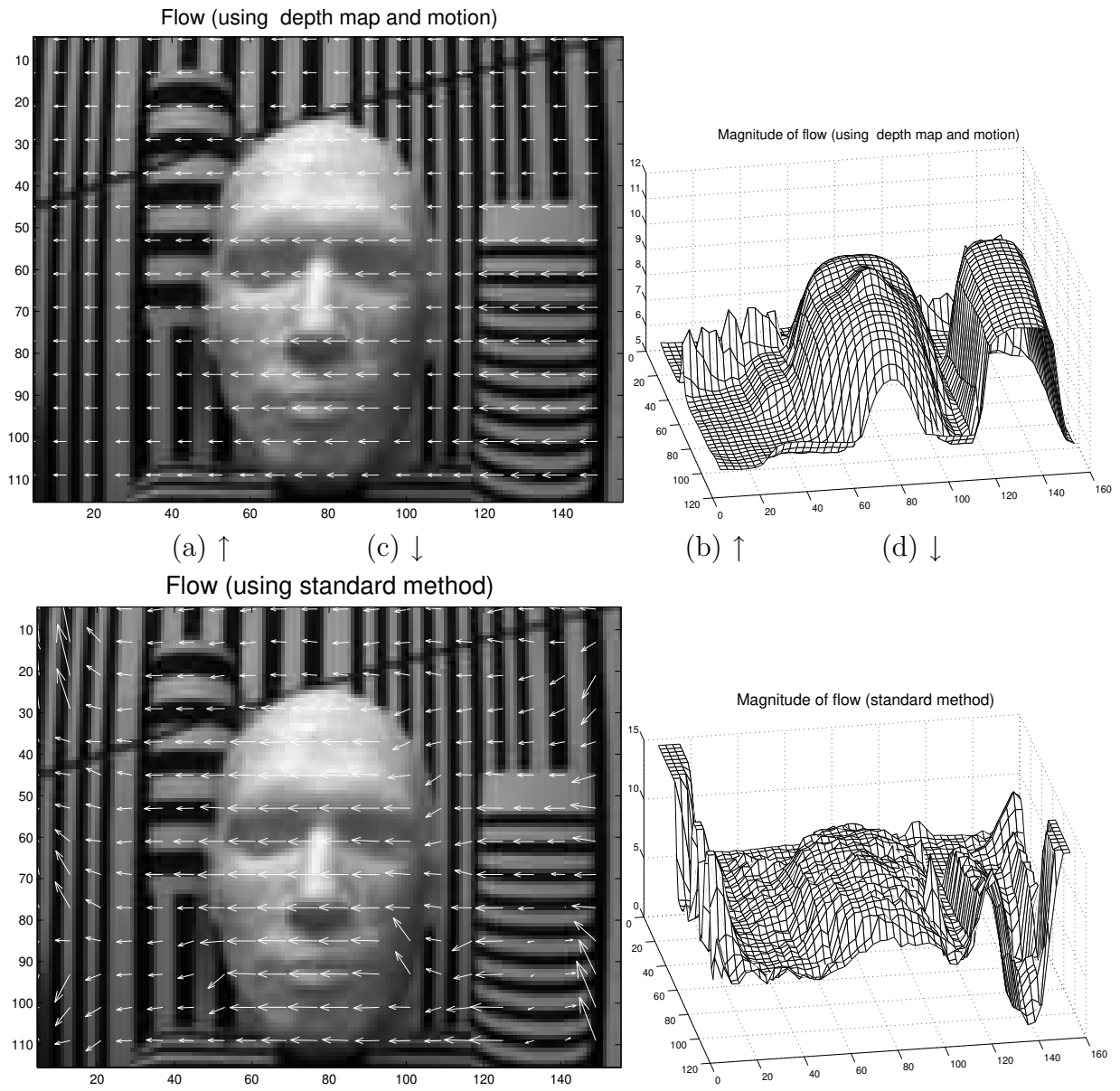


Figure 4-9: *Optical flow computed for Figure 3-1. (a) Correct optical flow computed using depth map and recovered motion. The camera motion was horizontal parallel to the image plane. The flow vectors have a zero Y component. (c) Optical flow computed using code from Bergen & Hingorani. Near the vertical bars on the left the flow vectors have a strong Y component. This is due to the aperture problem. Even a large window (aperture) will see image gradients in only one direction giving no constraint on the Y component of the flow.*

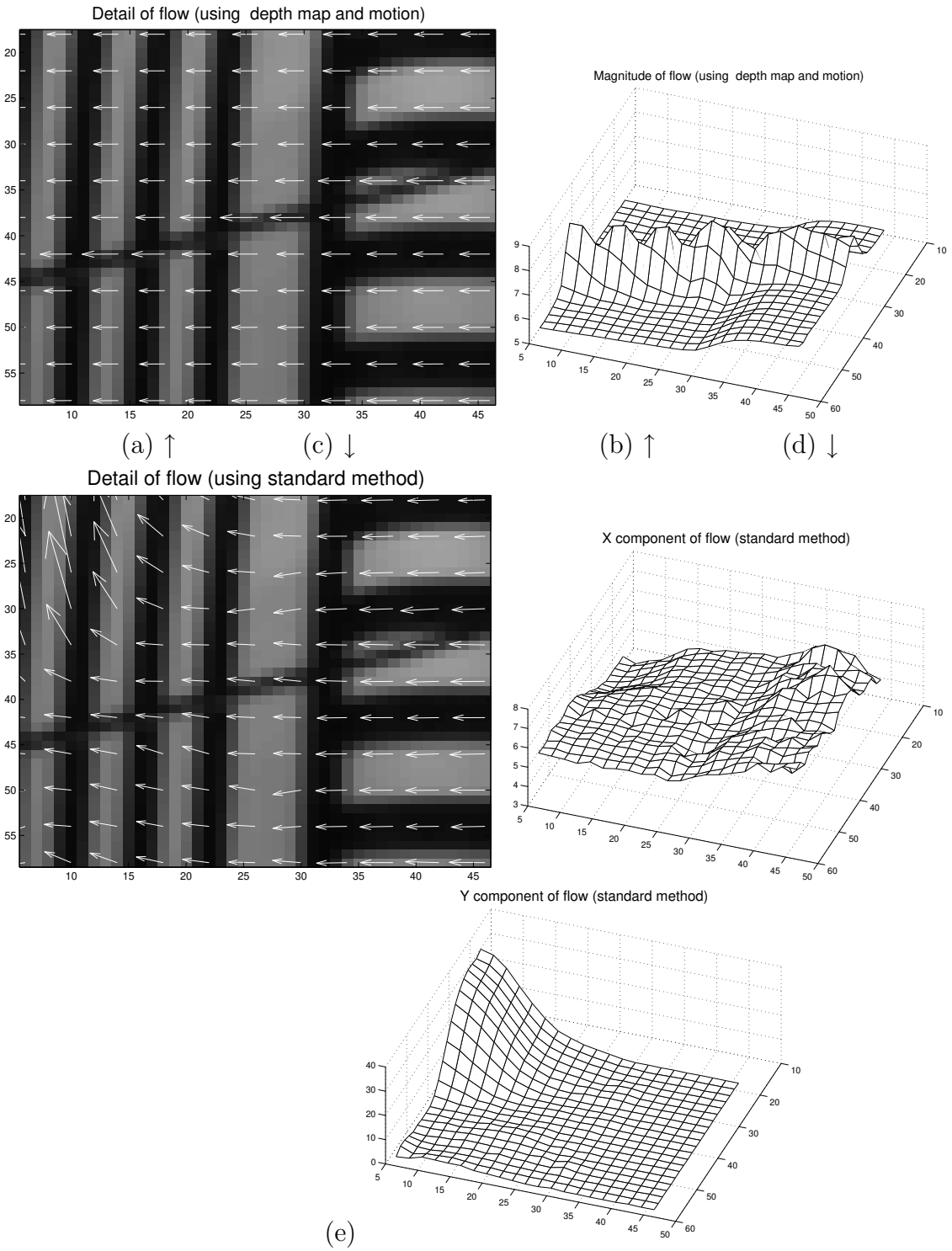


Figure 4-10: *Detail of optical flow computed for Figure 3-1. Near the intersections (in the image) of the diagonal line and the vertical bars the flow has a small Y component. This error occurs because the program 'tracks' the intersection point as if it were a real 'feature' point, but the lines do not intersect in space. Although smaller in magnitude this is the more significant error because the program will also give a high confidence to this value.*

in magnitude of the flow in the background areas. These can been seen clearly by comparing the mesh plots.

This experiment was intended to highlight the problems with optical flow techniques and how they are overcome by using direct methods. While this was not a simple 'straw man' and the optical flow program was one of the best available a full system would based on optical flow would have more components intended to overcome the problems described. Together with flow vectors the optical program would compute a confidence ellipse for each vector. These estimates would then be combined using robust methods to reject outliers during the motion computation. After the camera motion was estimated the flow would be recomputed using the epipolar constraint and the new flow would be used to compute the depth. Probably three views would be used with different camera motions. Building such a worthy competitor and fine tuning it to these image sets is of course beyond the scope of this thesis but this description does give an idea of the complexity of such a system.

4.3 Translation and Rotation

4.3.1 Experiment 4: Translation and Rotation (Reanalysis of Data from Stein & Shashua, CVPR97)

In this experiment we test the modified algorithm for recovering both translation and rotation (Section 3.7.5) on the same data set used by Stein & Shashua [74].

Experimental Procedure:

The images used by [74] were taken with an $8.5mm$ lens in the following way: The camera was translated first vertically ($10mm$) and then horizontally ($5mm$) to the right. At this third position the camera was rotated to various angles ranging from -4.0^o to 1.0^o. The depth in the scene ranged from $170mm$ to $400mm$.

Figures 4-12a,b,c show the three input images for pure translation. Figure 4-12d shows the image of the scene after the camera has been rotated -4.0^o. The flow due to a rotation of -4.0^o is much greater than the flow due to the translation even for points nearest the camera. The direct of the flow due to rotation by a negative angle was in the same direction as the image flow induced by the translation.

Results:

Figure 4-13a shows the recovered depth map using images in Figures 4-12a,b,c. Figures 4-13b,c,d show the 3D rendering of the depth map. In this case the true motion was pure translation. Another example of the reconstruction for the case when the true motion was 1.8^o was shown in Chapter 1, Figure 1-2.

Figure 4-14 shows the recovered rotation estimates for true rotations ranging from -4.0^o to 0.8^o. Outside this range the old version of the algorithm did not converge. Using the new algorithm the rotation estimates are within 5% of the correct value. There is no scaling error as was observed using the old method.

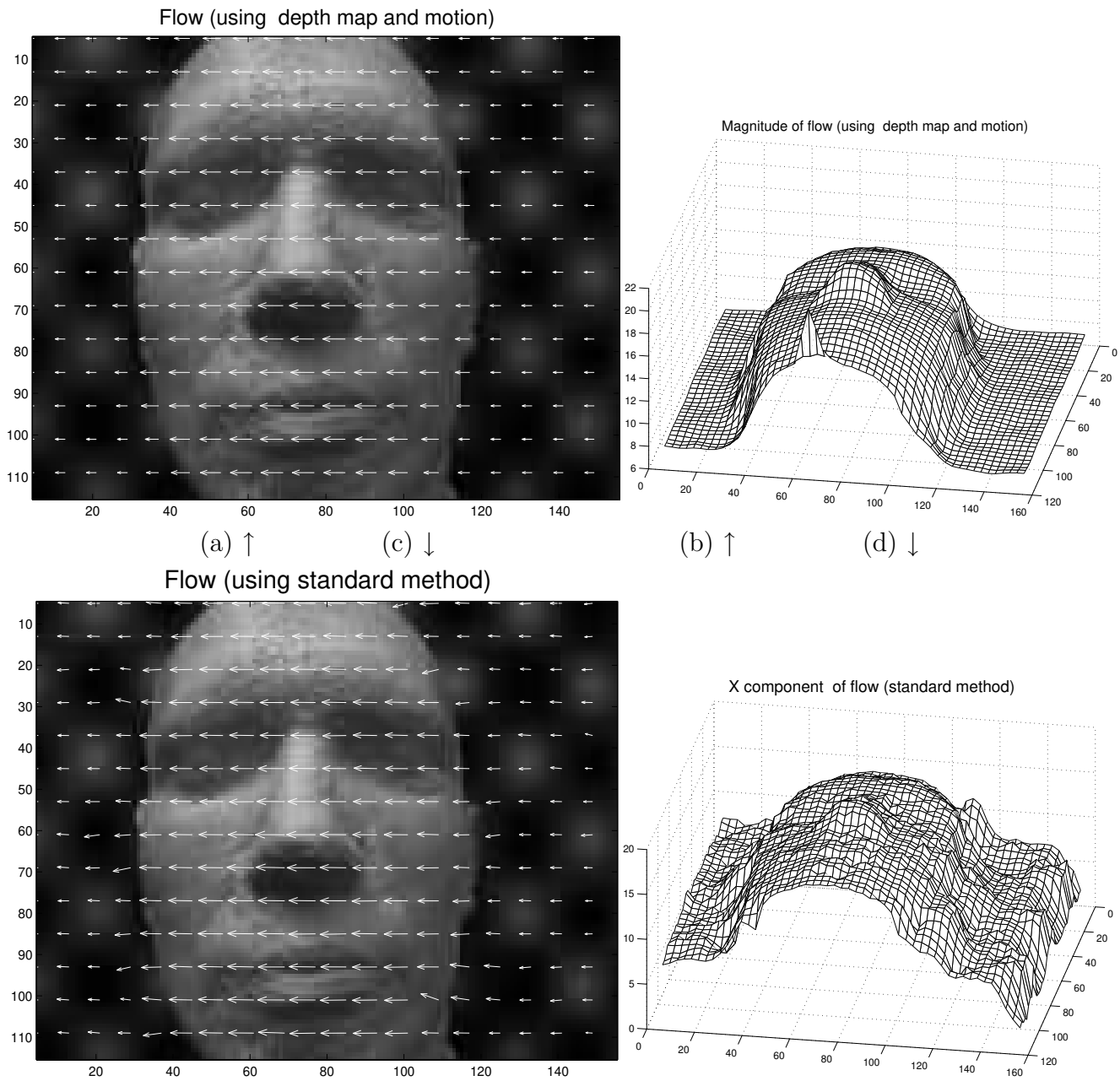
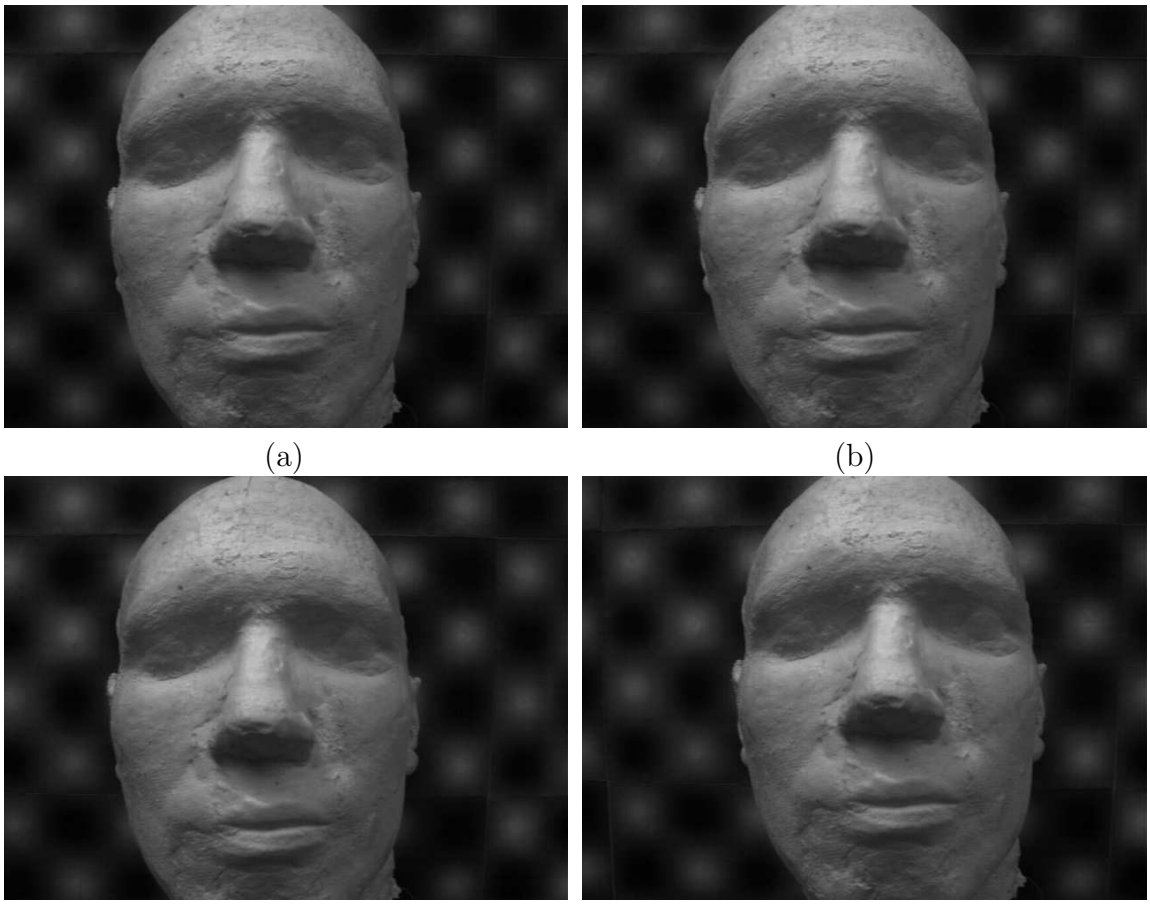


Figure 4-11: *Optical flow computed for Figure 4-12. (a) Correct optical flow computed using depth map and recovered motion. The camera motion was horizontal parallel to the image plane. The flow vectors have a zero Y component. (c) Optical flow computed using code from Bergen & Hingorani. Errors in flow direction can be seen near the edges of the bust. The most significant errors are errors in magnitude of the flow in the background areas. These can been seen clearly by comparing the mesh plots.*



(a) (b)

(c) (d)

Figure 4-12: *The three input images (a,b,c) with pure translation. (d) another image taken at the same location as (b) with* -4.0^o *rotation. Note that the image flow due to translation is much smaller than the flow between (b) and (d) which is due to the camera rotation.*

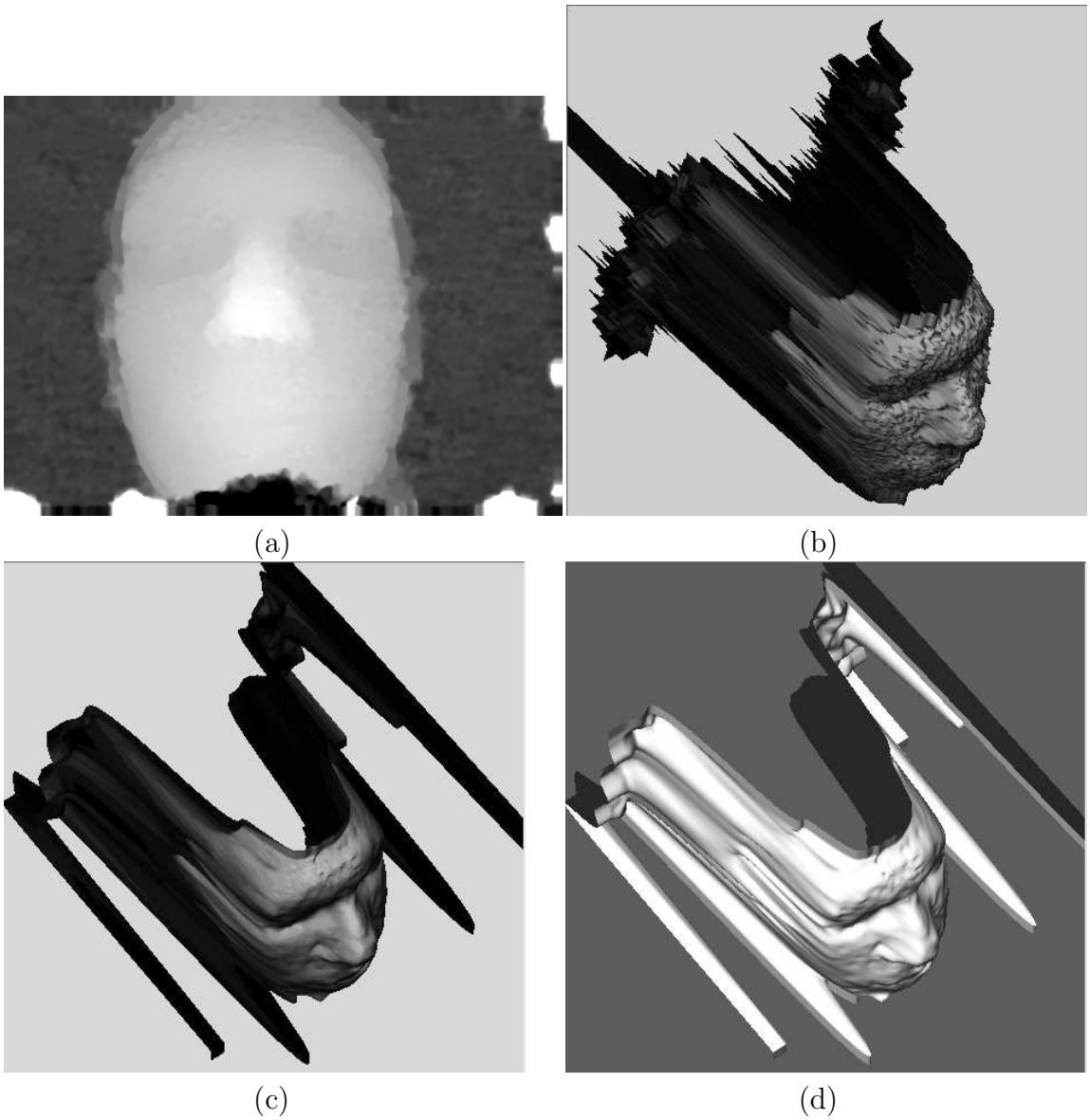


Figure 4-13: *The estimated depth map (a) and 3D rendering of the surface (b). There are errors in the depth estimates around the edges. Depth estimates were computed using* 9×9 *region and a local constant depth model. In (c) and (d) a* 3×3 *region was used to compute the depth together with a weak smoothness constraint.*

The new algorithm also has a wider range of convergence. The original images were not available for larger rotations but a similar setup yielded good results for rotation for the range of -5.0^o to 5.0^o. Since rotations can create large amounts of image flow, key to the successful convergence are strong signals with low spatial frequency such as the white head on a darker background. This leads to convergence at the coarse level which is then used as the starting point for motion estimates at finer levels. Strong repetitive high frequency patterns such as checker board patterns are more difficult. The convergence range for the scene in Figure 4-6 is only $\pm 0.5^o$.

4.3.2 Experiment 5: Motion Heading Estimation with Rotation

The aim of this experiment is to test the motion estimation algorithm over a wide variety of motion directions when the motion included some rotation. For example, this experiment tests whether the algorithm works when the direction of motion is inside and outside the FOV. We also find the error created by neglecting even small amounts of rotation. The two translations are non-collinear.

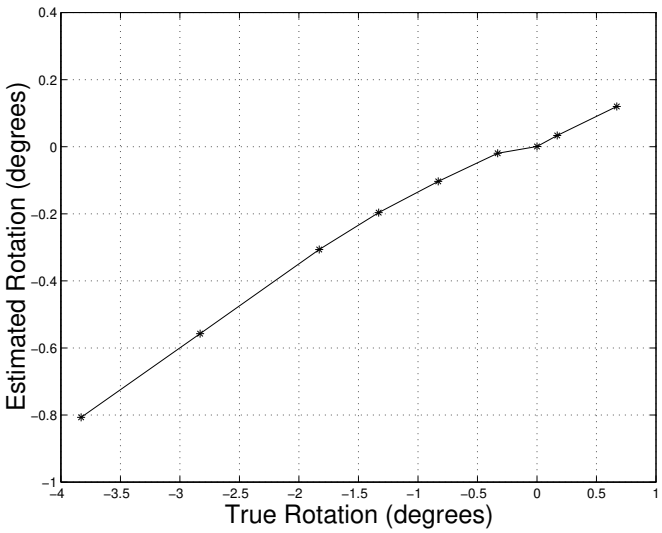
Experimental Procedure:

The camera setup is similar to that of experiment 1 (Section 4.2.1). The camera is rotated to a particular heading direction $(0^o, 10^o, ..., 90^o)$. The camera was translated vertically and then horizontally. After the horizontal translation the camera was rotated $\pm 0.5^o$ and $\pm 1.0^o$. Therefore, for each image heading, we have an image triplet which is translation only and then triplets where the second motion also includes some small rotations. The rotation axis of the stage passed within $5mm$ of the camera center of projection thus rotations of 1.0^o produce translations of less that $1mm$. Figure 4-15 shows input images where the horizontal translation heading was 60^o to the camera optical axis.

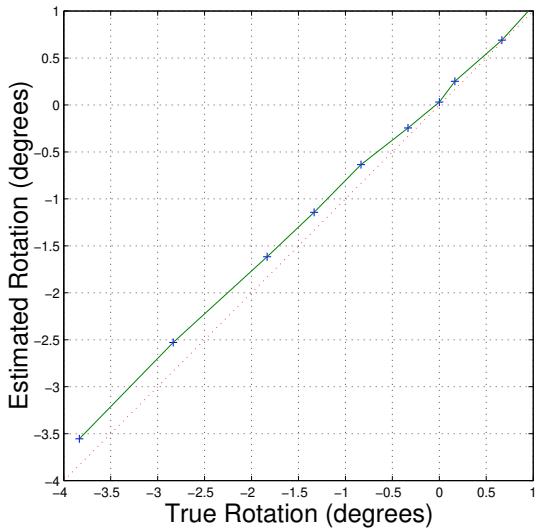
Results:

Depth and motion estimates were computed either assuming pure translation or allowing possible rotation. The results for various motion and motion-assumption combinations are shown in Figure 4-16, Figure 4-17 and Figure 4-18. We notice that for even a small rotation (0.5^o) the heading estimate is off by over 10^o if we do not take the rotation into account. When we estimate both rotation and translation the motion estimates are good (RMS error 2.1^o) although not as good as those obtained from a purely translating camera under the pure translation assumption (RMS error 1.44^o).

Figure 4-19a shows the depth map estimated from the images in Figure 4-15a), Figure 4-15c and Figure 4-15d allowing for both translation and rotation. The depth map is qualitatively correct even though the second camera motion included -1.0^o rotation. Figure 4-19b shows the depth map estimated assuming pure translation when in fact the second camera motion included 0.5^o rotation.

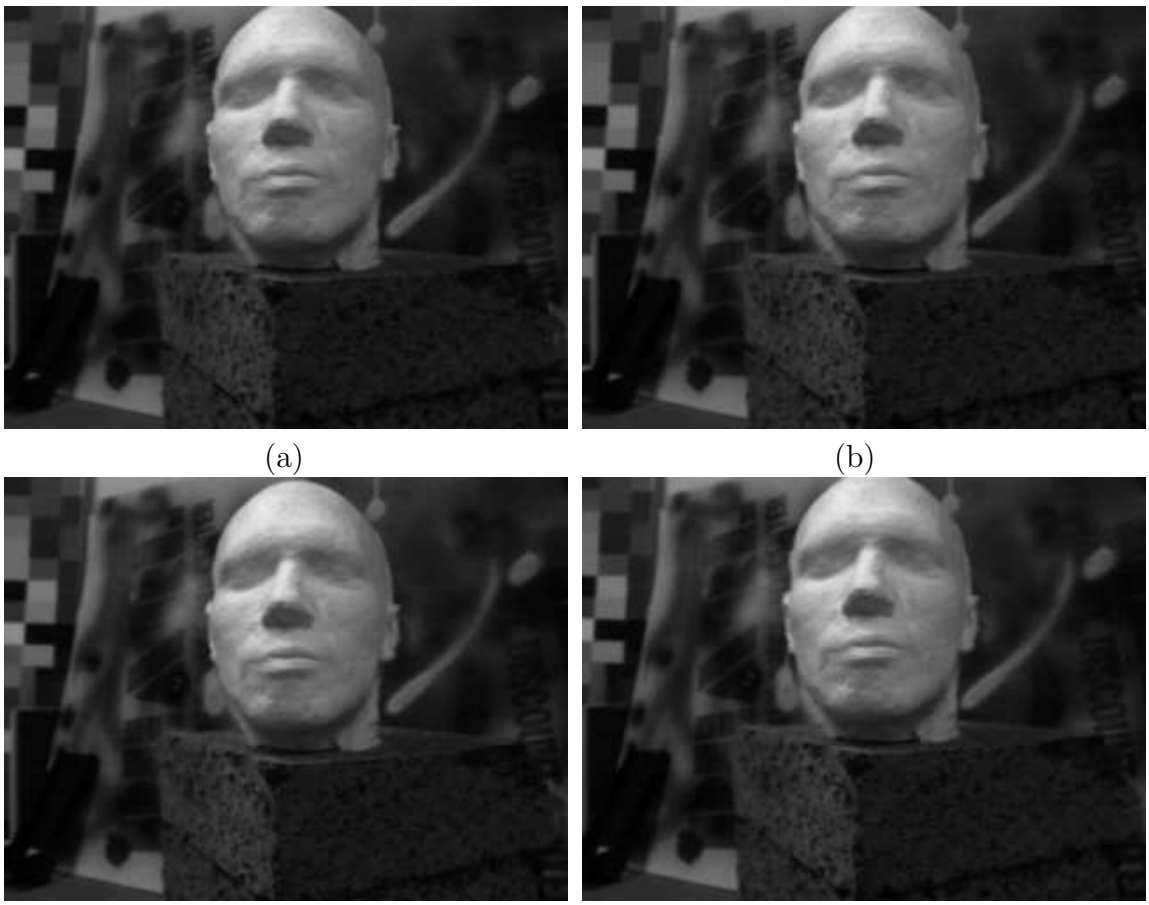


(a)



(b)

Figure 4-14: *Estimated rotation as a function of true rotation. (a) Old results: Although there is the correct linear relationship there also appears to be a significant scale error (note Y axis). (b) New results - the rotation estimates are correct.*



(a) (b)

(c) (d)

Figure 4-15: *Four of the input images used in the experiment to test motion heading estimation in the presence of rotation. Motion (a) to (c) is vertical motion. Motion (a) to (b) is horizontal motion forward and to the left (*60^o *from the camera optical axis). Motion (a) to (d) has the same translation as (a) to (b) but also with a rotation of* 1.0^o*. In this case the flow due to rotation of* 1.0 *cancels out the flow due to translation in the area of the face and more than cancels out the flow in regions of greater depth. (Lens* 8.5*mm lens /* 52^o *FOV)*

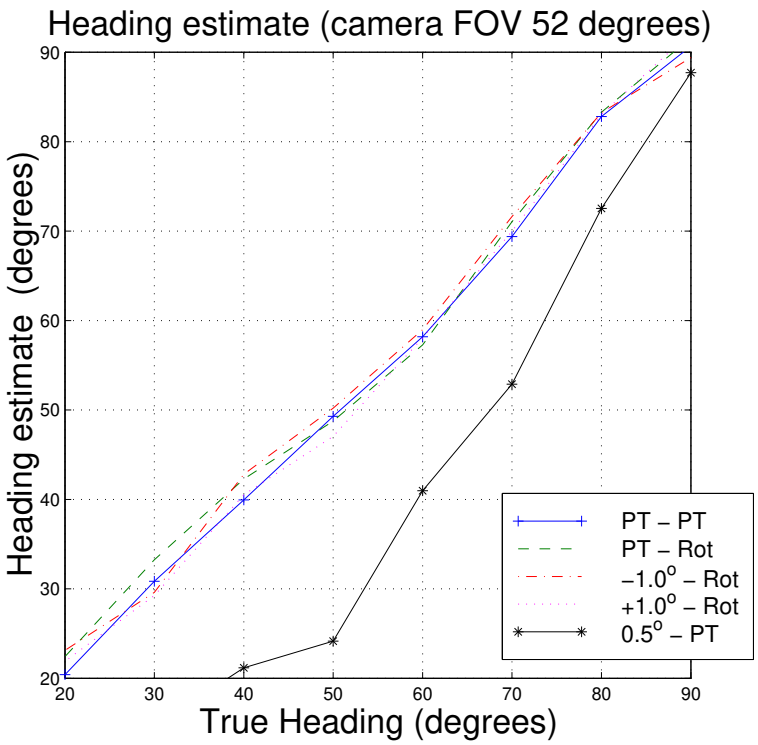


Figure 4-16: *Heading estimates from a moving camera. (PT - PT) Both true motion motion model was pure translation. (*0.5^o *- PT) True motion included* 0.5^o *and motion model was pure translation. Note that even a small rotation, such as* 0.5^o*, is enough to create large heading estimation errors if rotation is not taken into account. (PT-Rot,* $+1.0^o$ *- Rot,* -1.0^o*- Rot) Motion model included rotation Actual rotations were* $0^o, 1.0^o, -1.0^o$*. (*8.5_{mm} *lens /* 52^o *FOV)*

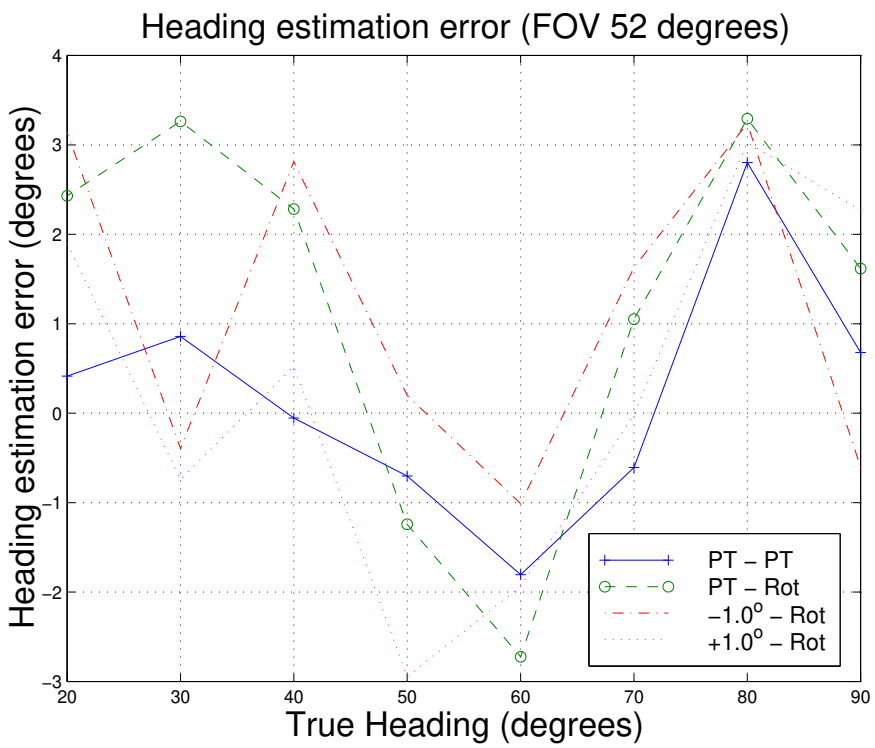


Figure 4-17: *Heading estimate error: the difference between heading estimate and motion stage reading. Even in the presence of camera rotation (*$+1.0^o$ *- Rot,* -1.0^o*- Rot) the heading estimation errors are small (RMS error* 2.1^o*) although not as good as obtained from purely translating camera under the pure translation assumption (PT-PT) where the RMS error was* 1.44^o*. (8.5mm lens /* 52^o *FOV)*

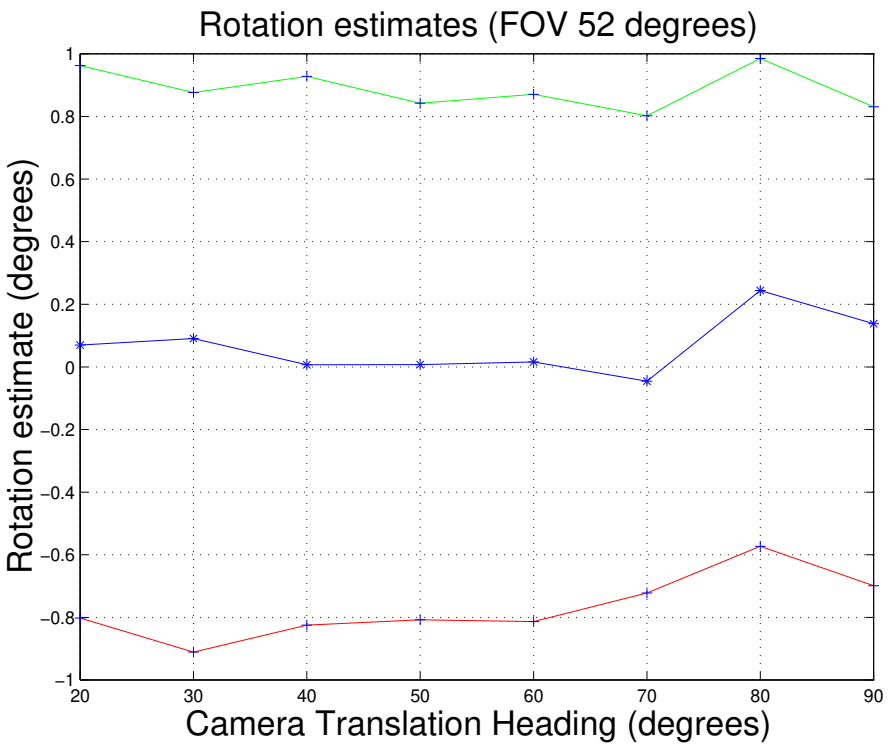
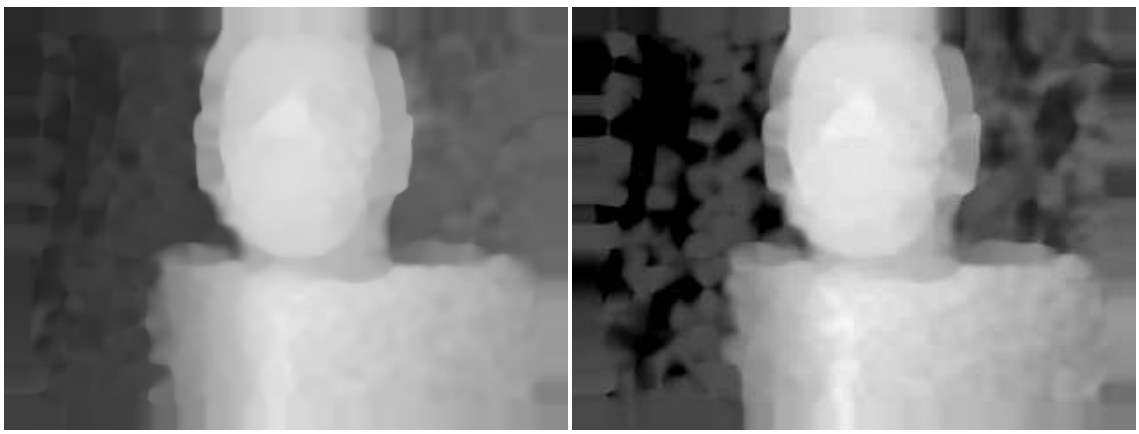


Figure 4-18: *Rotation estimates obtained from a camera translating in different translation directions with some added rotation. Actual rotations were* $0^o, 1.0^o, -1.0^o$. *(camera with* 8.5_{mm} *lens /* 52^o *FOV)*



(a) (b)

Figure 4-19: *(a) Depth map estimated after computing both translation and rotation. The second camera motion included* -1.0^o *rotation. (b) Depth map estimated assuming pure translation when the second camera motion included* 0.5^o *rotation. Notice the depth errors in the background.*

4.3.3 Experiment 6: Euclidean Structure Estimation in the Presence of Rotation.

In this section we evaluate the Euclidean structure estimation when the motion includes rotation. We use a simple scene, a cube, in front of a planar background. We perform the 3D reconstruction of the scene. We then measure whether the right-angles between the cube faces are correctly recovered.

Experimental Procedure

Figure 4-20 shows the simple scene of a textured cube in front of flat, low-contrast, background. The camera ($8.5mm$ lens / 52^o FOV) was translated vertically and then horizontally parallel to the image plane. The camera was then rotated to $+2.0^o$ and -1.0^o from its original heading.

Since we were interested in the 3D structure and not jut motion estimates, after estimating both structure and motion we went back down the pyramid and recomputed the structure while keeping the motion estimates constant (Section 3.7.6). The depth estimates before and after the second pass are shown in Figure 4-20d and Figure 4-20f respectively. In Figure 4-20d there is a "hole" on the right of the depth map. This is due to the appearance of a white patch near the border of the image. As with occlusions this sudden appearance violates the constant brightness assumption. The situation is fixed by the second pass. In general points near the edge of the depth map are unreliable.

Results:

The rotation estimates were 1.68^o and -0.67^o for true rotations of $+2.0^o$ and -1.0^o respectively. The 3D rendering of the cube is shown in Figures 4-21 and 4-22. The wire-frame rendering of the overhead views (Figures 4-21b and 4-22b) show the recovered angle between the cube faces. The estimated angles were 95^o and 86^o respectively. Figure 4-21d shows a side view of the cube. The estimated angle between the top and side face of the cube is 90^o.

4.4 Experiments with a Hand-held Camera

4.4.1 Experiment 7: Euclidean Structure Estimation Using a Hand-held Camera

In previous experiments the camera was mounted on a motion stage so that ground truth motion values were available. In this experiment the camera was hand-held to show that the system can deal with the more natural situation. Simple objects were used to create the scene so that it would be easier to evaluate the shape reconstruction.

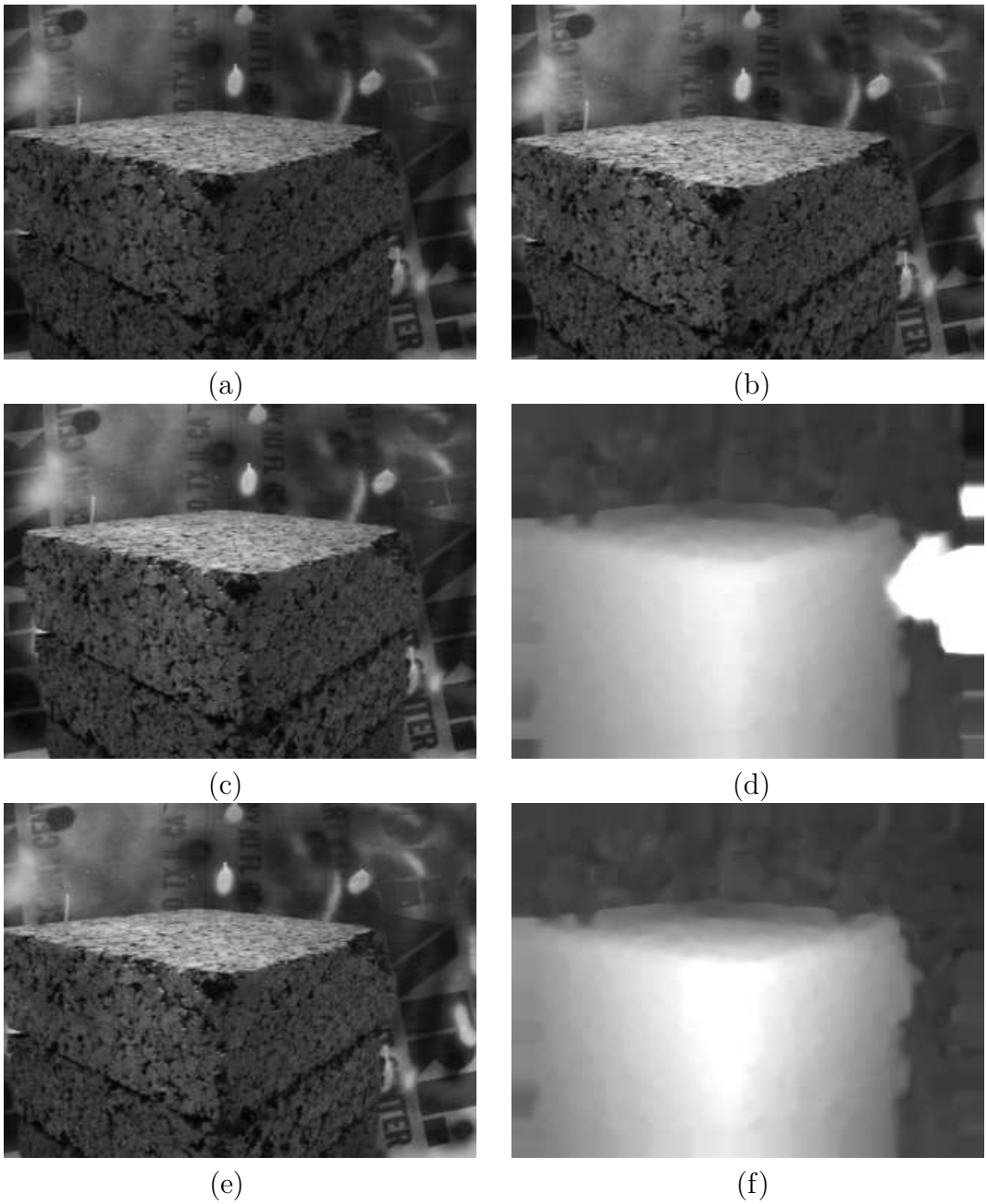
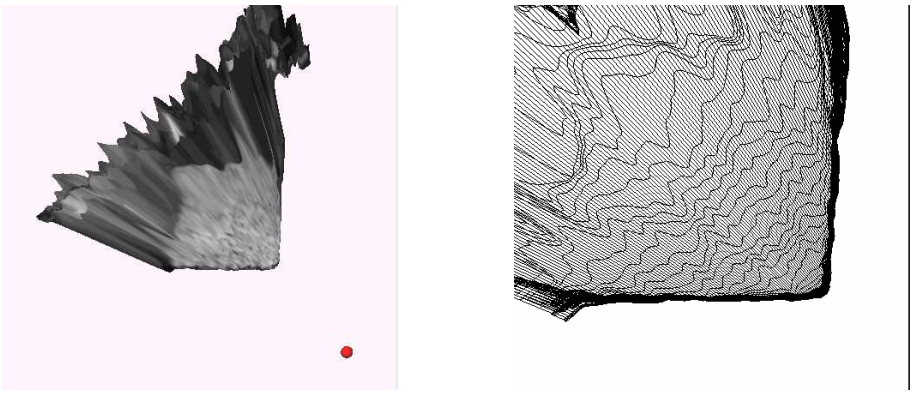
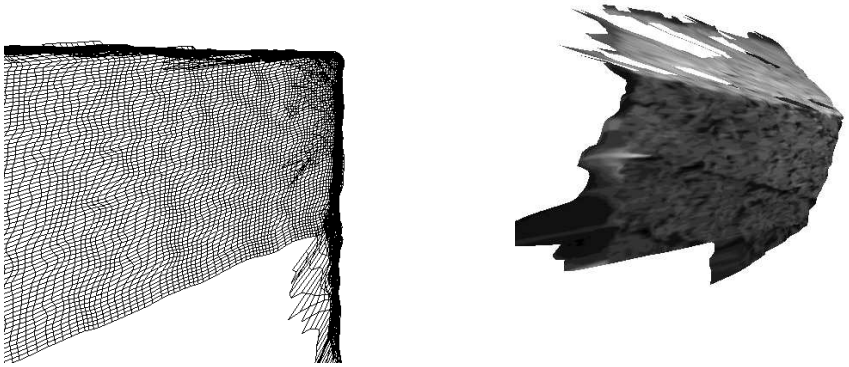


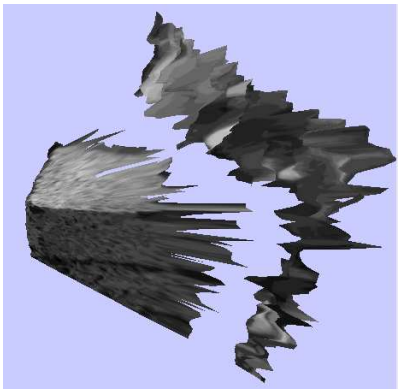
Figure 4-20: *A simple scene used to test 3D Euclidean reconstruction: a cube constructed from cork blocks in front of a flat background. Motion (a) to (b) was vertical. Motion (a) to (c) was sideways translation with* -1.0^o *rotation. Motion (a) to (e) was same sideways translation but with* $+2.0^o$ *rotation. (d) initial depth map from images (a), (b) and (c). Note how the disappearance of a white blob in the background near the right edge of the image creates an error in one region of the depth map. (f) After second pass of depth estimation the "hole" is fixed. (see text)*



(a) (b)

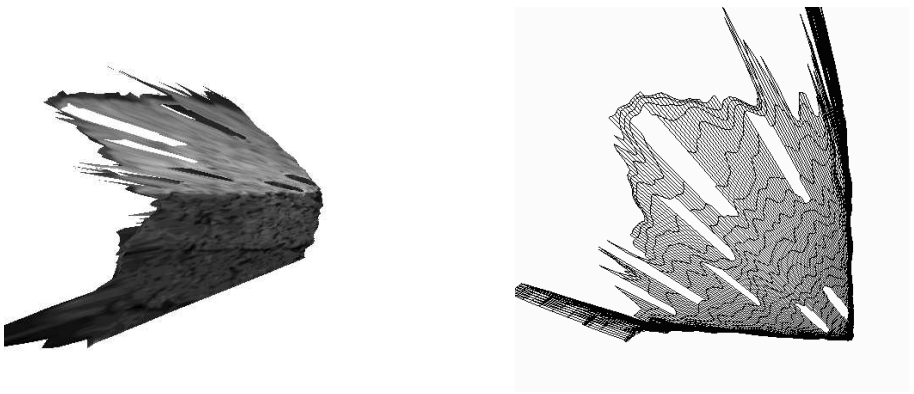


(c) (d)



(e)

Figure 4-21: *3D rendering of the recovered scene. The camera motion included* -1.0^o *rotation. (a) Overhead view. The estimated camera position is marked by the sphere in bottom right-hand corner. (b) An enlarged view of the corner of the cube. The angle between the faces of the cube is estimated* 95^o *instead of* 90^o*. (c) Side view of the segmented cube from up and to the side. (d) enlarged wire-frame side view of the cube. The estimated angle between the top face of the cube and the right face is* 90^o*. Another view of the scene showing the foreground/background segmentation. The depth estimates of the back panel are noisy because they are further away from the camera and because the texture does not have strong gradients.*



(a) (b)

Figure 4-22: *3D rendering of the recovered cube. The camera motion included* $+2.0^o$ *rotation. The cube has been segmented from the background using depth based foreground/background segmentation. (a) View from up and to the side. (b) Wire-frame rendering of the overhead view. From the overhead view the estimated angle between the cube faces is* 86^o *instead of* 90^o

Experimental Procedure

A progressive scan camera (Pulnix TMC9701) with a wide angle lens ($4.9mm$ lens / 82^o FOV) was hooked up directly to the frame grabber of an SGI Indy workstation. The cameras help in the hand with no tripod. An image sequence of 30 frames was captured while the camera was moved up and down and side to side the combination of which produced a circular motion. An effort was made to keep the camera rotation down to a minimum.

From the image sequence three images were selected which seemed to give two distinct motion directions.

Results

Figure 4-23a,b,c show three input images of a cube. The motion and depth was estimated for three different cases. Figure 4-23d shows the estimated depth map allowing for both rotation and translation and taking into account the radial distortion. 3D renderings are shown in Figure 4-24. The rotation estimates for the two motions are shown in Table 4.1. Figure 4-23f shows the estimated depth map allowing for both rotation and translation but neglecting radial distortion. The resulting depth map is qualitatively the same but the measured angles between the cubes faces are larger and further from the true value of 90^o (Figure 4-25). If we assume pure translation the resulting depth map (Figure 4-23e) appears to be qualitatively the same but the 3D renderings (Figure 4-26) clearly show that the results are more noisy and further from true Euclidean reconstruction. Figure 4-27 and Figure 4-28 show results from a more complex scene.

Errors in depth reconstruction when we neglect the rotation are to be expected according to [14]. They show that for small total rotation angles ($|w|$) the error in

Table 4.1: *Rotation estimates (in degrees) from the images in Figure 4-23. The camera was hand-held so ground truth values of the motion are not known.*

	w_x	w_y	w_z
Motion 1	-0.7809	0.2262	-0.5520
Motion 2	-0.0538	0.2684	1.4246

depth due to an error in rotation ($|\delta w|$) is given by:

$$\frac{\delta Z}{Z} \approx \left(\frac{f}{|u|}\right)(|\delta w|)$$

where f is the focal length, $|u|$ is the magnitude of the image motion. In our case by assuming pure translation we get a rotation error of 1.45^o or $0.025rad$ (Table 4.1). The average image motions were on the order of $15pixels$ and the focal length $f \approx 180$. So we expect:

$$\frac{\delta Z}{Z} \approx 0.3.$$

[14] refers to depth from image motion. The errors we observe are not so large but are definitely noticeable. We also note that even if we neglect the rotation none of the depth estimates were negative so a *depth is positive* constraint, as suggested by [19], would not help in this example.

4.4.2 Experiment 8: Outdoor Scenes - Church Wall

In this experiment we tested the algorithm on outdoor scenes. These experiments are important because they show that the photometric constraints can be used in uncontrolled lighting conditions. Figures 4-29a,b,c show three images of a small part of a church wall. The camera was Sont Hi8 camcorder with the lens open as wide as possible (FOV 40^o). The camera was mounted on a tripod. Moving the camera horizontally ($a \rightarrow b$) was quite smooth but moving the camera up and down included about 0.5^o rotation. Figure 4-29d shows the recovered depth map which can be seen to be qualitatively correct. The Euclidean 3D renderings of the depth map are shown in Figure 4-30.

4.4.3 Experiment 9: Outdoor Scene 2 - Side Entrance

Figure 4-31 shows three views of the side entrance to a building. The camera was a progressive scan camera (Pulnix TMC9701) with a wide angles lens (4.9mm / FOV 82^o). The camera output was recoded on a Sony Hi8 camcorder. The camera was mounted on a tripod and an effort was made to reduce the amount of rotation, but rotation was 0.32^o. Figure 4-31d shows the recovered depth map. The 3d Euclidean rendering of the depth map is shown in Figure 4-32.

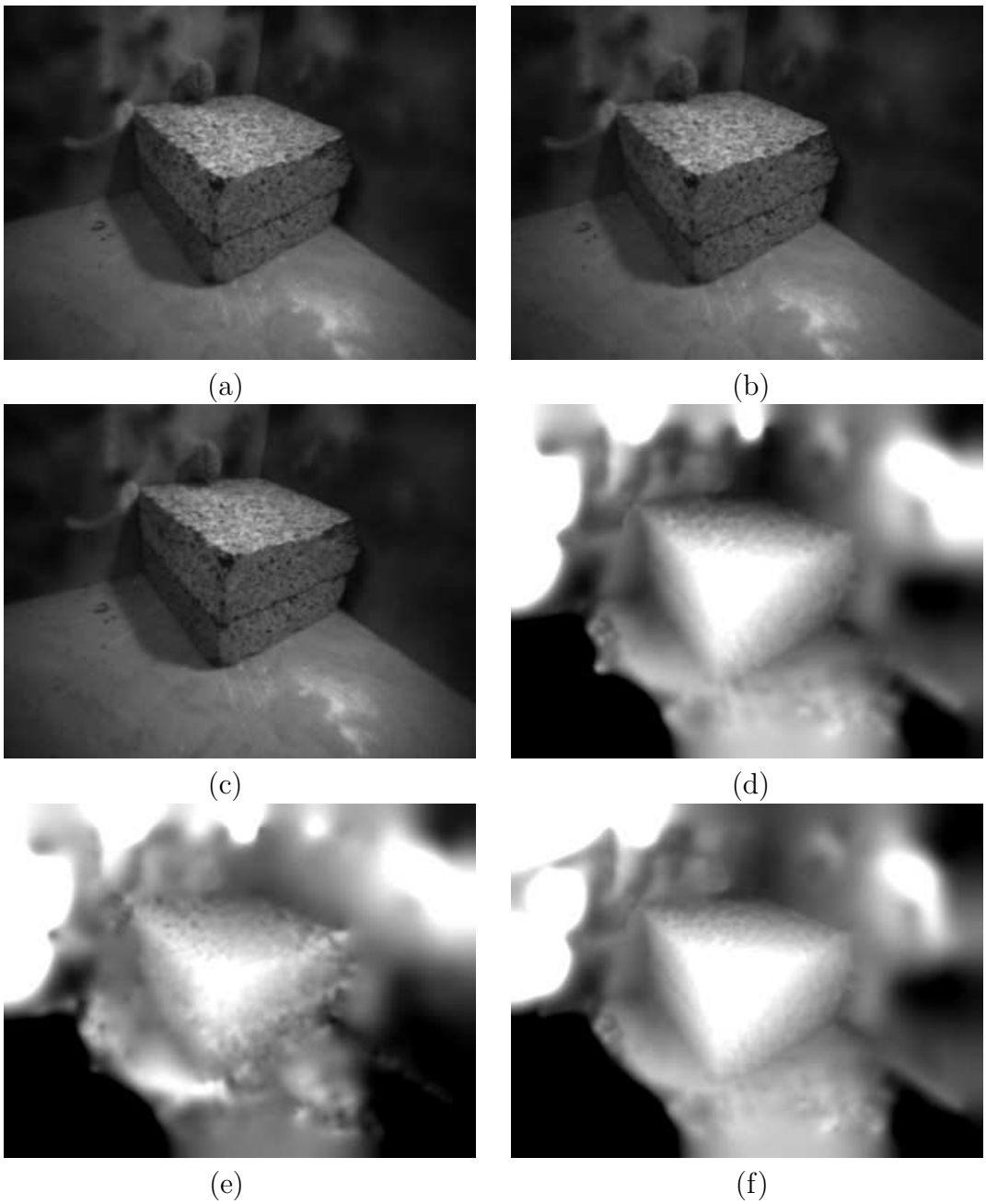


Figure 4-23: *A simple scene used to test 3D Euclidean reconstruction. Camera was hand-held so exact motion between images (a), (b) and (c) is not known. (d) Depth reconstruction allowing for both rotation and translation with radial distortion correction. (e) Assumes pure translation. (f) Allows for both rotation and translation but no radial distortion correction. The cube and it's surrounding are recovered well while textureless areas near the boundaries have large errors. The three depth maps look qualitatively the same but 3D renderings in following pages clearly show the differences.*

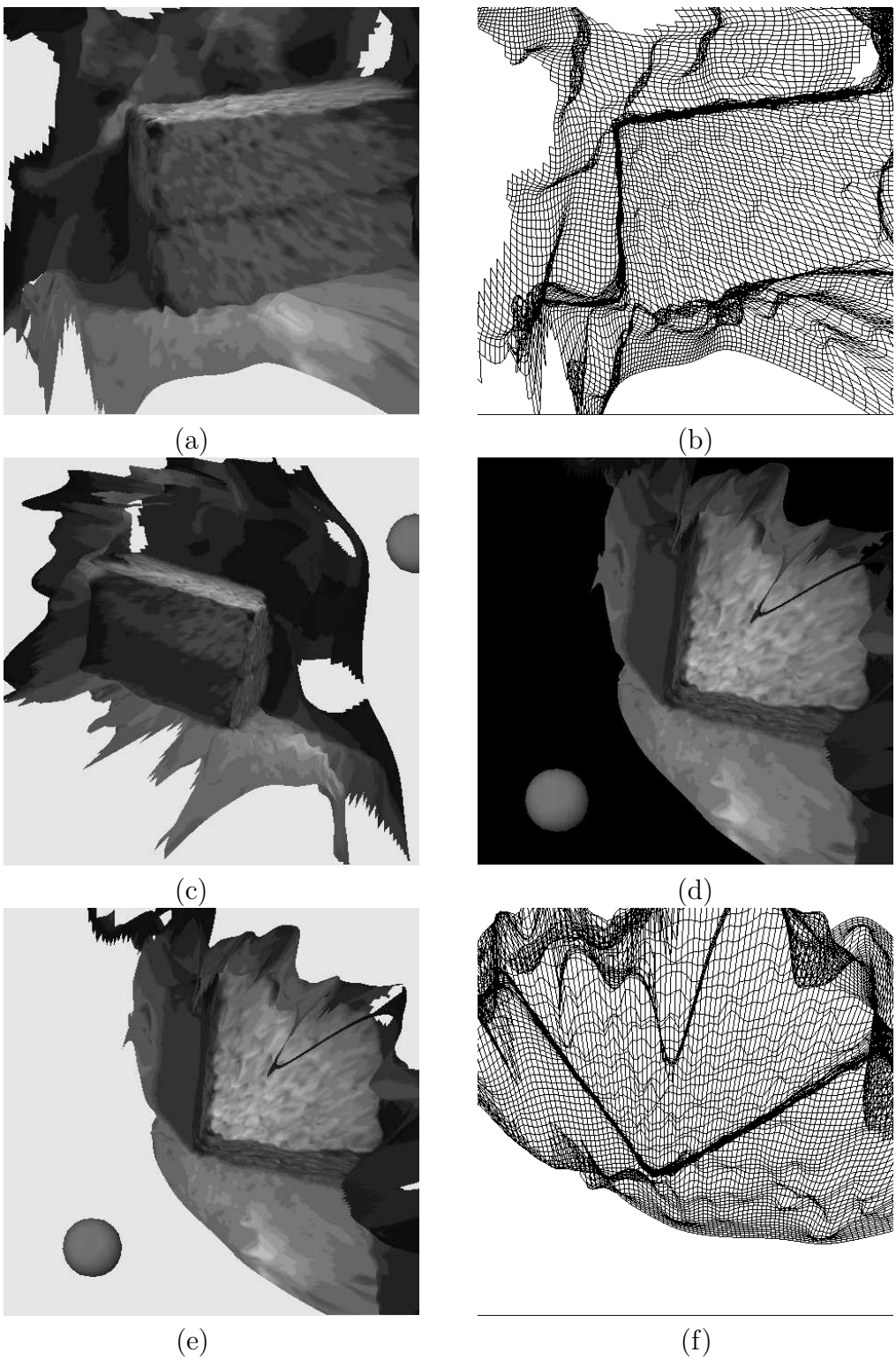


Figure 4-24: *3D renderings of the depth map in Figure 4-23d. The motion estimation allowed for rotation and translation and the input images were preprocessed to correct for lens distortion. (a,b,c) side views. (d,e,f) top views. The estimated angles between cube faces are (b)* 96^o *and (f)* 97^o*.*

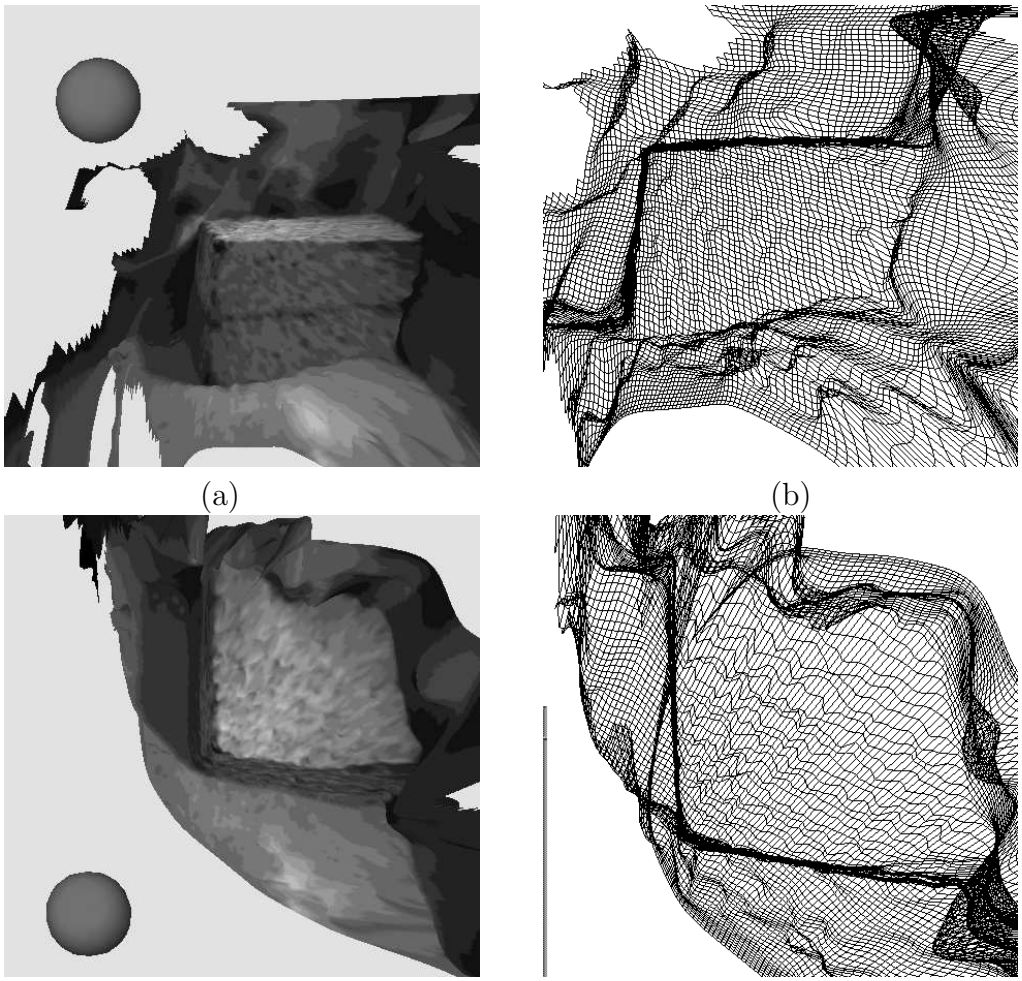
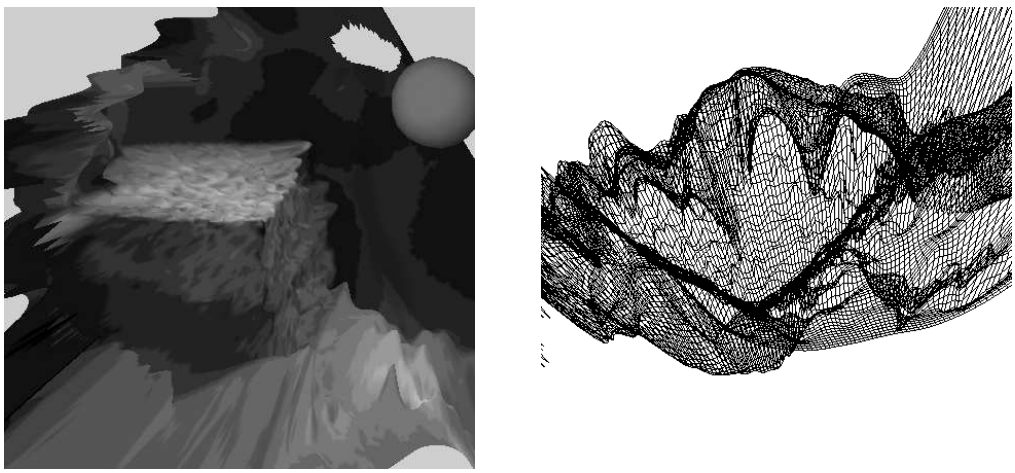


Figure 4-25: *3D renderings of the depth map in Figure 4-23f. The motion estimation allowed for rotation and translation. There was no correction for lens distortion. The estimated angles between cube faces are (b)* 97^o *and (d)* 101^o*.*



(a) (b)

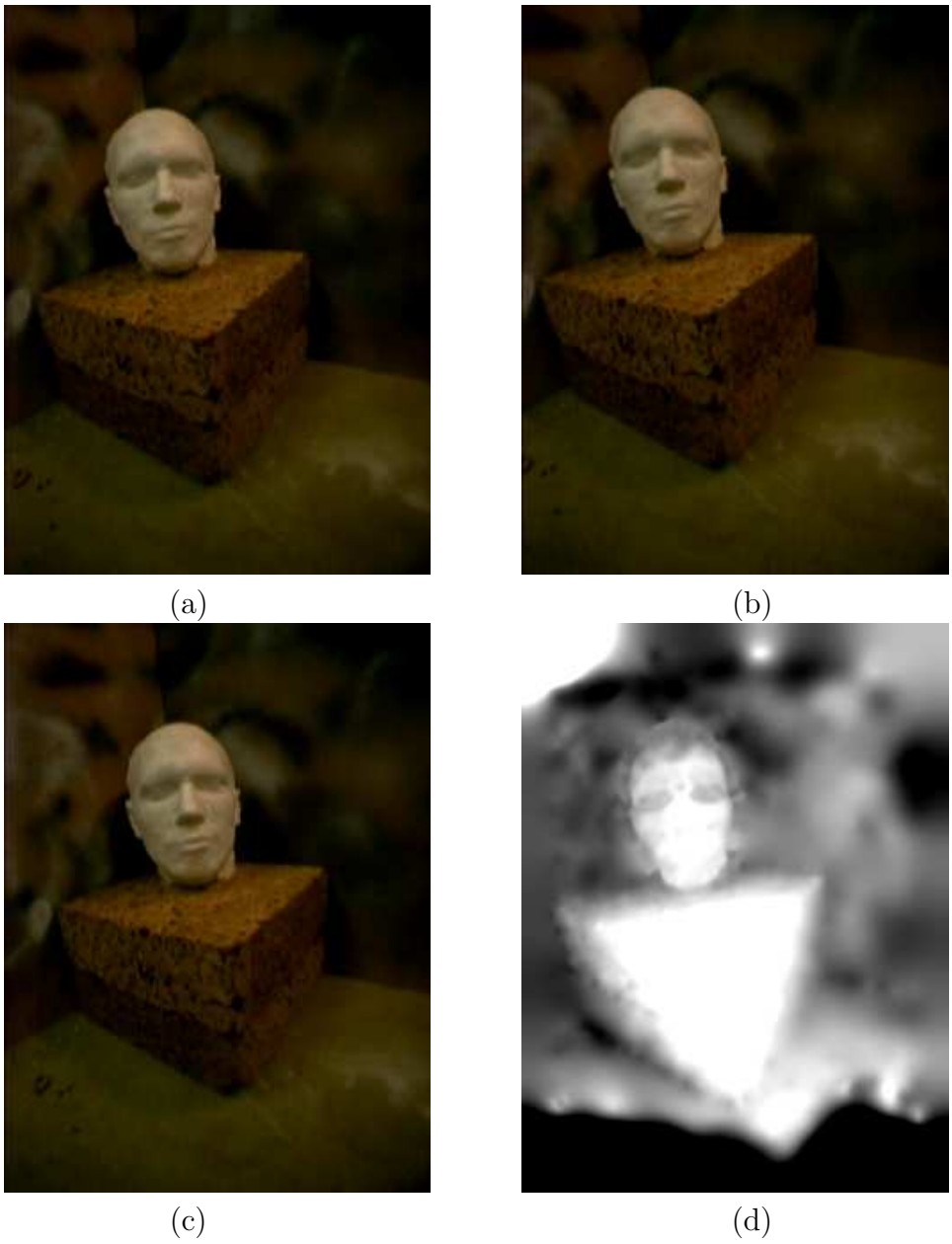
Figure 4-26: *3D renderings of the depth map in Figure 4-23e. The motion estimation assumed pure translation only. The reconstruction has a lot more noise. The edges are less clean, especially the edge between the left and top faces. Also large errors are noticeable on the ground plane. The estimated angle between the side faces (b) is* 105^o.

4.5 Summary of Experimental Results

Using the proposed method for direct estimation of motion and structure we have found that heading direction can be estimated to within a few degrees for both purely translating motion and for motion that includes a few degrees of rotation. The estimates of camera rotation are within a fraction of a degree from the true rotations. The estimation of rotation is critical. If we neglect even small rotations (0.5^o) both heading estimates and structure estimates have large errors.

The flow due to even relatively small rotations can be quite large. This causes aliasing problems and therefore the program fails to converge on images with highly repetitive texture. While the problem is important in the case of pure translation as well, it is the typically larger image motions that make the problem more severe when rotations are involved. One solution to this problem might be to first find a rotation that best aligns the images (perhaps using a pyramid version of Horn & Weldon [34]) and only then try and find the translation and residual rotation. While this is similar to the idea of prealligning a plane it does not remove all rotation but on the other hand leaves us in the Euclidean realm.

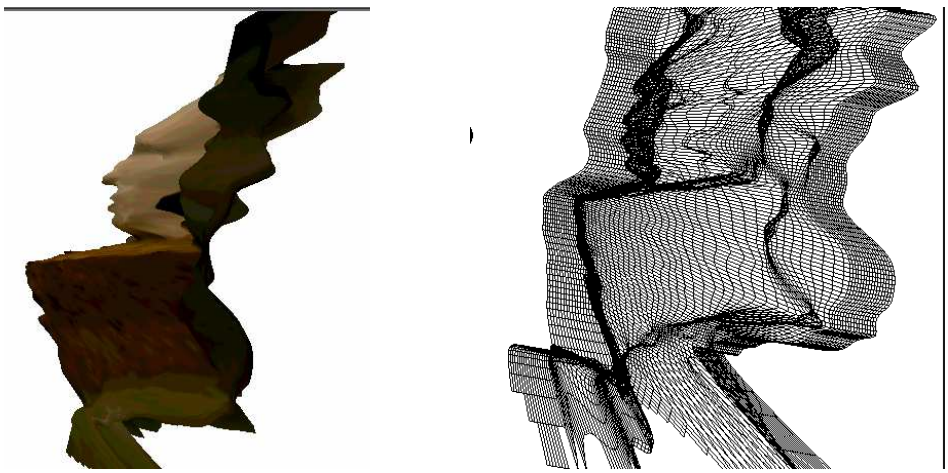
The program converges well for non-collinear purely translating motion. In the presence of collinear motion the magnitude of the translation is more critical. In the presence of rotation the program converges well if either the rotational flow does not overwhelm the translational flow or if there is a strong signal with a low spatial frequency. That is why the program work well with the sequences in Experiment 4 and Experiment 5 but does not work well with the sequence in Experiment 1.



(a) (b)

(c) (d)

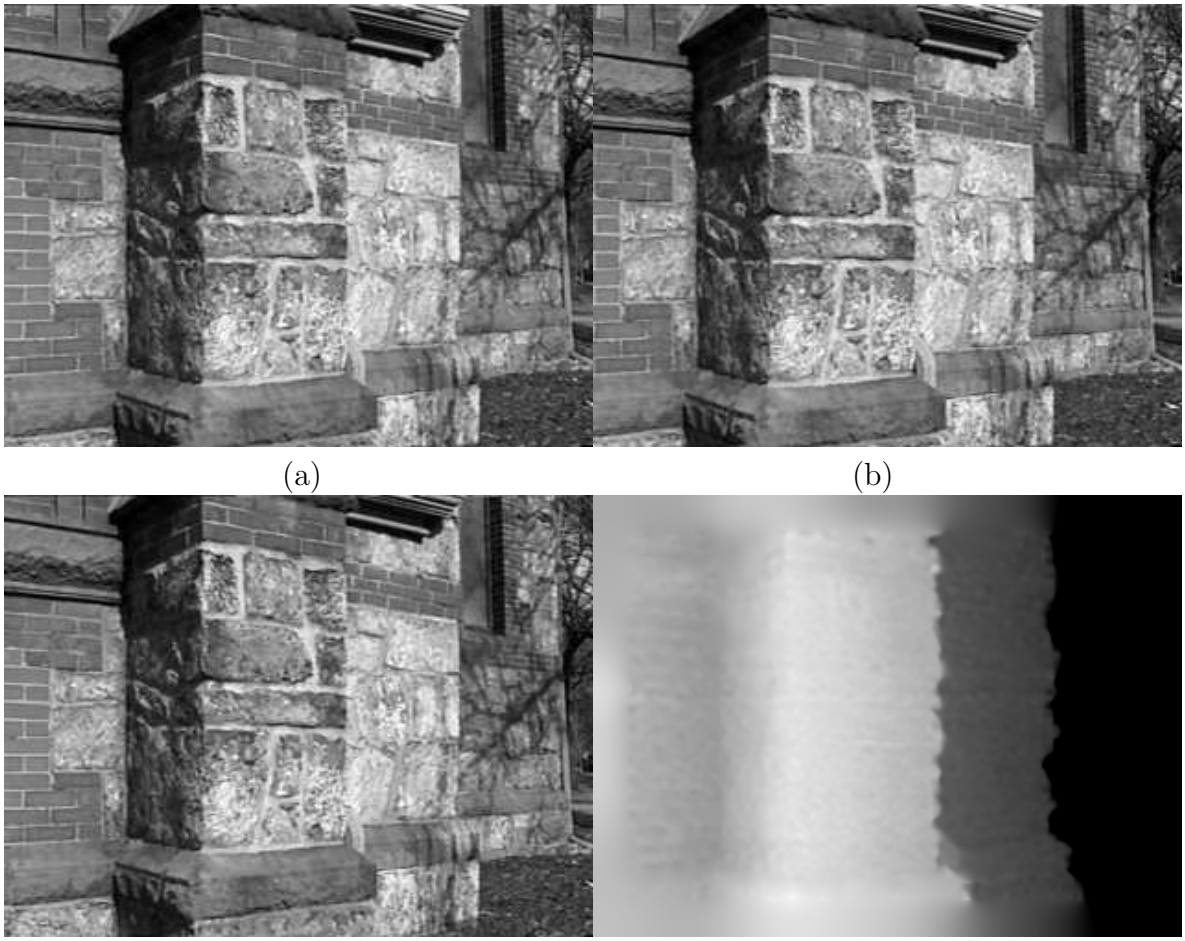
Figure 4-27: *Three input views from a hand-held camera and the resulting depth map. The motion model included both rotation and translation.*



(a) (b)

(c)

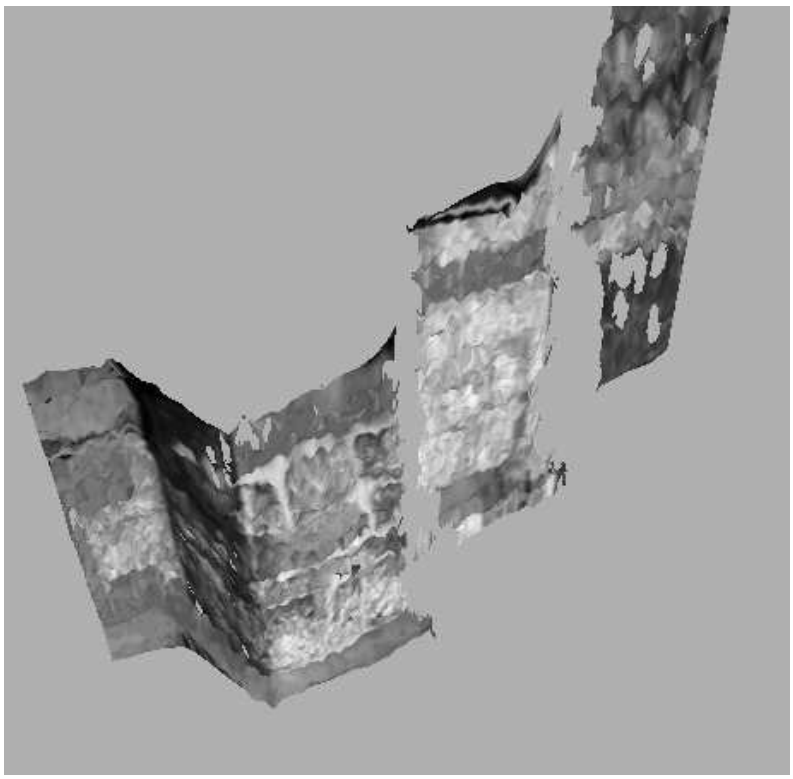
Figure 4-28: *3D renderings of the depth map in Figure 4-27d.*



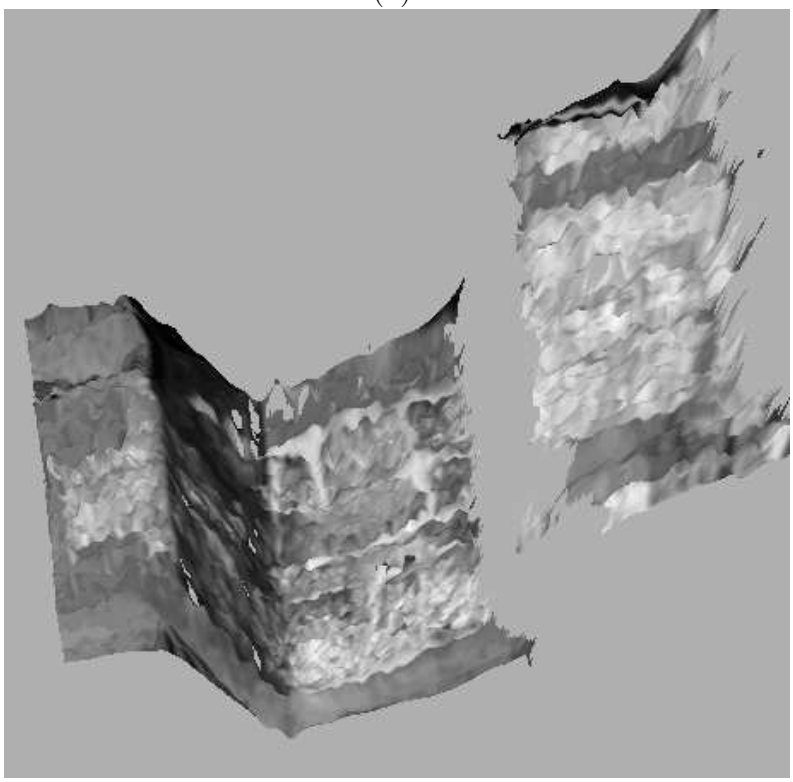
(a) (b)

(c) (d)

Figure 4-29: *Detail of church wall: Three input views and resulting depth map. (Sony Hi8 camcorder, FOV* 40^o*)*

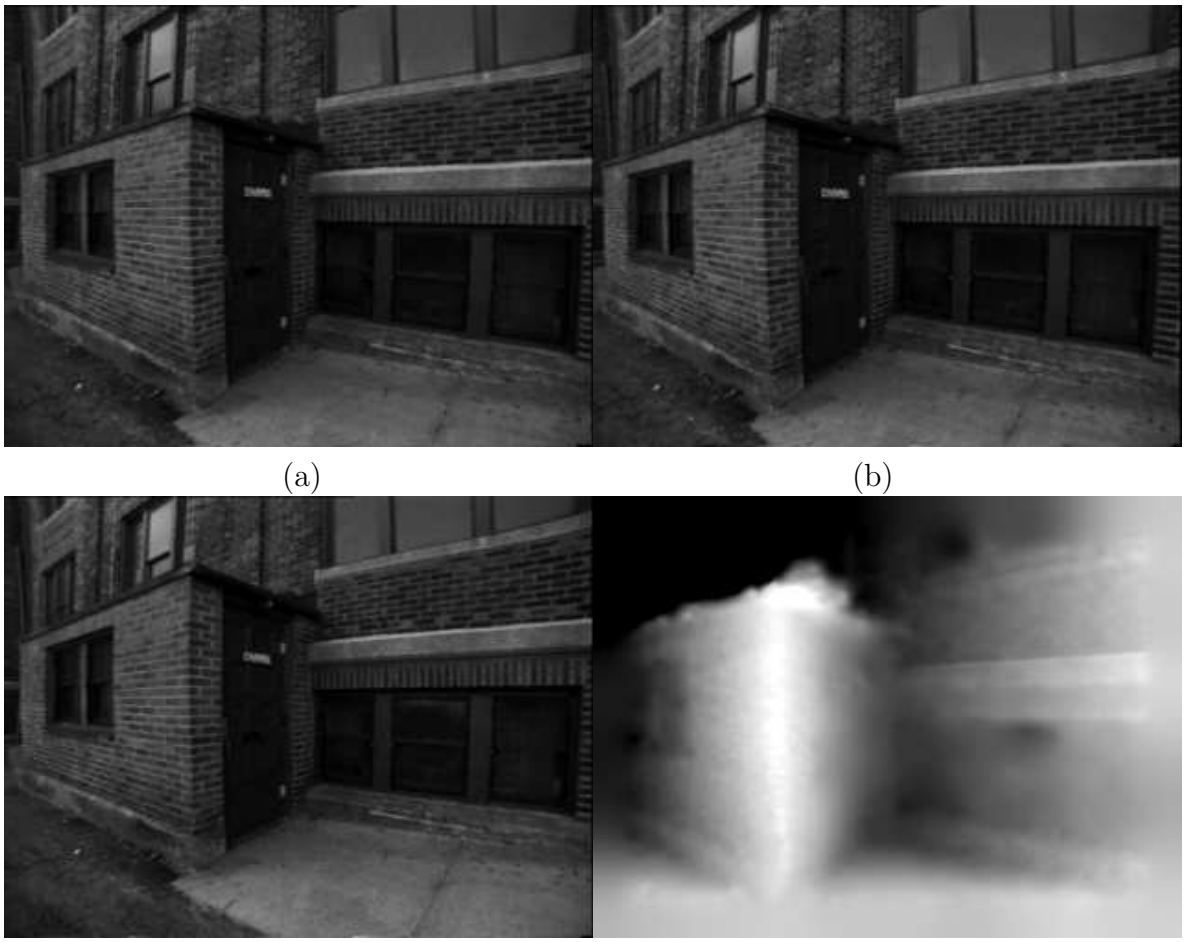


(a)



(b)

Figure 4-30: *3D Euclidean renderings of the depth map in Figure 4-29d.*



(a) (b)

(c) (d)

Figure 4-31: *Three input views of a side entrance to a building and resulting depth map. Camera was mounted on a tripod but motion included up to* 0.5^o *rotation.*

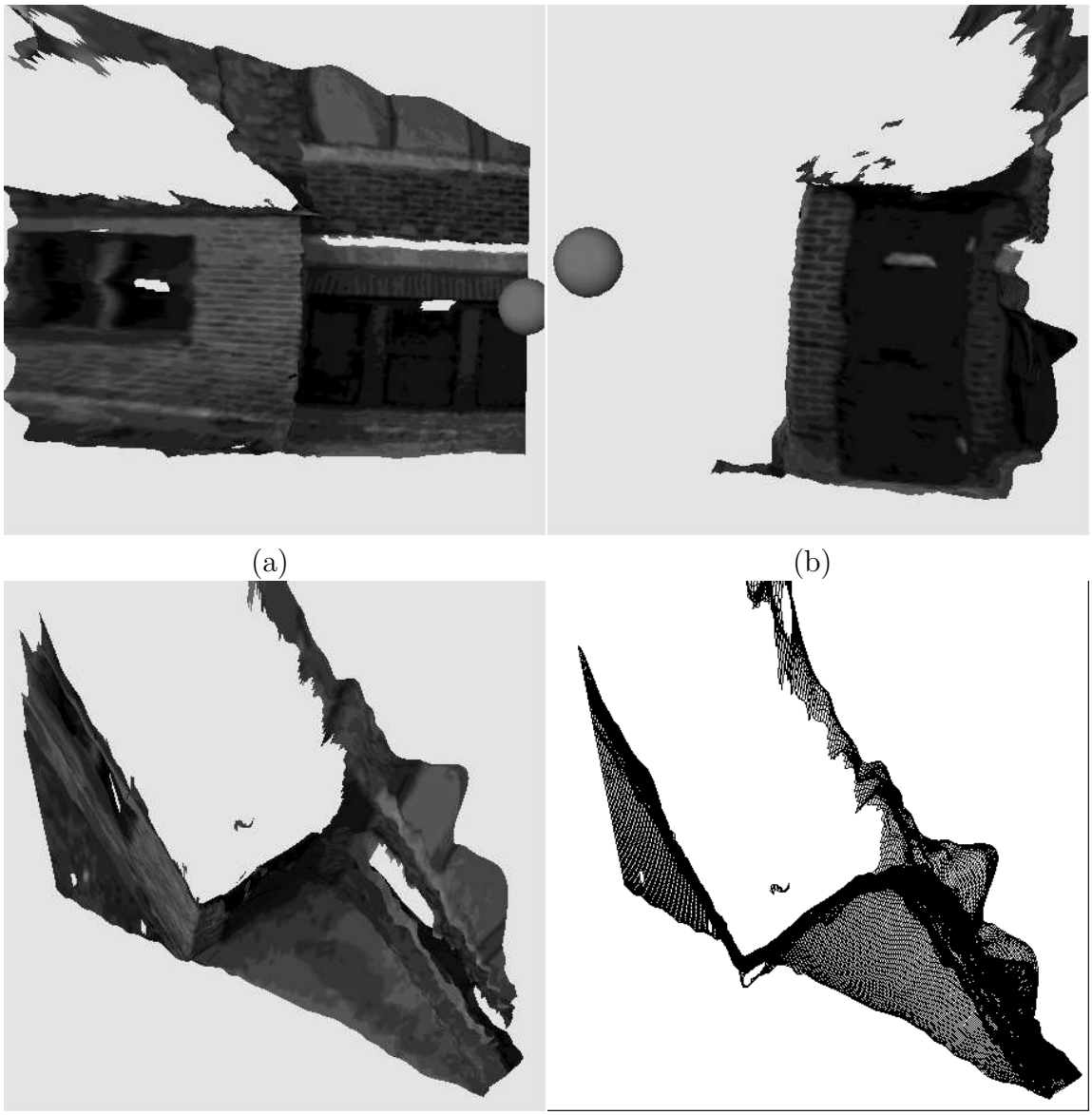


Figure 4-32: *3D renderings of the depth map in Figure 4-31d. The spheres in (a) and (b) indicate the estimated camera location. Note in overhead view (d) true Euclidean structure is recovered with correct* 90^o *angles.*

Chapter 5

Direct Estimation of Motion and Extended Scene Structure from a Moving Stereo Rig

5.1 Introduction

Stereo and motion together form a powerful combination [23, 91] with a variety of applications from ego motion estimation to extended scene reconstruction. In this chapter we describe a new method for directly estimating the motion of a stereo rig thus greatly simplifying the process of combining the information from multiple image pairs. We show that given two stereo pairs one can compute the motion of the stereo rig directly from the image derivatives (spatial and temporal). Correspondences are not required.

The core of the method is to apply the multi-view geometric constraints (Chapter 2) to a situation where some, but not all, of the camera motions are known. For example, the trilinear tensor equation (Section 2.4, equation 2.40) relates feature correspondences (points or lines) and the two camera motions. The equations are bilinear in the camera motion parameters but if one of the motions is known then the equations are linear in the remaining unknown motion parameters.

In this chapter we will derive the equations for calibrated cameras and small motion but the results can be extended to the projective case. We show that point correspondences among three views, give two constraint equations each. Three points are therefore required, and sufficient, to recover the six unknown motion parameters (the rotation and translation vectors). This highlights one of the advantages of using stereo image sequences over monocular image sequences: the motion can be recovered given a planar object. There are further advantages. The method is stable in the case of collinear motion, in the case of pure rotation and also for planar objects. There is no scale ambiguity since all motions are relative to the known baseline length [23].

In order to avoid the need to find feature correspondences we can use the model-based brightness constraints (Chapter 3), which combines geometric and photometric constraints of three views. As with feature correspondences, the model-based brightness constraint equations are bilinear in the motion parameters, but if one of the motions is known they result in a set of linear equations for the second motion parameters. If we assume calibrated cameras and small motion, the 15-parameter

model-based brightness constraint (Section 3.4) yields one equation for each point in the image. The equations are non-homogeneous linear equations in the 6 unknowns. This results in a highly over-constrained, and very stable, computation.

After the motion has been found, one can compute a dense depth map using information from the stereo pair and the motion (see Figure 5-1d). This can now be viewed as a multi-baseline stereo system with all the advantages of such as system: it reduces aliasing, helps deal with occlusions and it extends the dynamic range of the system as detailed by [38] and others. Applying an edge detector to the depth map (Figure 5-1e) highlights possible areas of occlusion. In a traffic scene, for example, these are locations where a pedestrian might suddenly appear. The motion estimates between stereo pairs enable us to combine depth maps from all the pairs in the sequence to form an extended scene reconstruction.

The typical approach to reconstruction from a moving stereo rig [39, 91, 85, 16, 88, 22] has been to first compute a depth map from each pair (or 3D location of features points). One also computes a confidence measure for each depth estimate. Then one registers the stereo pair with the current 3D model to compute the motion and finally one updates the model (using a Kalman filter or a batch method). Abdel-Mottaleb et al [1] use the motion to get a rough depth map in one image which is then used to simplify the correspondence problem between the images in a stereo pair. Waxman & Duncan [82] first use the monocular optical flow to determine depth discontinuities which cause problems in stereo matching. Then they use the relationship between stereo disparity and the change in stereo disparity to help resolve the correspondence problem for stereo and also compute the motion of the stereo rig. Tirumalai et al [78] use stereo depth values and initial motion estimates to help matching features over the images sequence.

Navab et al [51], in contrast to other approaches, combine motion and stereo. This is applied to the reconstruction of motion and structure from line correspondences. Given two distinct non-parallel lines in two stereo image pairs they obtain a set of 8 linear equations in the 6 motion parameters. The method requires accurate tracking of straight lines in an image and it still uses what is fundamentally, two-view geometry.

Directly computing the motion of the stereo rig simplifies the problem of registering all the depth reconstructions into one coordinate frame. There is no need to explicitly compute confidence values for the depth estimates since the causes for error are taken into account implicitly in the equations. Our method gives good estimates of the camera motion between frames. By concatenating these motion estimates one can bring all the depth maps into one coordinate frame. Combining incremental motion in this way will accumulate errors. To get a more accurate reconstruction one can use the results as a starting point for global reconstruction schemes such as [21, 23].

5.1.1 Overview of the Chapter

Based on the geometric constraints on three views (Section 2.4), we derive, in Section 5.2, the geometric constraints for a moving stereo rig based. In Section 5.3 we briefly recall the model-based brightness constraints and then derive the constraint in the case where one motion is known. This enables us to avoid having to find feature

correspondences. Section 5.4 describes some of the implementation details. In particular we describe a simple two step procedure to calibrate the stereo pair. Section 5.5 shows results for an extended scene reconstruction.

5.2 Geometric Constraints for Combined Motion and Stereo

As in Chapter 2 let $P \in \mathcal{P}^3$ be a 3D point and its image in the three views $p \in \psi$, $p' \in \psi'$ and $p'' \in \psi''$. As we recall:

$$\begin{array}{rcl} p' & \cong & \mathbf{A}p + \rho t' \\ p'' & \cong & \mathbf{B}p + \rho t'' \end{array} \tag{5.1}$$

Let s' and s'' be lines through points p' and p'' respectively. Premultiplying the left and right hand sides of the first equation by $s'^\top$ and the second equation by $s''^\top$ we get:

$$\begin{array}{rcl} s'^\top \mathbf{A}p + \rho s'^\top t' & = & 0 \\ s''^\top \mathbf{B}p + \rho s''^\top t'' & = & 0 \end{array} \tag{5.2}$$

Eliminating ρ from the above equations results in the equation:

$$s'^\top t' s''^\top \mathbf{B}p - s''^\top t'' s'^\top \mathbf{A}p = 0. \tag{5.3}$$

In Section 2.4 we rearranged the terms to arrive at the trilinear tensor equations (2.40). It is clear from the above equation, that if the homography matrix $\mathbf{A}$ and the epipole t' are known, the equation is linear in the matrix $\mathbf{B}$ and the second epipole t''.

Assuming calibrated cameras and small rotation (as in Section 3.3) we have:

$$\begin{array}{rcl} \mathbf{A} & = & [\mathbf{I} + [w']_x] \\ \mathbf{B} & = & [\mathbf{I} + [w'']_x]. \end{array}$$

To simplify things even further we can assume that the stereo pair images have been rectified so that: $\mathbf{A} = \mathbf{I}$. This leads to:

$$s'^\top t' s''^\top p + s'^\top t' s''^\top [w'']_x p - s'^\top p s''^\top t'' = 0. \tag{5.4}$$

Rearranging the terms we get:

$$s'^\top p s''^\top t'' - s'^\top t' v''^\top w'' = s'^\top t' s''^\top p \tag{5.5}$$

where $v'' = p \times s''$. The equation is a non-homogeneous linear equation in the unknowns: t'' and w''.

This equation relates a point p in Image 1 and lines s' and s'' passing through the corresponding point p' and p'' in Image 2 and Image 3 respectively. Using the *standard* form where s' (and s'') represent either the vertical or horizontal scan lines, i.e.,

$$s' = \begin{pmatrix} -1 \\ 0 \\ x' \end{pmatrix} \quad or \quad s' = \begin{pmatrix} 0 \\ -1 \\ y' \end{pmatrix}$$

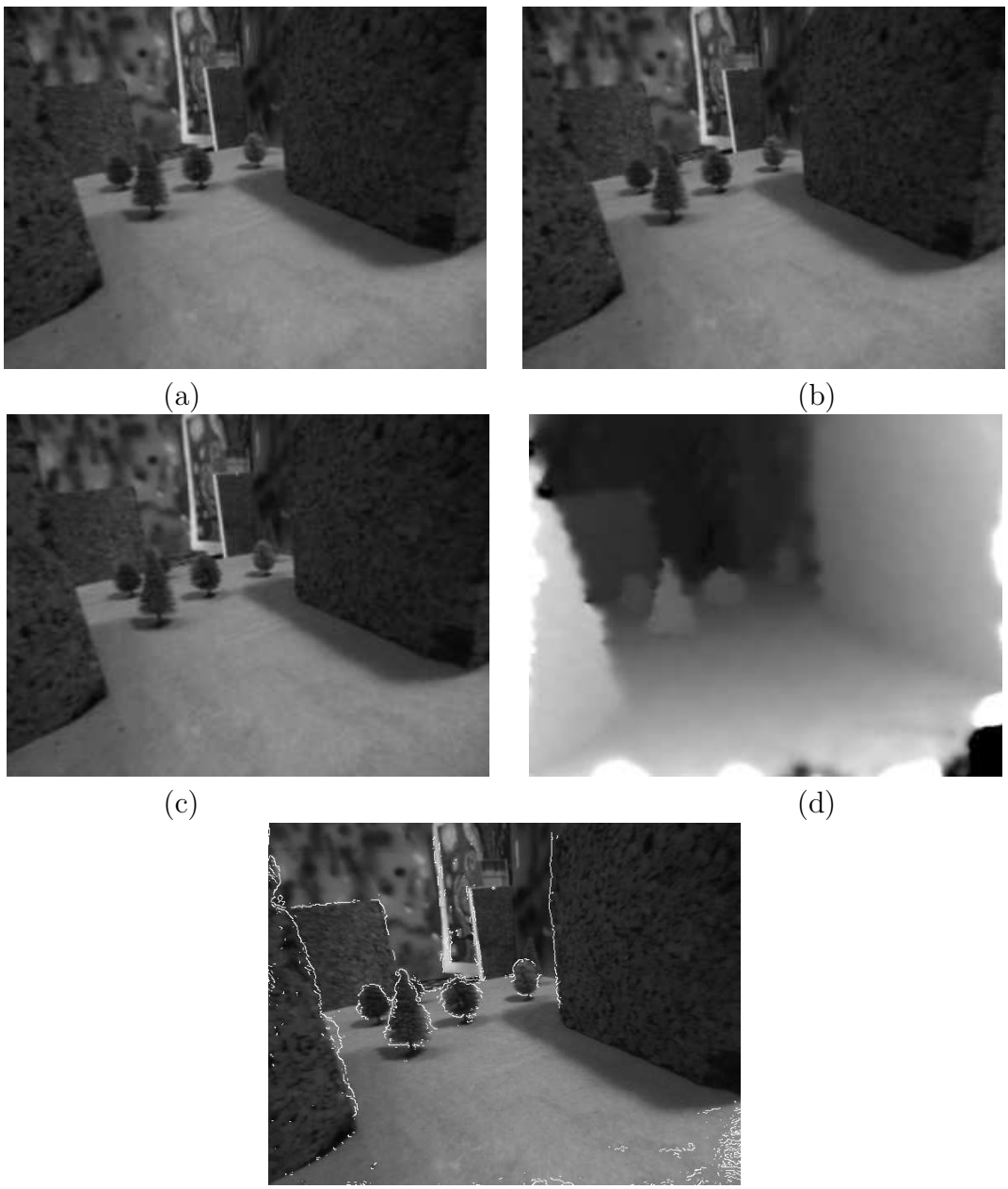


Figure 5-1: *The three input images (a,b,c). Image (a) and (c) are the first stereo pair in the sequence with the cameras displaced vertically. Image (b) is the image from camera (a) taken at the next time instance. (d) the estimated depth map.. (e) shows the results of a Canny edge detector on the depth map. This identifies occluding contours.*

(a)

(b)

(c)

Figure 5-2: *(a,b,c) Show 3D renderings of the scene. Very distant surfaces have been removed. The spheres show the motion of the stereo rig in the whole sequence. The large sphere is the the location of the camera when this image was taken. Note in view (b) that the ground plane is correctly found to be flat and the two cork walls are at right angles to the it.*

and

$$s'' = \begin{pmatrix} -1 \\ 0 \\ x'' \end{pmatrix} \quad or \quad s'' = \begin{pmatrix} 0 \\ -1 \\ y''. \end{pmatrix}$$

we get 4 equations for each point correspondence in 3 views. But we find that all lines s' yield the same equation scaled by a scalar value. Thus we have only two linearly independent equations (corresponding to the two choices of s'') from each point correspondence. Three points (in general position) will give $6 = 3 \times 2$ equations which is sufficient to recover the second camera motion parameters.

That three points is sufficient is not a surprising result. With the traditional method for combining motion and stereo, the stereo pair provides the 3D coordinates of the three points before and after the motion. Given the 3D coordinates of three points in two images the relative orientation can be determined [32].

5.3 The Brightness Constraint of a Moving Stereo Rig

The Tensor Brightness Constraint is developed in Section 3.2. Assuming calibrated cameras and the Longuet-Higgins and Prazdny model we obtained (Section 3.4) the 15-parameter *model-based brightness constraint*:

$$\boxed{I_t'' s^\top t' - I_t' s^\top t'' + s^\top [t' w''^\top - t'' w'^\top] v = 0} \tag{5.6}$$

where s, v are defined below:

$$s = \begin{pmatrix} I_x \\ I_y \\ -xI_x - yI_y \end{pmatrix} \quad v = p \times s = \begin{pmatrix} -I_y - y(xI_x + yI_y) \\ I_x + x(xI_x + yI_y) \\ xI_y - yI_x \end{pmatrix} \tag{5.7}$$

Note:

- There is one such equation for each point in the image.
- The equation involves image gradients and image coordinates in Image 1 only. It does not require correspondences (x', y', x'', y'').
- Since it is based on the optical flow constraint equation it is correct only for infinitesimal motion.
- In the case of collinear motion (i.e. $t' \propto t''$) the system of equations becomes degenerate.

Let us now assume that t' and w' are known. We can then rewrite equation (5.6) in the form:

$$(I_t' + w'^\top v) s^\top t'' - (s^\top t') v^\top w'' = I_t'' s^\top t' \tag{5.8}$$

which is linear in the unknown translation and rotation, (t'' and w'' respectively). If the stereo pair has been rectified (i.e. $w' = 0$) then equation (5.8) can be further simplified to:

$$I'_t s^\top t'' - (s^\top t')v^\top w'' = I''_t s^\top t'. \tag{5.9}$$

The above derivation was performed for the case of calibrated cameras but an equivalent derivation is possible using uncalibrated cameras for the projective case.

Note:

- Equations (5.8) and (5.9) are non-homogeneous equations and the set of equations can be solved even if the translation $t'' = 0$.

5.4 Implementation Details

For a general discussion of implementation issues see Section 3.7. We will repeat some of them briefly here in order to describe the exact choices and parameter values used in this application. In particular, the issues of calibration are unique to this application.

5.4.1 Calibration

We assume the internal parameters of the first camera are known. These can be found using methods described in [72]. For true Euclidean reconstruction an accurate estimate of the focal length is required but the whole process degrades gracefully when only approximate values are provided.

Calibration of the stereo pair is performed in two stages. First we take an image of a distant scene (the plane at infinity) and find the homography between Image 2 and Image 1 using the method described in [36]. Since the rotation angle is small we can assume an affine model rather than a full planar projective transformation. This stage takes into account both the rotation between the two cameras and also the variation in internal camera parameters. We can now use this mapping (projective or affine) to preprocess all the images coming from Camera 2. This preprocessing stage is often called rectification.

The second stage is to find the translation between the two cameras. We move the whole stereo rig in pure translation. We then use equation (3.20) which gives accurate results under the assumption of pure translation [74] to compute both the translation of the rig and the displacement between the two cameras.

Lens Distortion

Since we us are using a lens with a wide field of view there is noticeable lens distortion. This does not affect the stability of the method but the accuracy of the motion estimates is reduced and the 3D reconstruction suffers non-projective distortion. Flat surfaces and straight lines appear slightly curved. A variety of methods to compute lens distortion appear in the literature (see [73]) and a new method is presented in Chapter 7).

5.4.2 Computing Depth

To compute the depth at every point we use equations (3.16). Information is combined from both image pairs and over a local region by minimizing:

$$\min_{K} \arg \sum_{x,y \in R} \sum_{j} \beta(x,y) |s^T t^j|^p \left(k s^T t^j + v^T w^j + I_t^j \right)^2 . \tag{5.10}$$

For more details see Section 3.7.1.

- The windowing function $\beta(x, y)$ allows one to increase the weight of the closer points.
- The $|s^T t^j|^p$ term reduces the weight of points which have a small gradient or where the gradient is perpendicular to that camera motion since these cases are highly affected by noise. We used $p = 1$.
- During the iteration process we used a region R of 7×7.
- After the last iteration, we reduced the region R to 1×1 but added a very weak global smoothness term and performed multi-grid membrane interpolation. This stabilizes regions where there is no image gradient.

5.4.3 Coarse to Fine Processing and Iterative Refinement

In order to deal with image motions larger than 1 pixel we use a Gaussian pyramid for coarse to fine processing [5, 10]. For a 640×480 image we used a 5 level pyramid.

The linear solution can be thought of as a single iteration of Newton's method applied to the problem. At each level of the pyramid we iterate as follows:

1. Calculate motion (using equation 5.8).
2. Compute depth (using equation 5.10).
3. Using the depth and motion, warp Image 2 and Image 3 towards Image 1.
4. Compute new time derivatives I_t' and I_t'' by the pixelwise subtraction of Image 1 from the warped Image 2 and warped Image 3 respectively.
5. Compute a new motion and depth estimate using equations (5.8) and (5.10).

As explained in Section 3.7.2, one cannot simply compute the incremental model from the previous iteration because as the iterations proceed the system of equations of the incremental model will become badly conditioned. We followed the procedure described in Section 3.7.2.

The situation here is a bit different since there is an asymmetry between t' which is due to the separation between the two cameras in the stereo pair and t'' which is the motion of Camera 1. Equation (5.8) results in a non-homogeneous set of equations

and we do not enforce $||t''|| = 1$ as we did when both translations were unknowns. Thus, we could in fact warp Image 3 towards Image 1 (but not Image 2 towards Image 1) and compute only the corrections to the motion $\delta t''$ and $\delta w''$. We have not implemented this alternative method.

At the finest level (640×480) we performed 2 iterations and we recursively doubled the number of iterations at the coarser levels. We can afford to do this because the number of computations per iteration at each level drops by a factor of 4.

After we have computed the structure and motion at the finest level we keep the motion constant and repeat the depth computation down the whole pyramid. This fixes a few areas particularly near the borders of the image.

5.5 Experiments

5.5.1 Experimental Details

A single camera was mounted on a 3 degree of freedom motion stage (horizontal and vertical translation and rotation around the vertical axis). A stereo pair of images was captured by translating the camera vertically by $8.4mm$ between images. This in effect means that the first stage of calibration (Section 5.4) is not required and that none of the measurement error can be attributed to different internal geometric or photometric parameters between the cameras. Initial experiments (not reported here) have been performed using two cameras mounted vertically one above the other. These show qualitatively similar results to those presented but quantitative results are not available.

The camera was a high resolution BW CCD camera with a $4.9mm$ lens giving a corner to corner viewing angle of 100^o (82^o FOV across the y axis). The images were captured at 640×480 resolution.

5.5.2 Extended Scene Reconstruction

For the 'fly through' sequence (Figure 5-1) the camera was mounted on a extension bar which positioned the camera's center of projection $400mm$ from the axis of rotation (see Figure 5-3). Thus a rotation of 1.5^o produces a translation of $10.5mm$. The camera was mounted facing 20^o down from the horizontal and 10^o inwards. The focus of expansion is therefore inside the image towards the top right.

The extension bar perpendicular to the translation stage will be denoted an angle of $\theta = 0^o$. The bar was moved in 1.5^o decrements from a starting position of $\theta = 45^o$ to a position of $\theta = 0^o$. The camera was then translated in $10mm$ increments through a distance of $120mm$ and then rotated again 1.5^o decrements to $\theta = -12^o$. At each camera location, the vertical stage motion provided a stereo pair.

The camera motion and depth map was computed for each motion in the sequence. The motion estimates are plotted in Figure 5-4. Figure 5-1, Figure 5-6 and Figure 5-7 show four example reconstructions made at four points along the path. Areas which were very close or very far ($60 \times baseline$) from the camera are masked out and not

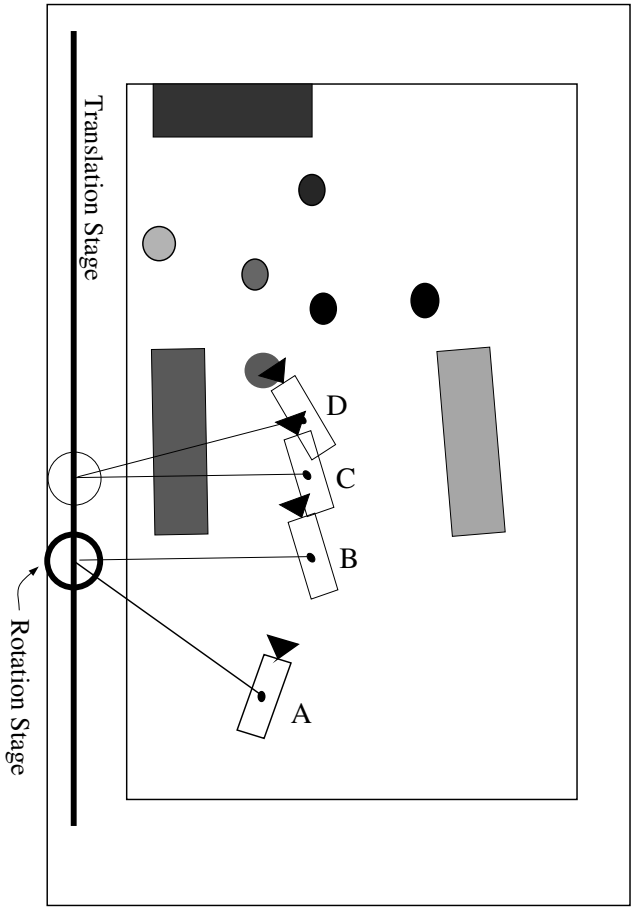


Figure 5-3: *Schematic overhead view of the 'fly-through' scene indicating the three cork blocks, 6 model tree (circles) and the motion stage (not to scale). The camera arm is rotated* 45^o *in* 1.5^o *steps from position A to position B. The arm then translates in twelve* $1cm$ *steps to position C. Then it Rotates through* 12^o *to position D.*

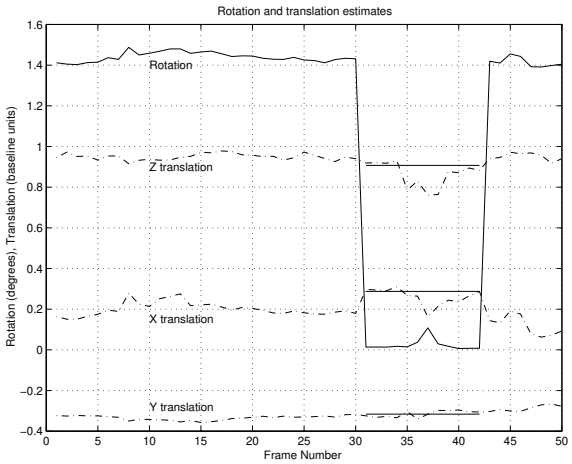


Figure 5-4: *Estimated Rotation and translation. The rotation angle (solid lines) should be* 1.5^o *and is on average* 1.45^o*. The translation estimates (dot-dashed lines) are plotted along with the true values for the pure translation section (short solid lines).*

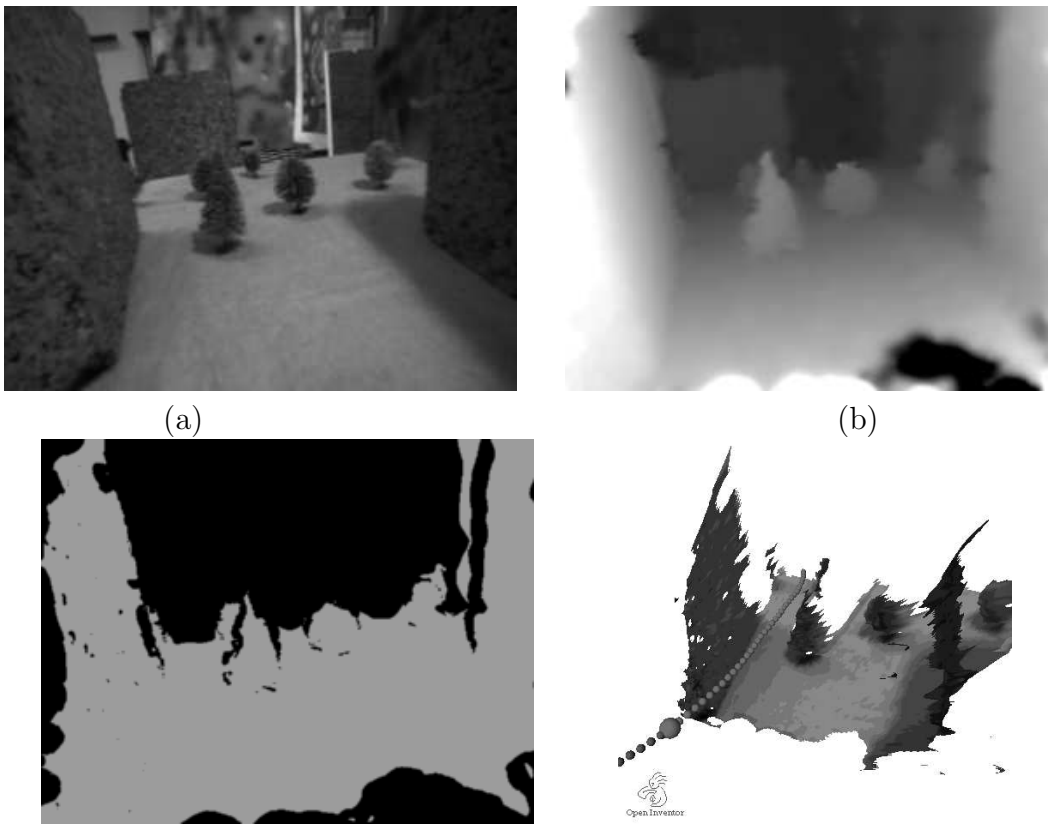


Figure 5-5: *(a) The 14th image in the sequence. (b) The depth map. (c) The mask. We do not render surfaces that are far from the camera or where the surface is facing the camera with a sharp angle. (d) rendering of the surfaces.*

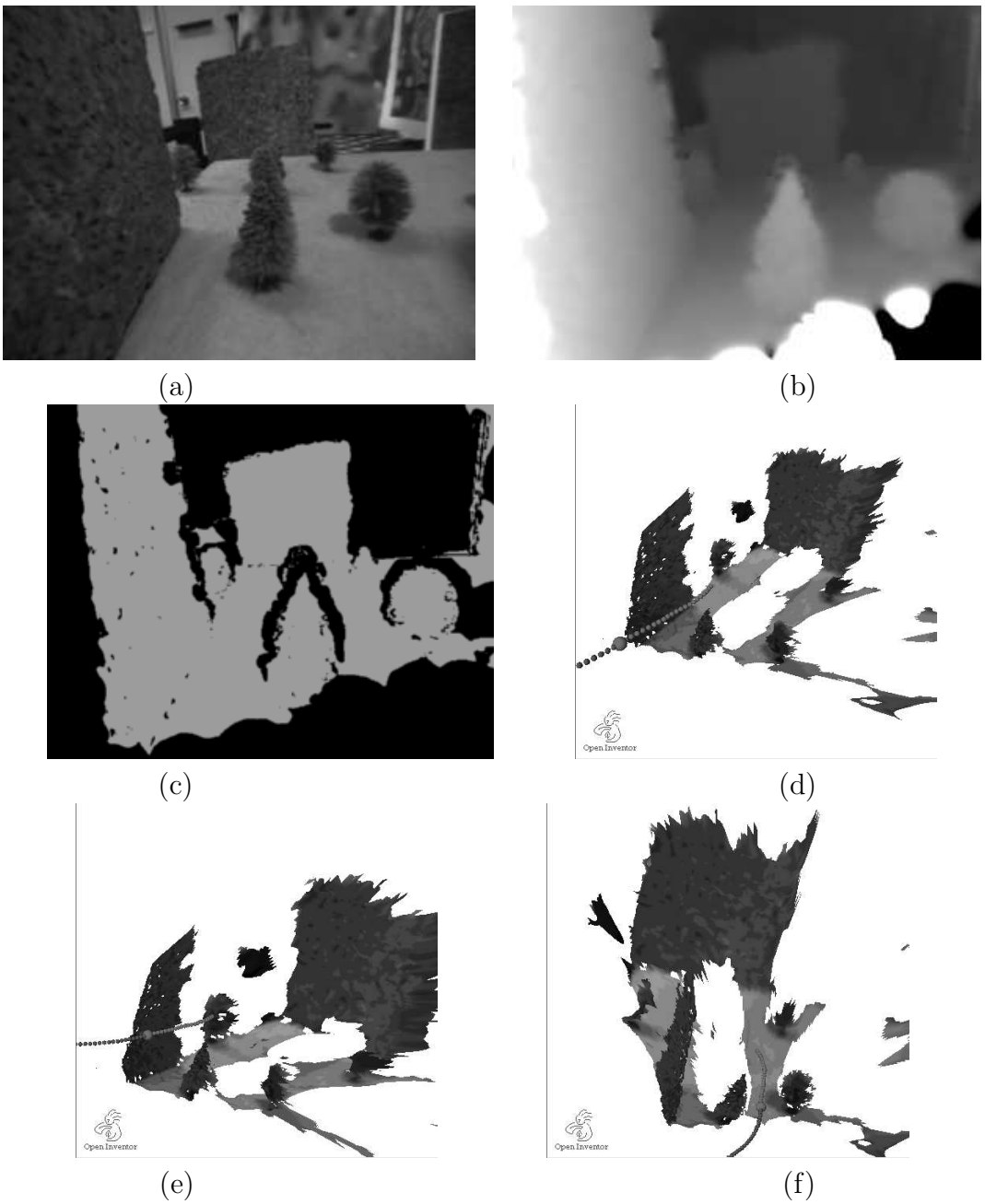


Figure 5-6: *(a) The 28th image in the sequence. (b) The depth map. (c) The mask. We do not render surfaces that are far from the camera or where the surface is facing the camera with a sharp angle. (d), (e),(f) rendering of the surfaces.*

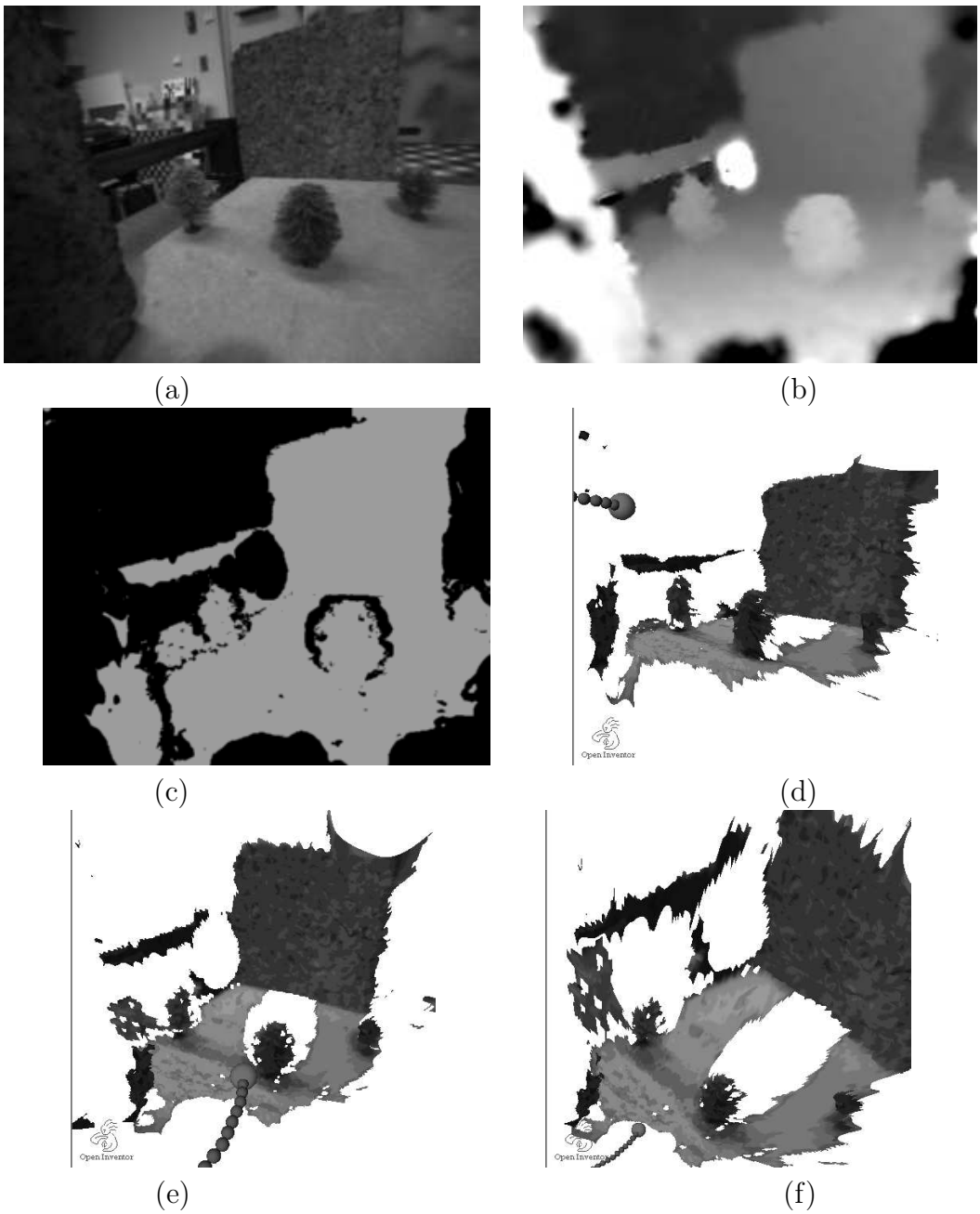


Figure 5-7: *(a) The 49th image in the sequence. (b) The depth map. (c) The mask. (d), (e),(f) 3D renderings of the surfaces. In this view a bit of the translation stage can be seen (top left quadrant).*

rendered and so are points where the (inverse) depth map has a large gradient since these are points where depth reconstruction is inaccurate. Figure 5-1e shows the results of a Canny edge detector applied to the (inverse) depth image highlighting potential occluding contours.

Figure 5-8 shows the four reconstructions aligned using the estimated camera motions. The scene has more information than can be seen from any one camera.

5.6 Discussion and Future Work

We have presented a new method for recovering motion and structure from stereo image sequences. The method gives good motion estimates with errors less than 5% but this is not a zero mean error and there is a clear bias even when the focus of expansion is inside the field of view. Initial experiments show that the bias increases when the translation has a large component parallel to the image plane.

While we have shown that this method can give good motion estimates these are only incremental motion estimates. A full system could incorporate this method as the estimation stage inside a Kalman filter or batch global estimation framework. The improved accuracy of the estimates provided by this method should improve the convergence properties of the whole system. We have also not dealt with the issue of how to represent the final 3D reconstruction. At this point we draw multiple $2\frac{1}{2}D$ surfaces. Better methods are described in [21].

The idea of using multi-view geometric constraints to recover the motion of a stereo rig can also be applied to feature correspondences, either point or line features. For example, let us denote the i'th image pair Images I'_i and I''_i. Then if we use line features, then every line correspondence from Images I'_i,I''_i and I'_{i+1} will provide us with two constraint equations. Thus, three line correspondences will give us enough equations to solve for the six motion parameters. Navab et al [51] have shown that two line correspondences is enough. We can trace this inconsistency to the fact that we do not use Image I''_{i+1}. It was not used because Faugeras & Mourrain [17] show that there are no constraints among four (or more) images that are not simply two and three image constraints. Thus it was assumed that other image triplets would provide redundant equations (which also be used to give better noise properties). It is probably the case that when some of the motions are known, using other image triplets will give additional linearly independent constraint equations. This is an open question for future study.

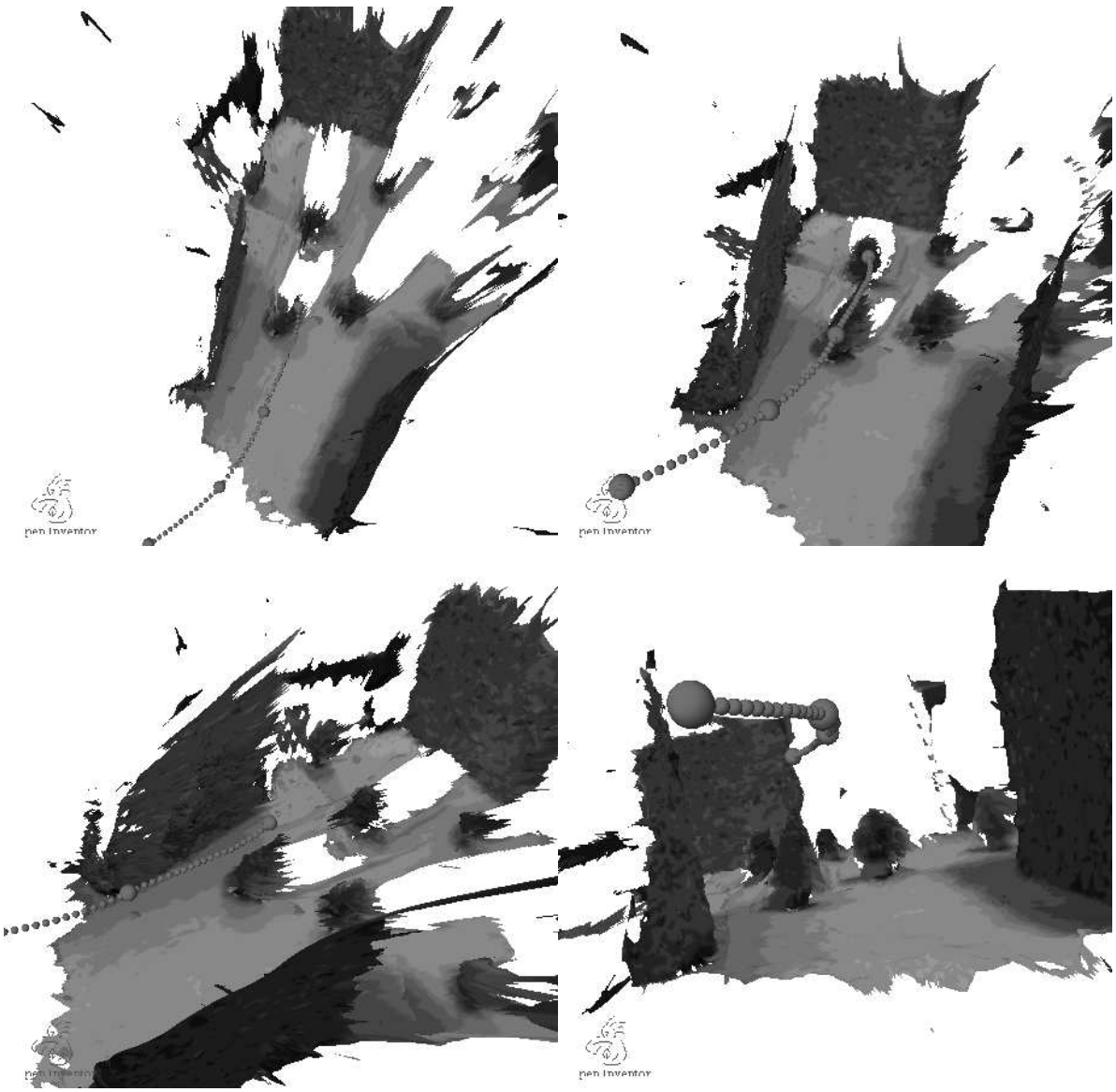


Figure 5-8: *Euclidean reconstruction of the extended scene created by aligning four separate reconstructions into one coordinate system using the recovered camera motion. The small spheres show the camera path. The large spheres show the camera positions of the four Euclidean reconstructions.*

Chapter 6

On Degeneracy of Linear Reconstruction from Three Views: the Linear Line Complex

6.1 Introduction

It is known that point and line image features across three perspective views can generally give rise to a linear system of equations for a unique solution for 3D structure and camera motion. The camera and motion parameters are represented by a $3\times3\times3$ tensor. The image measurements of matching points and lines provide constraints, trilinear in image coordinates, that as a whole make a linear system of equations for the (unknown) coefficients of the tensor. Finally, the tensor has only 18 degrees of freedom, i.e., the 27 coefficients are subject to non-linear admissibility constraints. Details of the geometry of three views and the trilinear tensor were presented in Chapter 2. In the presence of errors in image measurements one often starts with the Linear solution and improves it further by employing a numerical Gauss-Newton style iterative procedure until a solution that satisfies the admissibility constraints is obtained.

In this chapter we investigate the cases in which the linear solution is degenerate. As it happens, the degeneracy occurs in typical real situations. We show that when the sample of features is taken from a configuration of lines that have a common intersection, known as a *Linear Line Complex* (LLC), then the rank of the linear system reduces from 26 (in the general case) to 23 — yet, there exists a *unique* solution for the tensor when the non-linear admissibility constraints are accounted for[1]. An LLC includes in particular the case of lines on parallel planes whose degeneracy was observed in [84].

To appreciate the practical importance of investigating LLC configurations, consider a few typical outdoor and indoor scene examples depicted in Figure 6-1. In Figure 6-1a the common intersecting line is the edge of the building. All horizontal lines on the two faces of the building meet the edge in the image plane, and the vertical lines meet the edge at infinity. Note also that the vertical line representing the lamp-post also meets the edge of the building (at infinity) thereby included in the

[1]This in contrast to *critical line configurations* from which a unique solution is not possible, see [44, 9, 50]

LLC configuration. This leaves very few lines (the sidewalk and the oblique line of the lamp-post) not part of the LLC. The common line in Figure 6-1b is the edge of the book-case leaning on the wall. All vertical lines in the scene intersect the common line at infinity and the horizontal lines of the file cabinet and the arm chair intersect it in the image plane. Again, very few lines in the scene do not belong to the LLC (the horizontal line segments of the box near the far end wall and two horizontal line segments attached to the book-case). Next, imagine a robot moving down the hallway in Figure 6-1c. All the vertical and horizontal lines are on a set of parallel plane and so belong to an LLC. The rest of the lines are in the direction of the robot's (the camera's) motion and are all degenerate. Figure 6-1d is an LLC (with the exception of one short line segment) where the common line is the meeting between the file cabinet and the wall on the right.

Finally, an LLC situation occurs also as an artifact in direct estimation of the Tensor from spatio-temporal derivatives of image brightness (see Chapter 3). The spatio-temporal derivatives provide an axis of certainty (a one-dimensional uncertainty) for the location of the matching points in Images 2 and 3 relative to points in the reference Image 1. The uncertainty axes in Images 2 and 3 are parallel which means that the information gathered from a *general* scene by means of first-order spatio-temporal derivatives is at most comparable to the information gathered from an LLC configuration of discrete matching lines.

Given our main result, an attempt to reconstruct structure and motion from the image line information of the scenes in Figure 6-1 using conventional approaches would be at best unstable. The linear system of equations is singular or near singular, and would most likely not serve as a reasonable starting solution for the subsequent Gauss-Newton iterations. Therefore an investigation into degeneracies caused by an LLC and their remedy is important also in practice.

The remainder of the chapter is organized as follows. Section 6.2 contains the main results which include the statement of degeneracy of the linear system forming a null space of dimension 4, and the statement of uniqueness by incorporating the admissibility constraints with a simple constructive algorithm for obtaining a unique solution from an LLC configuration. In Section 6.3 we verify the theory and the algorithm with experiments with real images. We use a schematized version of the real scene shown in Figure 6-1a because it allows for a wider set of experiments. One can accurately find both line and point correspondences. One can therefore perform the motion estimation using line correspondences and then verify the results against motion estimates obtained using points.

6.1.1 Notations

Notations in general, and tensorial notations in particular, theory and background of the trilinear tensor with its contractions, slicing properties and admissibility constraints, are discussed in Chapter 2. In this chapter we add the following notation conventions: The $3 \times 3 \times 3$ trilinear tensor $\mathcal{T}_i^{jk}$ might often be written explicitly as the three matrices $(\mathcal{T}_1^{jk}, \mathcal{T}_2^{jk}, \mathcal{T}_3^{jk})$. Additionally, the tensor might be written as a 27 element vector v where we assume there is a consistent one to one mapping between

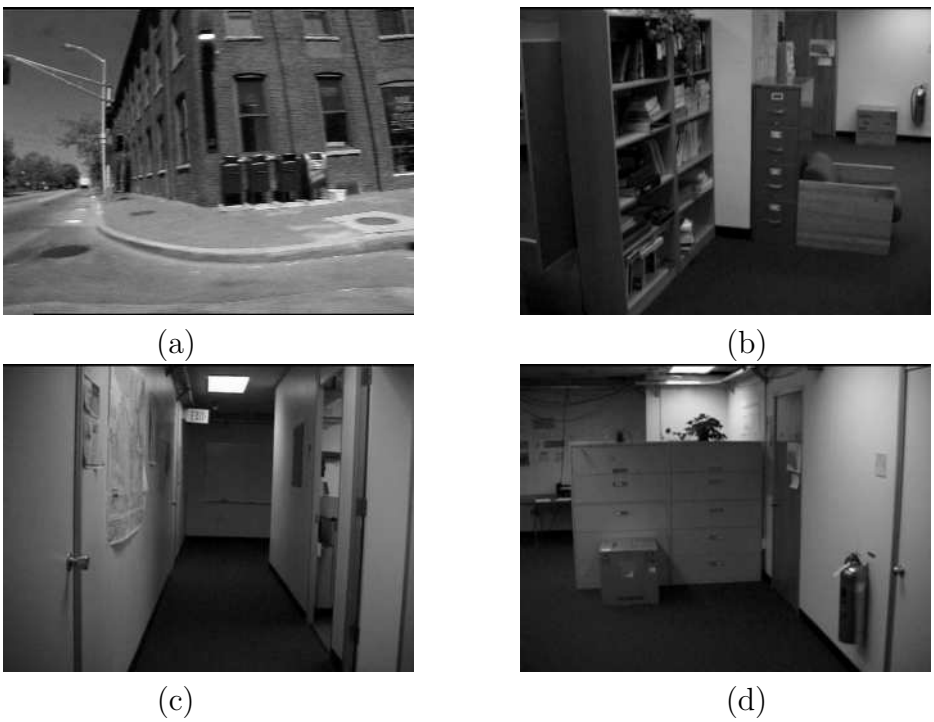


Figure 6-1: *Typical urban indoor and outdoor scenes. Most of the lines in the images form a Linear Line Complex. See text for more details.*

the tensor and vector notations.

6.2 Linear Line Complex Scene Structure

Consider the tensor:

$$\mathcal{T}_i^{jk} = v'^j b_i^k - v''^k a_i^j \tag{6.1}$$

applied to the point-line-line configuration:

$$s'_j s''_k (p^i \mathcal{T}_i^{jk}) = 0, \tag{6.2}$$

where p is a point in Image 1 and s', s'' are lines coincident with the matching points p', p'' in Image 2 and Image 3, respectively. (For the derivation of equations (6.1) and (6.2) see Section 2.4.) Note that $p^i \mathcal{T}_i^{jk}$ is a 3×3 matrix determined by p, which we will denote by $\mathbf{B}_p$. Using matrix notation the above equation becomes:

$$s''^\top \mathbf{B}_p s' = 0$$

for all pairs of lines coincident with p', p''. Assume that there exists a matrix $\mathbf{B}$, independent of p, such that $s''^\top \mathbf{B} s' = 0$, then clearly the tensor $\mathcal{T}_i^{jk}$ is not unique: slice the tensor into three matrices $(\mathcal{T}_1^{jk}, \mathcal{T}_2^{jk}, \mathcal{T}_3^{jk})$, then the tensors $(\mathbf{B}, \mathbf{0}, \mathbf{0})$, $(\mathbf{0}, \mathbf{B}, \mathbf{0})$ and $(\mathbf{0}, \mathbf{0}, \mathbf{B})$ (and their linear combinations) all satisfy the constraint $s'_j s''_k p^i \mathcal{T}_i^{jk} = 0$. Such a matrix $\mathbf{B}$ does not exist in general. We may, nevertheless, ask *whether there exists a special configuration of points and lines in space for which such a matrix* $\mathbf{B}$ *is valid?* Such a configuration is a Linear Line Complex (LLC):

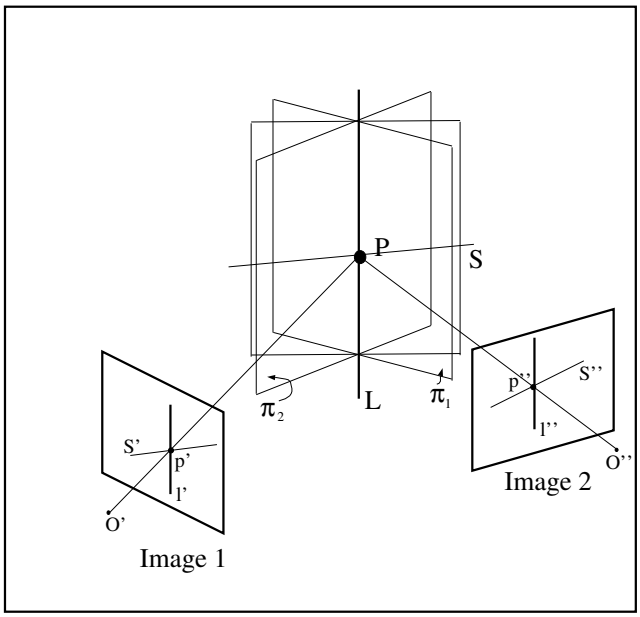


Figure 6-2: *Figure to accompany Theorem 1*

Theorem 3 *Let $\mathcal{S}$ be a set of lines in 3D which have a common intersecting line L (i.e., $S \wedge L = 0$ for all $S \in \mathcal{S}$). Let $\mathcal{Q}$ be a set of lines in 3D that intersect the line joining the two camera centers. Then, there exists a unique matrix $\mathbf{B}$ satisfying $s''^{\top}\mathbf{B}s' = 0$ for all pairs of projections s', s'' of lines $S \in \mathcal{S}$ onto two distinct views (images). The matrix $\mathbf{B}$ also satisfies $q''^{\top}\mathbf{B}q' = 0$ for all pairs of projections q', q'' of lines $Q \in \mathcal{Q}$.*

Proof: Let P be the intersection of a line $S \in \mathcal{S}$ with L and denote its projections by p', p'' onto Views 1 and 2 respectively (see Figure 6-2). Choose any plane π from the pencil of planes meeting at the line L, and let $\mathbf{H}_\pi$ be the corresponding 2D projective mapping (homography matrix) of points in View 1 to points in View 2 via (projections of) the plane π. Since π contains the line L, then

$$\mathbf{H}_\pi p' \cong p''.$$

Let l', l'' be the projections of L, then p' is the intersection of s' and l', thus,

$$\mathbf{H}_\pi [l']_x s' \cong p'',$$

where $[l']_x$ denotes the skew-symmetric matrix of cross products, i.e., $l' \times s' = [l']_x s'$. Likewise, s'' is coincident with p'', then

$$s''^{\top}\mathbf{H}_\pi [l']_x s' = 0.$$

Denote $\mathbf{B}_\pi = \mathbf{H}_\pi [l']_x$. We show next that $\mathbf{B}_\pi$ is unique, i.e., independent of the choice of π. Let π_1, π_2 be two distinct planes of the pencil and let $\mathbf{H}_{\pi_1}, \mathbf{H}_{\pi_2}$ be their corresponding homography matrices. It is known that any two homography matrices between two fixed views satisfy,

$$\mathbf{H}_{\pi_2} \cong \lambda \mathbf{H}_{\pi_1} + e'' n^{\top}$$

where e'' is the projection of the optical center of Camera 1 onto the image plane of Camera 2 (the epipole), and n is a free vector. Because π_1, π_2 intersect at L, then, $\mathbf{H}_{\pi_1} u \cong \mathbf{H}_{\pi_2} u$ for all $u^{\top} l' = 0$, thus $n \cong l'$ and we have:

$$\mathbf{H}_{\pi_2} \cong \lambda \mathbf{H}_{\pi_1} + e'' l'^{\top},$$

and from which it clearly follows that $\mathbf{B}_{\pi_1} = \mathbf{B}_{\pi_2}$.

Let D be the intersection of a line $Q \in \mathcal{Q}$ with the plane π and denote its projections by d', d'' onto Views 1 and 2. The image line q' passes through the point d' and through the epipole e' and therefore: $q' \cong e' \times d'$. and similarly $q'' \cong e'' \times d''$. We can then write:

$$\begin{aligned} q''^{\top}\mathbf{B}q' &= (e'' \times d'')^{\top}\mathbf{H}[l']_x(e' \times d') \\ &= (e'' \times d'')^{\top}\mathbf{H}(d' \cdot l')e' - (e'' \times d'')^{\top}\mathbf{H}(l' \cdot e')d' \\ &= (d' \cdot l')(e'' \times d'')^{\top}e'' - (l' \cdot e')(e'' \times d'')^{\top}d'' \\ &= 0 \end{aligned} \tag{6.3}$$

where we used the identity:

$$a \times (b \times c) = (c \cdot a)b - (a \cdot b)c \tag{6.4}$$

and the fact that the homography $\mathbf{H}$ maps d' to d'' and e' to e''. □

Corollary 1 *The rank of the estimation matrix of the tensor from image measurements of lines across three views of a Linear Line Complex structure is at most 23.*

Proof: Let the tensor $\mathcal{T}_i^{jk}$ be sliced into three matrices $(\mathcal{T}_1^{jk}, \mathcal{T}_2^{jk}, \mathcal{T}_3^{jk})$, then the tensors $(\mathbf{B}, \mathbf{0}, \mathbf{0}), (\mathbf{0}, \mathbf{B}, \mathbf{0})$ and $(\mathbf{0}, \mathbf{0}, \mathbf{B})$ (and their linear combinations) span the tensors of the form:

$$\mathcal{T}_i^{jk} = \delta_i b^{jk}$$

where δ is a free vector of the family. Then,

$$s'_j s''_k p^i \mathcal{T}_i^{jk} = (p^i \delta_i)(s'_j s''_k b^{jk}) = 0.$$

Since $\delta_i b^{jk}$ does not include the general form of trilinear tensors (equation 6.1), the null space of the estimation matrix includes at least four distinct vectors: the true tensor describing the relative location of the three cameras, and the three 'ghost' tensors $(\mathbf{B}, \mathbf{0}, \mathbf{0}), (\mathbf{0}, \mathbf{B}, \mathbf{0})$ and $(\mathbf{0}, \mathbf{0}, \mathbf{B})$. Thus, the rank is at most $27 - 4 = 23$. □

The ambiguity can be further reduced by incorporating the tensor admissibility constraints (see Section 2.4.2) as detailed below.

Theorem 4 *The ambiguity of Tensor estimation from measurements coming from an LLC structure is* **at most** *an 8-fold ambiguity.*

Proof: We assume the correlation matrix slicing of the tensor into the three standard correlation matrices $(\mathcal{T}_1^{jk}, \mathcal{T}_2^{jk}, \mathcal{T}_3^{jk})$ (see Section 2.4.1). Let $\mathbf{W}$ be the $N \times 27$, $N \geq 27$, estimation matrix for linear estimation of the tensor, i.e., $\mathbf{W}v = 0$ where v is the tensor whose elements are spread as a 27 element vector, and v is spanned by the four-dimensional null space of $\mathbf{W}^\top \mathbf{W}$. Let v_1, v_2, v_3 be the three 'ghost' tensors corresponding to $(\mathbf{B}, \mathbf{0}, \mathbf{0}), (\mathbf{0}, \mathbf{B}, \mathbf{0})$ and $(\mathbf{0}, \mathbf{0}, \mathbf{B})$, respectively. Let v_0 be the (one dimensional) null space of

$$\mathbf{W}^\top \mathbf{W} - v_1 v_1^\top - v_2 v_2^\top - v_3 v_3^\top.$$

Since the null space span the admissible tensors, the three standard correlation matrices $(\mathbf{T}_1, \mathbf{T}_2, \mathbf{T}_3)$ of the admissible tensors are spanned by the tensors $v_0, ..., v_3$, i.e.,

$$\begin{aligned} \mathbf{T}_1 &= \hat{\mathbf{T}}_1 + \alpha_1 \mathbf{B} \\ \mathbf{T}_2 &= \hat{\mathbf{T}}_2 + \alpha_2 \mathbf{B} \\ \mathbf{T}_3 &= \hat{\mathbf{T}}_3 + \alpha_3 \mathbf{B} \end{aligned}$$

where $\hat{\mathbf{T}}_i$, $i = 1, 2, 3$, are the standard correlation matrices of the tensor v_0, and α_i are scalars. As part of the admissibility constraints (see Section 2.4.2), the standard correlation matrices $\mathbf{T}_i$ must be of rank 2, thus α_i are generalized eigenvalues of $\hat{\mathbf{T}}_i$ and $\mathbf{B}$, and since $\mathbf{B}$ is of rank 2, the characteristic equation for each α_i is of second order. Thus, we have at most 8 distinct solutions for $\mathbf{T}_i$. □

Empirical Observation 1 *Only one of the 8 solutions satisfies all the admissibility constraints.*

Explanation: the rank-2 constraint of the standard correlation matrices is closed under linear superposition (see Section 2.4.2). Numerical experiments show that only one out of the 8 possible solutions for the generalized eigenvalues α_1, α_2 and α_3 produces standard correlation matrices $\mathbf{T}_1, \mathbf{T}_2, \mathbf{T}_3$ whose linear superpositions produce rank-2 matrices. □

6.2.1 Algorithm for Recovering Structure and Motion in the Case of a LLC

1. Using robust estimation techniques determine the line correspondences which belong to the LLC and compute the matrix $\mathbf{B}$ (see Theorem 3). Here one might use a robust version of the 8 point algorithm [25].

2. From the matrix $\mathbf{B}$ create the 3 'ghost' tensors:

$$\begin{aligned} v_1 &= (\mathbf{B}, \mathbf{0}, \mathbf{0}) \\ v_2 &= (\mathbf{0}, \mathbf{B}, \mathbf{0}) \\ v_3 &= (\mathbf{0}, \mathbf{0}, \mathbf{B}) \end{aligned}$$

3. Using the point-line-line correspondences from the 3 views compute $\mathbf{W}$, the $N \times 27, N \geq 27$ estimation matrix for the linear estimation of the tensor.

4. Find v_0, the 4th vector spanning the (row) null space of $\mathbf{W}$ orthogonal to v_1, v_2 and v_3 by finding the null space of:

$$\mathbf{W}^\top \mathbf{W} - v_1 v_1^\top - v_2 v_2^\top - v_3 v_3^\top .$$

   In practice take the eigenvector corresponding to the smallest eigenvalue.

5. Find scalars α_i such that the vector:

$$v = v_0 + \sum_{i=1}^{3} \alpha_i v_i$$

   is an admissible tensor (see Theorem 4). This is done in two stages:

   (a) Let $\hat{\mathbf{T}}_i$, $\mathbf{T}_i$, $i = 1, 2, 3$, be the standard correlation matrices of the tensors v_0 and v respectively. Then:

$$\mathbf{T}_i = \hat{\mathbf{T}}_i + \alpha_i \mathbf{B}$$

   Enforce the constraint that $\mathbf{T}_i$ is of rank-2 to find α_i. Since the matrix $\mathbf{B}$ is of rank-2 this is quadratic constraint resulting in up to 2 solutions for each α_i for a total of $2^3 = 8$ solutions.

   (b) Prune the number of solutions down to one by enforcing the stronger admissibility constraint that any linear superposition of matrices $\mathbf{T}_i$ must be of rank-2. This is done by generating K random sets of linear coefficients δ_i such that $\sum \delta_i^2 = 1$ and computing the determinant of the linear superposition: $\sum \delta_i \mathbf{T}_i$ for each of the 8 possible solutions. The solution that consistently gives $\det(\delta_i \mathbf{T}_i) \simeq 0$ is the correct solution.

6.3 Experiments

6.3.1 The Experimental Procedure

Figure 6-3a, Figure 6-3b, and Figure 6-3c show the three input images used for the experiments. The scene is composed of two faces of a cube and another plane on the left which is parallel to the vertical edge of the cube. This is a schematic model of a typical urban scene with an edge of a building such as in Figure 6-1a.

Corresponding point features were manually extracted. The feature points were saddle points formed by the corners of two black squares which can be found with subpixel accuracy. The point features were grouped into four groups: Points from the left and right faces of the cube form one group each. Points on the planar surface were grouped into two vertical sets of features.

Line features were created by taking pairs of points. If no pair of points has members from more than one group (for example Figure 6-3) then we limit ourselves to a Linear Line Complex since all the 3D lines in the scene intersect the edge of the cube. By adding pairs that span two groups we can add lines that do not belong to the LLC. By judiciously choosing pairs we can add lines that are close or far from being part of the LLC (see Figure 6-3d). We can also choose pairs of points that define lines passing through the epipoles. This flexibility allows us to verify all the claims in Theorem 3.

Hardware Notes

The images were captured using a Pulnix TM9701 progressive scan camera with a $2/3_{inch}$ CCD and an 8.5_{mm} lens. The image resolution was $640 \times 480_{pixels}$.

To achieve the results presented here we had to take into account radial lens distortion. Only the first term of radial distortion was used. The radial distortion parameter, $K_1 = 6e-7$ was found using the method described in [73]. We note that that parameter value also minimized the error terms in equation (6.5).

6.3.2 Determining the LLC

The three input images (Figure 6-3a, Figure 6-3b, and Figure 6-3c) will be denoted Image 1, Image 2 and Image 3 respectively. We chose $N = 28$ pairs of points which defined lines all belonging to the LLC. These are overlaid as white lines in the figures. For each image pair (1,2), (2,3) and (1,3) we used the 8 point algorithm [25] applied to the line correspondences to compute the matrices $\mathbf{B}_{12}$, $\mathbf{B}_{23}$ and $\mathbf{B}_{13}$ that minimize:

$$E_{12} = \frac{1}{N}\sum_{i=1}^{N}(s_i \mathbf{B}_{12} s_i')^2 \tag{6.5}$$

$$E_{23} = \frac{1}{N}\sum_{i=1}^{N}(s_i' \mathbf{B}_{23} s_i'')^2$$

$$E_{13} = \frac{1}{N}\sum_{i=1}^{N}(s_i \mathbf{B}_{13} s_i'')^2$$

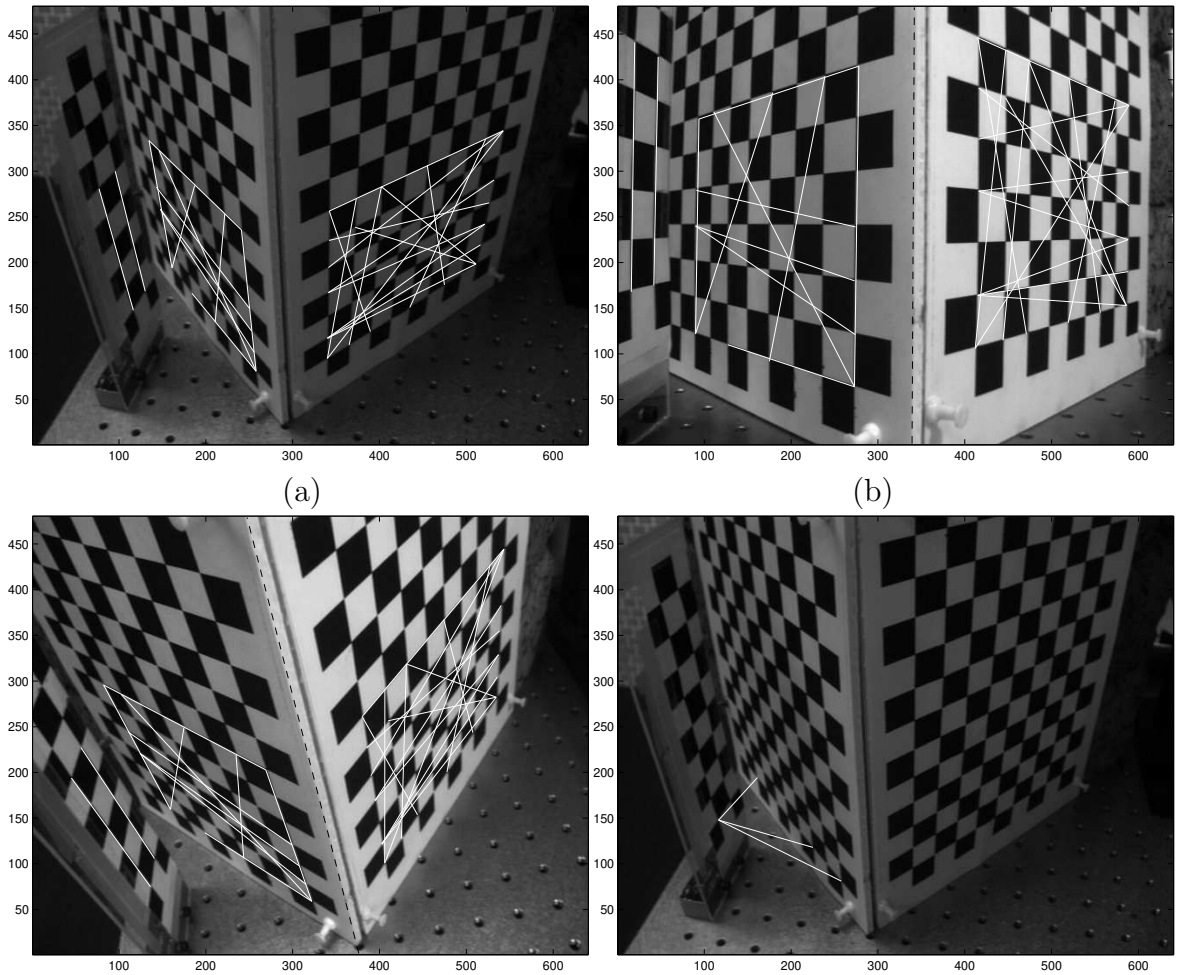


Figure 6-3: *The three input images used. All the lines marked are part of a linear line complex. They all intersect the line defined by the edge of the cube. Vertical lines intersect the edge at the point at infinity. The LLC was computed using images (b) and (c). The dashed lines in (b) and (c) are the projection of the common intersection line into the images. The results show it aligns very well with the edge of the true cube. (d) Three line which are not part of the LLC that are used in the experiments.*

Table 6.1: *Values of the error cost function for estimating the LLC when all the lines belong to the LLC (none) and when we add a line which passes close to, or far from, the common line of intersection (the edge of the cube).*

Extra Line	E_{23}	E_{12}	E_{13}
None	0.000055	0.00064	0.000079
Close	0.0019	0.0016	0.00054
Middle	0.0065	0.0038	0.0017
Far	0.0119	0.0919	0.0062

respectively. The coordinates of the lines s, s', s'' have been scaled as described in [25]. From Theorem 3, the left and right null spaces of $\mathbf{B}_{23}$ (for example) are the projections of the line L in Image 2 and Image 3. The dashed black line in Figures 6-3b, and 6-3c show the lines corresponding to the null spaces overlaid on the input images. They align well with the edge of the cube verifying the theory and showing that the matrix $\mathbf{B}$ can be recovered accurately. Similar results were found using the other image pairs.

Figure 6-3d shows Image 1 on which we have overlaid three lines not belonging to the LLC. Table 6.1 shows the error terms of equations (6.5) when all the line are from the LLC and when we add one of the lines shown in Figure 6-3d. When the extra line is far away from the common line of intersection the error is large. Even when the line nearly intersects the edge of the cube the error is still significant. Therefore, if most of the line come from an LLC, robust methods can be used to detect outliers. Other experiments, not reported here, use lines that pass through the epipole to verify the second half of Theorem 3.

6.3.3 Recovering Motion and Structure

We computed the motion tensor from the three views using four methods. First we used the linear method for a set of 131 point correspondences. Then we used the linear method for a set of 34 non degenerate line correspondences. Next we applied the linear method in a naive way to the set-of 28 line correspondences from an LLC. In other words we ignored the fact that the lines come from an LLC. Finally we estimated the tensor from the 28 degenerate lines using the algorithm described in Section 6.2.1.

Condition of the Estimation Matrix

Figure 6-4a (top) shows the four smallest singular values of the estimation matrix $\mathbf{W}$ used to compute the tensor from 34 non-degenerate lines. The smallest eigenvalue is considerably smaller than the others indicating that the null space of the matrix is of $rank = 1$ and the problem is well conditioned. Figure 6-4a (middle) shows the 5 smallest singular values for the estimation matrix computed from 28 lines belonging to an LLC. The 4 smallest singular values are about equal and are considerably smaller than the next smallest. This indicates that the null space of $\mathbf{W}$ is of $rank = 4$

as expected from Theorem 3. Simply taking the eigenvector corresponding to the smallest eigenvalue would be a mistake.

Figure 6-4b shows the projection of the vectors v_1, v_2 and v_3 on the eigenvectors of the estimation matrix $\mathbf{W}^\top\mathbf{W}$. The projections onto the eigenvectors corresponding to the first 23 eigenvalues are close to zero verifying that the vectors v_1, v_2 and v_3 are orthogonal to the first 23 eigenvectors. This verifies part of Theorem 4 which states that the vectors v_1, v_2 and v_3 are in the null space of $\mathbf{W}^\top\mathbf{W}$ (the remaining four eigenvectors).

Following the algorithm described in Section 6.2.1 we compute the eigenvalues and eigenvectors of the matrix

$$\mathbf{W}^\top\mathbf{W} - v_1 v_1^\top - v_2 v_2^\top - v_3 v_3^\top.$$

Figure 6-4a (bottom) shows the three smallest eigenvalues. The smallest eigenvalues is significantly smaller than the next smallest value indicating that the null space is now of $rank = 1$.

Reprojection of Lines Using the Tensor

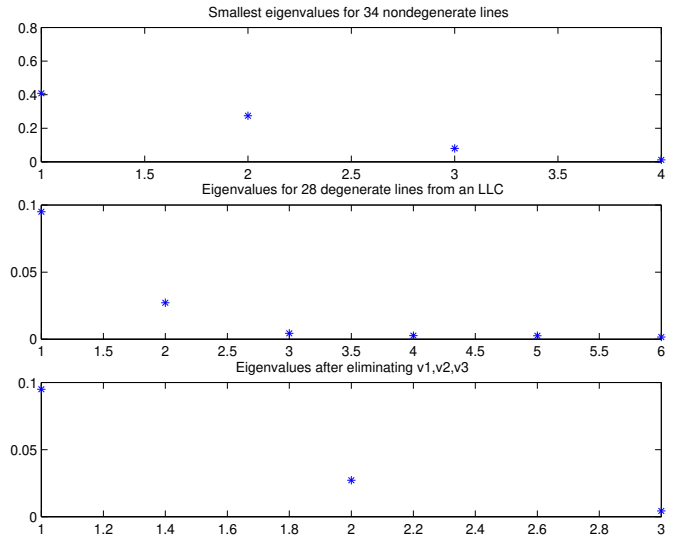
After recovering the tensor one can use the tensor to reproject a line given in two images into the third image. In order to test the tensor estimates we used ten additional lines shown in Figure 6-5. Three of the lines lie in the LLC on the left face of the cube. The other 7 do not lie on the LLC.

Figures 6-5b,c,d show the reprojection results (dashed lines) together with the original lines (solid lines) overlaid on Image 1. One can see that if the set of lines used to estimate the tensor all belong to an LLC then other lines in the LLC reproject more or less correctly, but the reprojection of lines not in the LLC is incorrect (Figure 6-5c). On the other hand, reprojection using the tensor computed by taking into account the degeneracy, (Figure 6-5d) gives results as good as if we had a non-degenerate set of lines to estimate the tensor (Figure 6-5b).

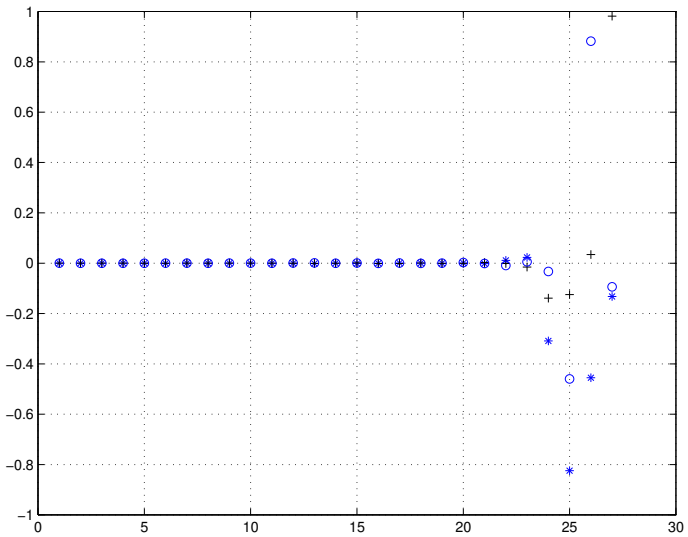
6.4 Summary

We have shown that linear methods for estimating motion and 3D structure from lines lead to a degenerate set of equations in the case of a Linear Line Complex. The LLC, a configuration of lines that all intersect a common line in $\mathcal{P}^3$, is in fact a common configuration of lines occuring frequently in man-made environments. This degeneracy is due to a bilinear constraint on lines in two views. The constraint equation has a form similar to the epipolar constraint but where lines replace points and the epipoles are replaced by the image of the common line of intersection in the two views. This constraint can be used to determine whether a set of lines belongs to an LLC and enables us to reject a outliers using least-median-of-squares or other robust estimation methods.

An LLC, in general, is not degenerate for non-linear methods. The theoretical analysis leads to a modification of the linear methods that can recover the structure



(a)



(b)

Figure 6-4: *(a) The smallest singular values of the estimation matrix* $\mathbf{W}$*, for a degenerate and non degenerate set of lines. (b) With a degenerate set of lines, the vectors* v_1*,* v_2 *and* v_3 *were projected onto the 27 eigenvectors of* $\mathbf{W}^\top\mathbf{W}$*. The values for the first 23 eigenvectors are close to zero verifying that the vectors* v_1*,* v_2 *and* v_3 *are orthogonal to the first 23 eigenvectors and are therefore in the null space of* $\mathbf{W}^\top\mathbf{W}$*. (See Theorem 2.)*

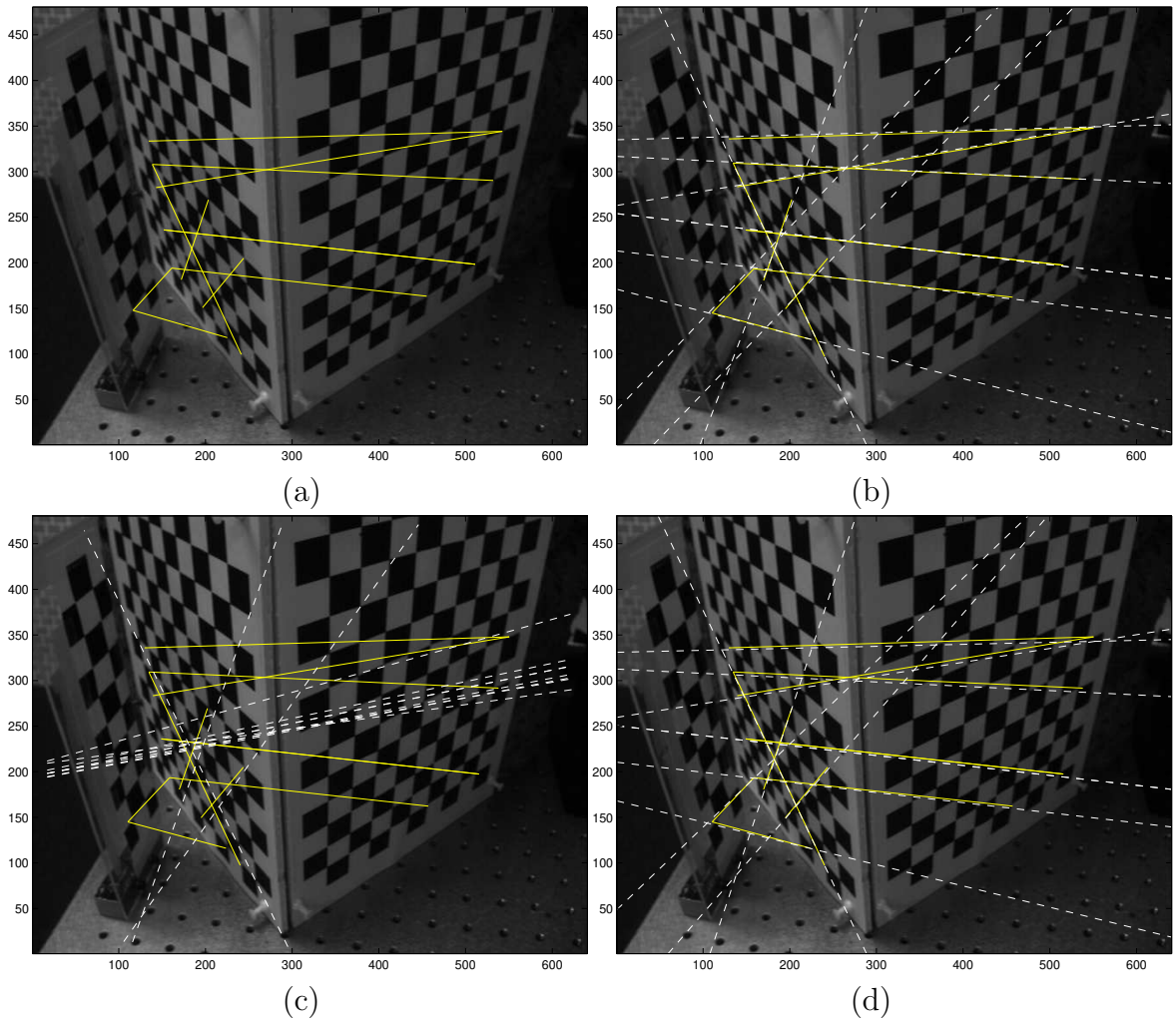


Figure 6-5: *(a) Ten additional lines are used to test the recovered tensors. Three of the lines lie on the left face of the cube and are therefore the part of the LLC. The other 7 are not. The test lines from Image 2 and Image 3 are reprojected back into Image 1 using the recovered tensors. Solid lines are the true locations. Dashed lines are the reprojected lines. (b) The tensor was computed using a set of 34 non degenerate lines. (c) The tensor was computed using a degenerate set of 28 lines using the standard method. Since it does not take into account the degeneracy the tensor fails to correctly reproject the lines. The only lines in (c) that reproject correctly are those that belong to the LLC. (d) Using the new algorithm for the degenerate case the computed tensor correctly reprojects all the lines.*

and motion in the LLC case. We have proven that the motion can be recovered up to 8 discrete solutions. Empirical evidence shows that the number of solutions can be reduced further to a single solution one.

We have implemented the algorithm, and experiments with real images verify the theoretical analysis. Although the results using the modified linear algorithm compare favorably with the results obtained using a non-degenerate set of lines, the system at this point is not robust. For example, it requires that lens distortion be taken into account. Further engineering would be involved in making a practical system.

Chapter 7

Lens Distortion Calibration Using Point Correspondences

7.1 Introduction

Radial lens distortion can be a significant factor in medium to wide angle lenses. The errors can be 10-100 pixels at the edges of the image [71]. These are typically the lenses used when performing image based 3D reconstruction of large objects or in a confined space. These are also the type of lenses which have been used in the experiments throughout this thesis. This chapter describes a new method for lens distortion calibration using point correspondences in multiple views and using the geometric constraints described in Section 2. There is no need to know either the 3D location of the points or the camera locations. In fact one can use the same point correspondence data which is to be used for subsequent vision tasks such as 3D reconstruction. The method is therefore suitable as an online preprocessing stage.

The simulation results in Section 7.5 show that the geometric configuration of the cameras is important for the success of the calibration. In summary, the cameras must be far apart with large angles between the camera views (i.e. large vergence angles). This result has practical implications for the estimation of structure and motion from image sequences (Chapter 3 and Chapter 5). In such applications the camera motion is typically small. On the one hand, this means that the radial distortion parameters cannot be estimated from the input images, and on the other hand, we find that the radial distortion does not affect the convergence. It only produces a nonlinear distortion in the final motion and structure estimates.

There are two variants of the new calibration method. The first uses two images and the second uses three images. Both methods will be presented but the major focus of this chapter (and the experimental results) will be on the three image method.

7.1.1 The Two Image Method

Given a point in one view we know that the corresponding point in the other view must line along an epipolar line. Given at least 8 point correspondences we can find the epipoles and epipolar lines in a linear manner [26]. With more than 8 points we can find the least squares solution.

The epipolar constraint holds for ideal pinhole cameras. Due to noise in the

feature detection and due to lens distortion we get some error. With more than the minimum 8 points we can define a cost function as the RMS of the distances from the points to the corresponding epipolar lines. We then find lens distortion parameters that correct the image coordinates of the feature points to minimize this error. The error function is in general well behaved, and the distortion parameters can easily be found by nonlinear search techniques.

A variant of this method was proposed by Zhang [90] using two cameras from a stereo pair. He found that it was difficult to find the lens distortion parameters unless the distortion was severe or the points were found very accurately. The apparent contradiction between his results and ours can be resolved if we observe that in a typical stereo camera configuration the angle between the two views is very small. As we have noted, in order to get a distinct minimum in the error function, the angle between views must be large.

7.1.2 The Three Image Method

Corresponding points in three images are related by 4 independent trilinear equations. [57]. These equations have 27 parameters which can be found in a linear manner given at least 7 point correspondences. These parameters allow us to reproject corresponding points given in two of the images into the third image.

Due to noise and lens distortion, the reprojection is not perfect. We define our cost function as the RMS reprojection error and find lens distortion parameters that minimize it.

7.2 Related Work

Known world coordinates: The classic method for lens distortion calibration is the bundle adjustment method [66]. It uses one or more views of a calibration object with known 3D coordinates (control points). Using iterative methods it then finds both external parameters (position and orientation) and internal camera parameters. The internal camera parameters include the parameters of the pinhole camera model (principal point, principal distance) and the parameters of lens distortion. See also Weng et al. [86].

Projective constraints: Under perspective projection, straight lines in space project to straight lines in the image. With real lenses the lines appear instead to be slightly to moderately curved. By searching for lens distortion parameters which straighten the lines the *Plumb Line* method and its derivatives [8] [20] [71] [12] find the lens distortion without needing to find the external parameters or the other internal camera parameters. One or more images can be used.

Unknown world coordinates: Stein [71], and Du and Brady [13] use corresponding points or edges in images where the camera has undergone pure rotation to find the internal camera parameters including lens distortion. The 3D location of the points is not required.

7.3 Mathematical Background

We will review here the basic relevant material from Chapter 2. As in the preceding chapters a point in 3D projective space ($\mathcal{P}^3$) will be represented by P_i where the subscript i denotes the i'th point. The subscript i might often be dropped for clarity. Written in homogeneous coordinates: $P = (X, Y, Z, T)$. The point P_i projects onto the image plane of the j'th camera at point p_{ij} where $p = (x, y, 1)$.

The 4×3 camera projection matrix for the j'th camera will be denoted $\tilde{\mathbf{A}}_j$. Using homogeneous coordinates the perspective projection can be written as:

$$\rho_{ij} p_{ij} = \tilde{\mathbf{A}}_j P_i \tag{7.1}$$

for points $i = 1...n$ in images $j = 1...k$. ρ_{ij} is an unknown scale factor which is different for each point and each image.

7.3.1 The Epipolar Constraint

As shown in Section 2.3, For every pair of cameras (j,k) there exists a 3×3 *fundamental matrix* ($\mathbf{F}_{jk}$) such that:

$$p_{i,j}^T \mathbf{F}_{jk} p_{i,k} = 0 \tag{7.2}$$

The above equation is the epipolar constraint. Given a particular camera geometry, embodied in the matrix $\mathbf{F}$, each point in one image defines a line v on which the corresponding point in the second image must lie. The equation of that line is given by:

$$v^T \mathbf{F}_{jk} p_{i,k} = 0.$$

Given 8 or more point correspondences, the fundamental matrix ($\mathbf{F}$), can be determined up to a scale factor using the *eight point algorithm* which is described in [26] together with many important implementation details.

7.3.2 The Trilinear Tensor Constraint

In Section 2.4 we derive the trilinear equations of [57]. Given a set of 3D points there exists a set of trilinear equations between the projections of those points into any three perspective views. In total there exist 9 such equations for each point with at most 4 being independent.

Four of the nine equations are as follows (equation 2.42):

$$\begin{aligned}
&x'' \mathcal{T}_i^{13} p^i - x'' x' \mathcal{T}_i^{33} p^i + x' \mathcal{T}_i^{31} p^i - \mathcal{T}_i^{11} p^i = 0, \\
&y'' \mathcal{T}_i^{13} p^i - y'' x' \mathcal{T}_i^{33} p^i + x' \mathcal{T}_i^{32} p^i - \mathcal{T}_i^{12} p^i = 0, \\
&x'' \mathcal{T}_i^{23} p^i - x'' y' \mathcal{T}_i^{33} p^i + y' \mathcal{T}_i^{31} p^i - \mathcal{T}_i^{21} p^i = 0, \\
&y'' \mathcal{T}_i^{23} p^i - y'' y' \mathcal{T}_i^{33} p^i + y' \mathcal{T}_i^{32} p^i - \mathcal{T}_i^{22} p^i = 0.
\end{aligned} \tag{7.3}$$

where x', y' and x'', y'' are the image coordinates in the first and second images respectively. p is the image point in the third view. $\mathcal{T}_i^{jk}$ are the 27 coefficients arranged

in a $3 \times 3 \times 3$ tensor. Seven point correspondences give 28 equations which are enough to recover the coefficients up to a scale factor.

Given 7 points in 3 images one can recover the tensor. Given the tensor and the locations of additional corresponding points in two of the images, one can calculate the locations of those points in the third image.

7.3.3 The Lens Distortion Model

These projective constraints assume a perfect pinhole model. This is a good model for long focal lengths, but medium to wide angle lenses have noticeable lens distortion. The standard model for lens distortion [66] is a mapping from the distorted image coordinates, (x_d, y_d), that are observable, to the undistorted image plane coordinates, (x_u, y_u), which are not physically measurable using the equation:

$$\begin{aligned} x_u &= x_d + x'_d(K_1 r_d^{'2} + K_2 r_d^{'4} + \cdots) \\ y_u &= y_d + y'_d(K_1 r_d^{'2} + K_2 r_d^{'4} + \cdots) \end{aligned} \tag{7.4}$$

where K_1 and K_2 are the first and second parameters of radial distortion and:

$$r_d^{'2} = x_d^{'2} + y_d^{'2} = (x_d^2 - c_{xr})^2 + (y_d^2 - c_{yr})^2 \tag{7.5}$$

It has been shown in [71] that allowing the center of radial distortion, (c_{xr}, c_{yr}) to be different from the principal point is a good approximation to adding the terms for decentering distortion as given in [66].

7.4 The Algorithm

The step by step algorithm is as follows:

1. Find point correspondences between 3 views.
2. Make an initial guess of the distortion parameters: an appropriate guess for (C_x, C_y) is the center of the image. Choose K_1 such that $K_1 \times r^3 \leq 0.1$ where r is the distance to the corner of the image.
3. Using the distortion parameters compute the undistorted feature locations (equation 7.4).
4. Compute the trilinear tensor. Then use the tensor to compute the reprojection error for each point (equation 7.3).
5. Adjust the distortion parameters to reduce the reprojection error. This is done using the Levenberg-Marquart algorithm (see Section 7.6.1).
6. Loop back to step 3 till convergence.

Comments: It is best to search initially for only one parameter of radial distortion (K_1). After finding the best K_1, one can use that value as a starting guess and try searching for other parameters as well. One can search for different values of K_1 for each camera. If one gets a much smaller RMS reprojection error than when enforcing the same value of K_1 for all the cameras then it is a good bet that the cameras are in fact different.

7.5 Simulation

Simulations were performed (using Matlab) to test the effect of camera configurations and noise on the shape of the cost function. Eighty 3D points were generated, uniformly distributed between -1 and 1 in the X,Y and Z directions. These points were projected using a projective camera model with optical axis aligned with the Z direction. The camera was located at $Z_0 = -4$. This simulates a camera with a viewing angle of about 55^o. The focal length was $f = 1000 pixels$.

The 3D points were rotated prior to projecting which is equivalent to the camera rotating around the center of the points at a given radius Z_0. Normally distributed random noise was added to the image coordinates which were then distorted using (equation 7.4) with only one radial distortion parameter, $K_1 = 2.5 \times 10^{-7}$, the value found in Section 7.6.

Consider the following 3 camera setup (Figure 7-1). Cameras 1 and 2 are rotated θ^o and $-\theta^o$ around the Y axis respectively. Camera 3 is rotated around the X axis with varying angles $\alpha = 0, 10^o, 20^o \ldots 80^o$. The radial distortion is varied from 0 to $2 \times K_1$ in $0.1 \times K_1$ increments and the reprojection error from camera images 1 and 2 to camera image 3 is calculated.

Due to lack of space we show only the simulation results for $\theta = 20^o$ and additive noise $\sigma = 0.3$ *pixel* (Figure 7-2). Note that for small α the error as a function of K_1 is very shallow and it would be hard to find the correct K_1 value. With larger noise values the minimum is sometimes outside the plotted range for $\alpha = 0$. Other simulation experiments show that increasing θ towards 40^o the curves become steeper and reducing θ makes them shallower. At $\theta = 2^o$ the results are unreliable even for large α values.

7.6 Experiments with Real Images

7.6.1 Experimental Details

The Images, Features and Feature Detector

Figure 7-3a, Figure 7-3b and Figure 7-3c show a typical image triplet used in the experiment (same camera). The camera motion included a large, unknown, degree of rotation and translation. Figure 7-3d was taken from approximately the same angle as Figure 7-3c but the camera was twice as far away and the focal length longer (zoom).

Three lines of feature points were marked on a flat sheet of metal. The features were two small triangles touching at a point. These points are saddle points in the gray level image and result in local minima in the determinant of the Hessian of the image. The point is located to subpixel accuracy by locally fitting the determinant of the Hessian image to a paraboloid and finding the minimum analytically.

Correspondence was performed manually. In total 75 feature points were detected: 7 points on the top surface of the metal cube, a total of 10 points on the other metal objects and 55 coplanar points in 3 lines plus one nearly coplanar point.

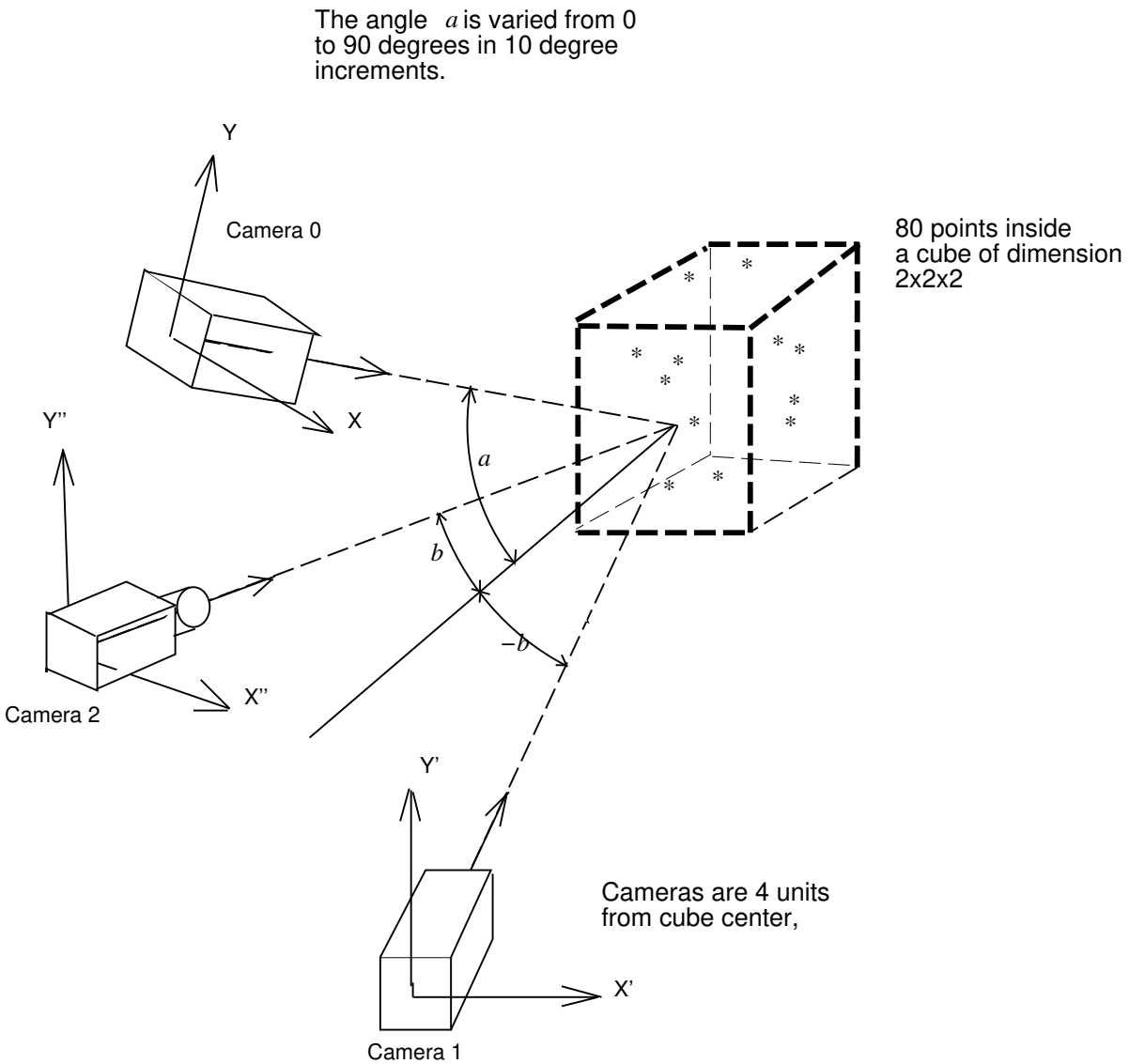


Figure 7-1: *Simulated camera setup.*

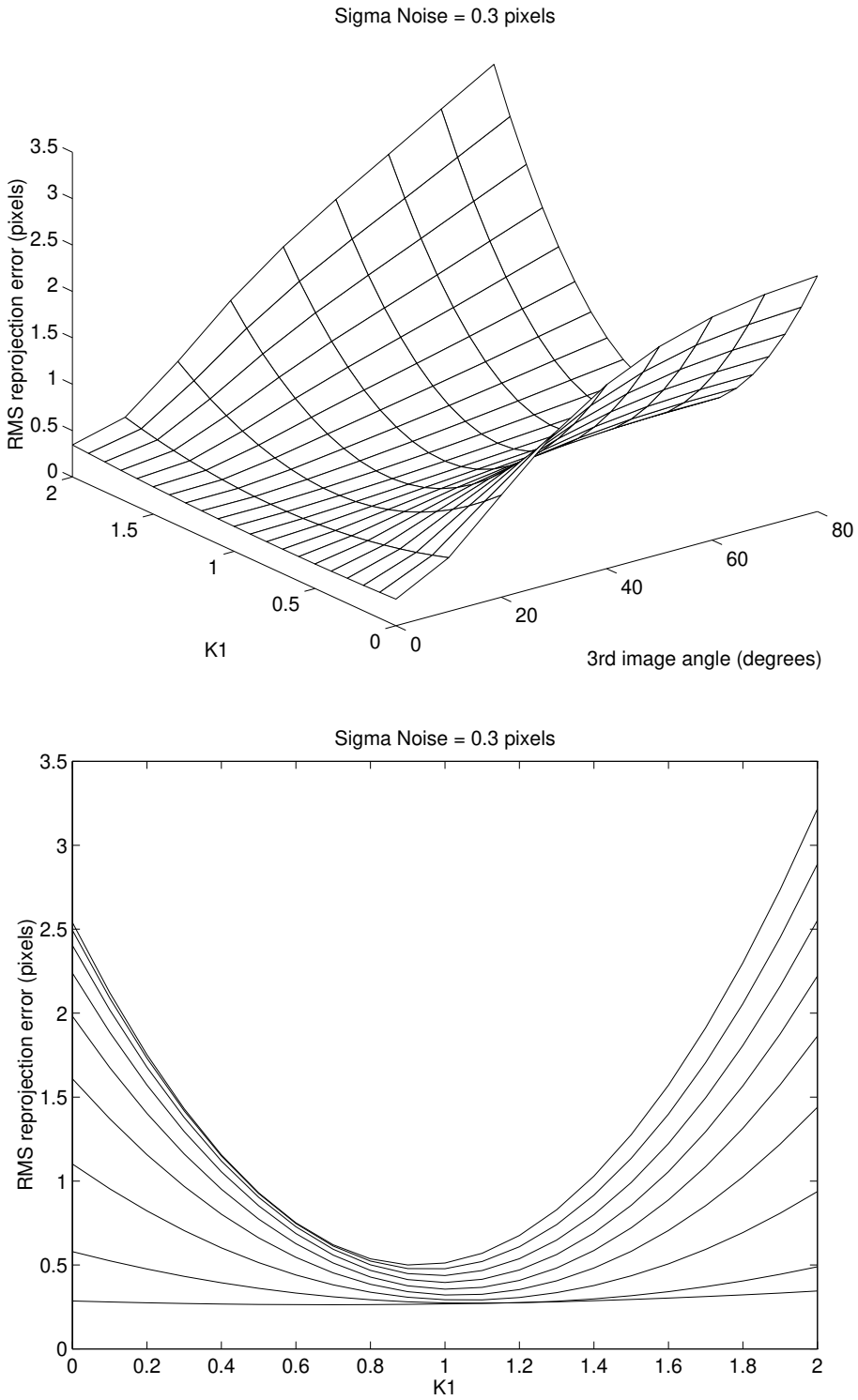
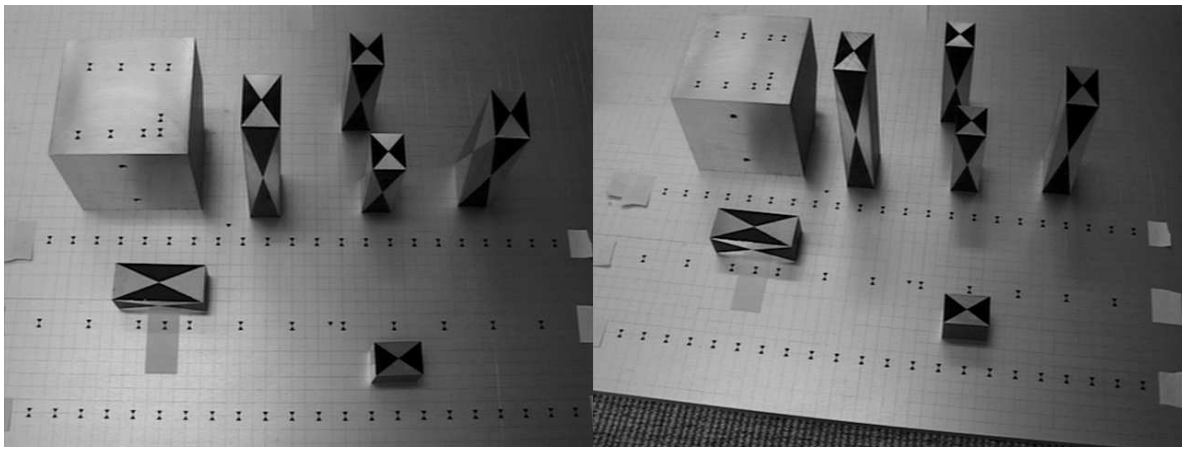
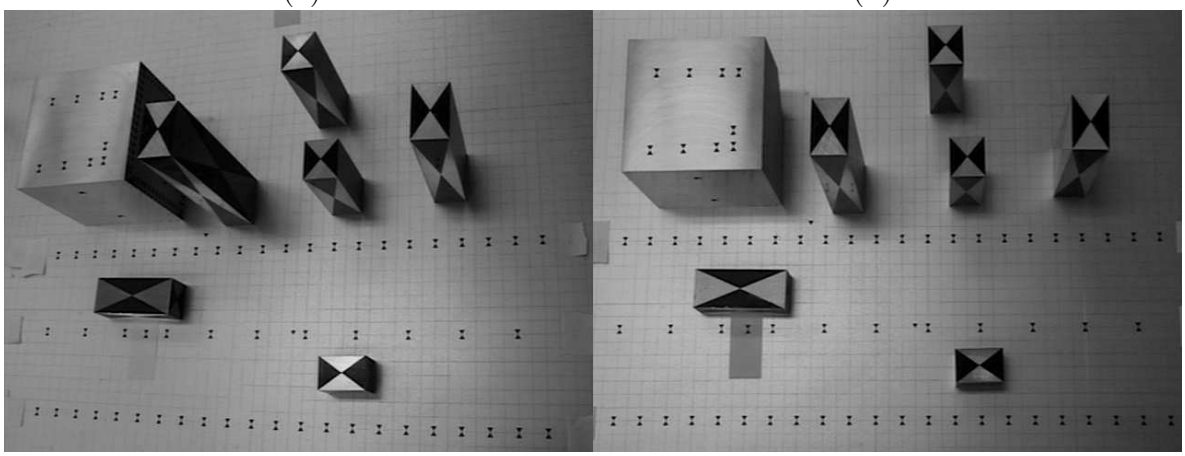


Figure 7-2: *Simulation results. Reprojection error while varying the radial distortion correction from 0 to 2 times the correct value and varying the viewing angle of the 3rd camera. Gaussian noise of sigma=0.3 was added to feature coordinates.*



(a) (b)



(c) (d)

Figure 7-3: *(a,b,c) The images used for calibration. (d) Alternative third image taken from further away with a longer focal length. Note that in this case the lines are not curved.*

Hardware and Software

The camera was a SONY DCR-VX1000, a high quality digital handycam. The lens was the built-in zoom lens open to the widest angle giving a corner to corner viewing angle of approximately 55 degrees. Image capture and processing was performed on an SGI Indy workstation.

The camera parameters were found using a nonlinear optimization program based on the subroutine LMDIF from the software package MINPACK-1 [47]. This subroutine uses a modified Levenberg-Marquart algorithm.

7.6.2 Experiment 1: Finding the Lens Distortion Parameters

Using all 75 points in the 3 images (Figure 7-3a, Figure 7-3b, and Figure 7-3c) we computed the trilinear tensor (see Section 7.3) and then the reprojection error from Image 2 (Figure 7-3b) and Image 3 (Figure 7-3c) to Image 1 (Figure 7-3a). No robust estimation techniques were used. I will denote the radial distortion parameters found as K_{1best} and K_{2best}.

Convergence Range

One must supply the non-linear search routine with an initial guess. Many initial guesses were tried for the radial distortion parameter (K_1). The second parameter (K_2) was held at zero and the center of distortion at the image center (320,240). The non-linear search would converge to a value $K_{1best} = 2.58 \times 10^{-7}$ with a starting guess anywhere in the range $(2.5 \times 10^{-2} : 2.5 \times 10^{-12})$. (i.e. through at least 4 orders of magnitude in either direction). Convergence took between 3 and 10 iterations.

Assuming two radial distortion parameters (K_1, K_2), the non-linear search would converge to a value $K_{1best} = 4.755 \times 10^{-7} \pm 0.1\%$ and $K_{2best} = -1.153 \times 10^{-12} \pm 0.5\%$ with a starting guess anywhere in the range $K_1 = (2.5 \times 10^{-5} : 2.5 \times 10^{-9})$ and $K_2 = (-1 \times 10^{-7} : 1 \times 10^{-7})$. (i.e. through at least 2 orders of magnitude in either direction of K_1). Convergence took between 3 and 10 iterations.

The Shape of the Cost Function

One of the dangers of using non-linear methods is the presence of local minima. Figure 7-4a shows the cost function evaluated over a range of values of K_1 from 0 to $2 \times K_{1best}$ (twice the final estimated value) in $0.1 \times K_{1best}$ increments. The function is smooth, qualitatively parabolic in shape and with no local minima. These characteristics have repeatedly been seen with no exception in many similar experiments with varying ranges and resolutions.

Assuming Different Parameters for Each Camera

The calibration method can work for different cameras. We can search for 3 separate sets of calibration parameters, one set for each camera. Allowing for one radial distortion parameter for each camera we get:

Table 7.1: Radial distortion parameter K_1 obtained using various ordering of the images

Order	K_1	RMS error
a,b,c	2.58162e-07	e=6.773012e-02
a,c,b	2.5811e-07	e=6.773012e-02
c,a,b	2.4687e-07	e=7.554849e-02
c,b,a	2.46804e-07	e=7.554849e-02
b,a,c	3.19903e-07	e=6.590745e-02
b,c,a	3.19834e-07	e=6.590745e-02
mean	2.7498e-07	
std	3.9313e-08	
std/mean	0.14	

$K_{1best} = (2.55 \times 10^{-7}, 2.69 \times 10^{-7}, 2.19 \times 10^{-7},)$
The 3 images were taken by the same camera so the 3 values for K_{1best} should come out the same. In fact, they are within 20% of each other. Better results might be achieved with a larger number of points.

The third image (Fig: 7-3c) was then replaced with an image taken from further away using a longer focal length (Figure 7-3d). In other words it is a 'different camera'. Allowing for one parameter for radial distortion for each camera the results were:

$K_{1best} = (2.15 \times 10^{-7}, 2.20 \times 10^{-7}, -5.93 \times 10^{-8})$
The third camera appears to differ significantly and has much less distortion as we would expect from the longer focal length.

Which Image is Used for Reprojection

The method has an asymmetry in that two of the images are used to reproject into the third image. Thus, the third image is treated differently. All the possible permutations of the 3 images were tried. The results are summarized in Table 7.1. The standard deviation in the value of K_{1best} was 14% which is not large. Nevertheless, to avoid the ambiguity one can use as a cost function the sum of reprojection errors in the 3 distinct possibilities.

7.6.3 Experiment 2: Testing the Radial Distortion Parameters in the Projective Domain

We have no ground truth about the radial distortion parameters. In order to test the radial distortion parameters, a special set of feature points was used. 44 of the points are coplanar. They are arranged in 2 straight lines of 22 equally spaced points each.

Collinearity of Points

Using one of the images (Figure 7-3a), for each of the two sets of 22 points the image coordinates were fit with a best straight line. The RMS distance of the points from

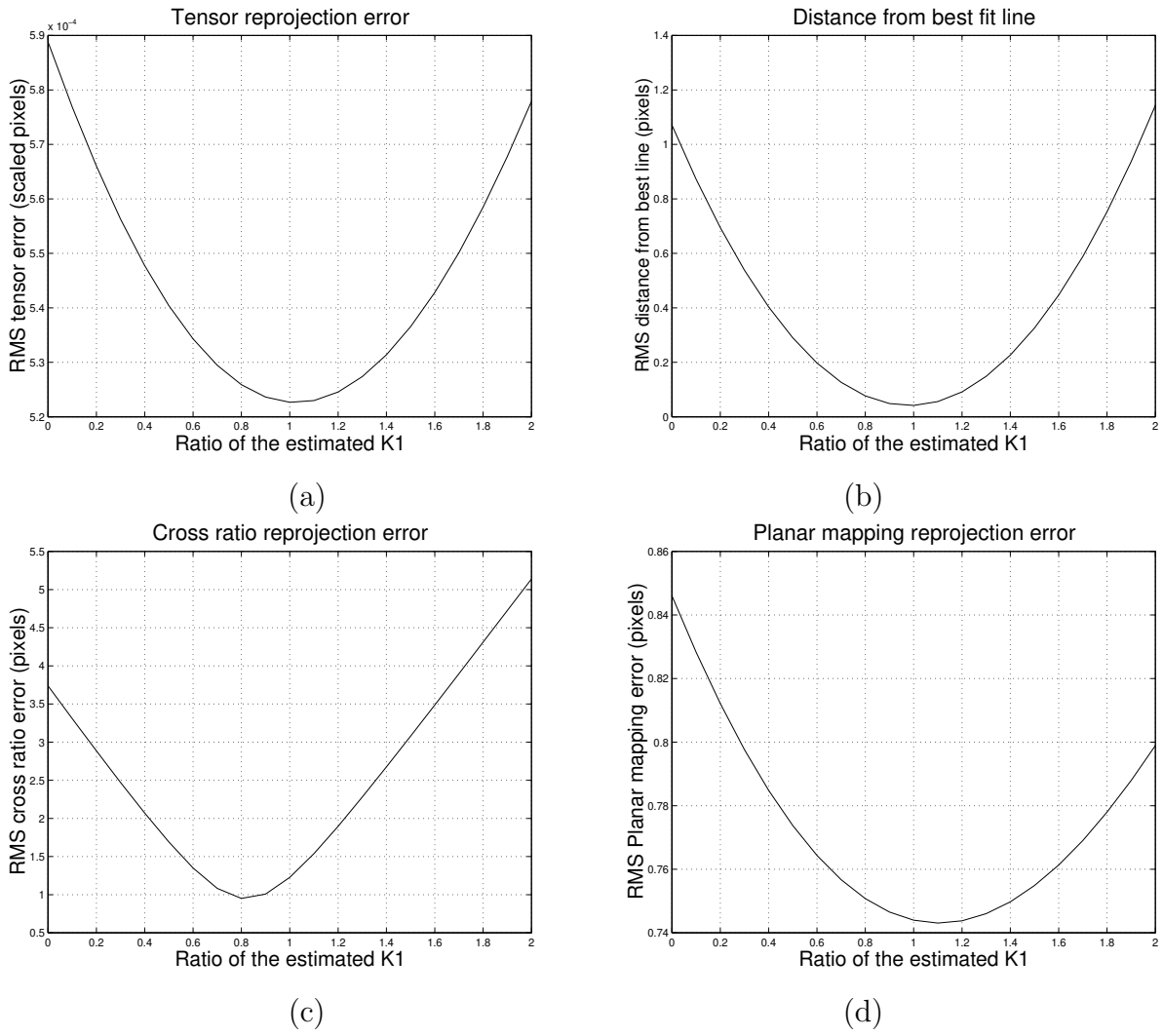


Figure 7-4: *(a) The tensor reprojection error as a function of the value of the radial distortion K_1. (b,c,d) The error in other projective measures as a function of the value of the radial distortion K_1.*

the corresponding lines was then computed. This is the measure used in the *plumb line method* described in Section 7.2.

Figure 7-4b shows the average distance as a function of the radial distortion parameter K_1, where K_1 varies from 0.0 to $2.0 \times K_{1best}$ in $0.1 \times K_{1best}$ steps. The function's minimum is at $K_1 = K_{1best}$. The average distance of points from the line drops from 1.15 pixels to 0.05 pixels after correction.

Cross Ratio of Points

The cross ratio of four points along a line is a projective invariant. Taking the 1st, 2nd and 22nd points along each of the lines, one can compute the cross ratio using

any of the other points:

$$r_i = \frac{\frac{\alpha_1 - \alpha_2}{\alpha_1 - \alpha_i}}{\frac{\alpha_{22} - \alpha_2}{\alpha_{22} - \alpha_i}} \tag{7.6}$$

One can invert equation (7.6) to obtain:

$$\alpha_i = \frac{r_i \alpha_1 (\alpha_{22} - \alpha_2) - r_i \alpha_{22} (\alpha_1 - \alpha_2)}{r_i (\alpha_{22} - \alpha_2) - r_i (\alpha_1 - \alpha_2)} \tag{7.7}$$

Given that we know that the points along each line are equally spaced we can use equation (7.7) to estimate the location of points 3 through 21 given the location of points 1, 2 and 22. The errors are due to feature detection error and also radial distortion. Figure 7-4c shows the RMS error in the estimation of all the 38 points as a function of K_1. The minimum error is obtained at a value of K_1 close ($0.8 \times K_{1best}$) to that obtained using the new calibration method.

Planar Mapping of Points Between Two Images

Given two sets of image points obtained from two views of a planar object one can compute a transformation from one set of points to the other which takes the form:

$$p' = \mathbf{A}p \tag{7.8}$$

Where p and p' are the corresponding image points in the first and second images and $\mathbf{A}$ is a 3×3 homography matrix. Given more than 4 points one can compute $\mathbf{A}$ in a least squares manner. One can then use the matrix $\mathbf{A}$ to map all the planar points from one image to the second. Figure 7-4d shows the RMS reprojection error using the images in Figure 7-3a and Figure 7-3b. The minimum error is obtained near the value of K_1 found using the distortion calibration.

7.6.4 Experiment 3: Euclidean Reconstruction

Projective reconstruction from the three images was performed according to [61]. Transformation to Euclidean 3D coordinates then required 5 control points. Three points were chosen from the planar surface and two other points were chosen such that the five were in *general position*, no 4 points coplanar. The world coordinate system was chosen such that the planar surface was the XY plane.

The reconstruction was performed twice, once with the original feature coordinates and once with the corrected coordinates using $K_1 = 2.58 \times 10^{-7}$. The best plane was found using the points which came from the planar surface. Figure 7-5 shows the height of the points from the best plane for the three lines. In Figure 7-5c one can see that for the line close to the edge of the image the line is distinctly curved in depth and that correcting for radial distortion removes the curvature. The curve along the whole length of the line is about $0.3cm$ which is about 1% of the total length of the line ($42cm$). The curvature of the lines closer to the center of the image is smaller.

Point number 45 in Figure 7-5b is the point that was drawn on the piece of paper and was in fact not coplanar with the other points. In the case of the corrected points reconstruction (dot dash line) it is clearly stands apart.

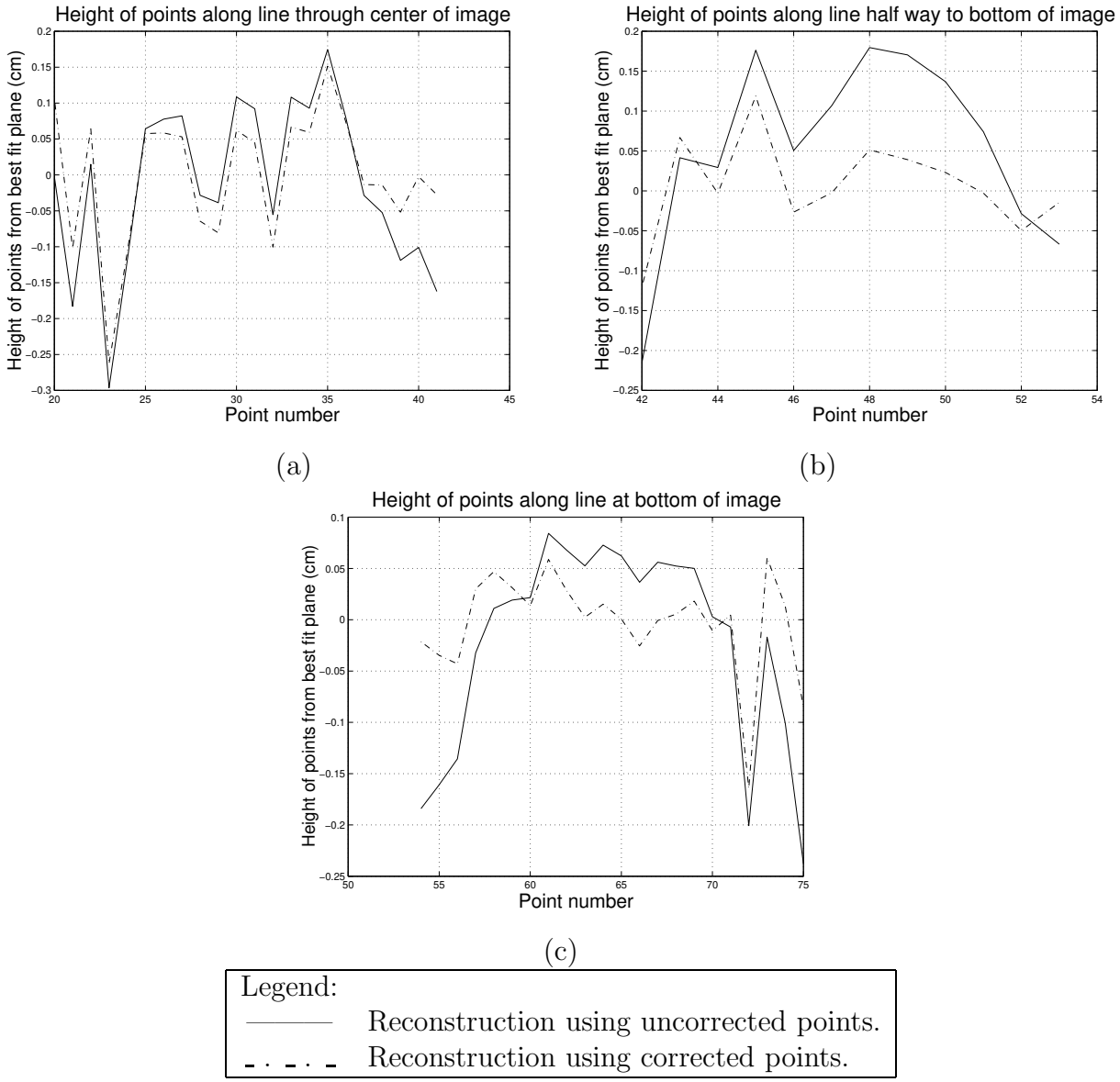


Figure 7-5: *The height of the reconstructed 3D points from the best fit plane. Figures (a), (b) and (c) are the points along the lines which pass close to the center of the image, half way down and close to the bottom, respectively.*

7.7 Discussion

A new method for determining the lens distortion has been described. Although the method is iterative it has been shown that the method converges fast and that the range of convergence is very wide. The resulting distortion parameters are reasonable and have been shown to agree with those determined by other methods.

Since this method uses only point correspondences in multiple views it fits very naturally inside feature based 3D reconstruction systems. It has been shown here that in the case of a 55^o lens, when the distortion is hardly noticeable in the image, radial distortion can cause a small but noticeable error in reconstruction results. Since the computational overhead of this method is small it makes sense to incorporate this method in all 3D reconstruction systems which use medium and wide angle lenses and where accuracy is desired. Especially, since most of the computational machinery required for tasks such as finding the trilinear tensor or fundamental matrix probably already exists in the overall system.

The Manual of Photogrammetry [66] warns us that "the strong coupling that exists between interior elements of orientation [principal point, focal length] and exterior elements can be expected to result in unacceptably large variances for these particular projective parameters when recovered on a frame-by-frame basis". This is also applicable to lens distortion parameters and it was seen in Section 7.6.2 and Section 7.6.2. Note that the interior and exterior parameters are determined implicitly in the fundamental matrix or the trilinear tensor. To determine the actual lens distortion parameters one can use many images from a variety of locations and apply this method to many triplets of those images. The computational complexity of the method is linear in the number of triplets and number of feature points.

Chapter 8

Summary and Future Work

We have presented a new method for recovering motion and depth from images sequences. The method we presented, the *model-based brightness constraints*, combines photometric constraints in the form of the optical flow constraint equation, together with geometric constraints between corresponding points and lines in three views. Thus, following the 'direct methods' approach of Horn & Weldon [34], we avoid the need to use feature correspondences .

Results from experiments with real images demonstrate that the method can accurately recover the motion parameters even when the motion includes both translation and rotation, and the method can accurately recover true Euclidean depth if the internal camera parameters are known. The reconstruction results degrade gracefully if there are errors in the internal camera parameters.

We can recover the motions reliably if the two translations are not collinear. We have also shown that the method, as presented in this thesis, does not work in the case of collinear motion unless we can assume pure translation. Of course, this does not constitute a proof that direct estimation of motion from monocular sequences cannot be done in the collinear case. We have presented some success in motion estimation from collinear sequences in the case of pure translation but it is not clear how to extend this to the rotation case. The problem seems to lie in the first order approximation of the optical flow constraint equation.

The lack of a solution to the collinear motion limits some of the potential applications (for example, movie compression). This technique can still be used for site modeling from video streams, but care must be taken when capturing the image data. Alternatively, an active vision head could be used, and controlled to make sure that the camera translated in two distinct directions. The motion control of the active vision head need not be precise, as we have shown that the motion can be accurately recovered from the images. In general, collinear motion is not ideal for 3D site reconstruction since there will potentially be many lines in the image which lie on, or close to, epipolar lines, and cannot provide depth information. Using two distinct motion directions avoids this problem.

Monocular image sequences have other drawbacks for 3D site reconstruction which are independent on the method used for estimating the motion and depth. There is the scale ambiguity. This makes it hard to align 3D reconstructions from different parts of the sequence since the magnitude of translation might have changed and so the 3D reconstructions will be at different scales. If the actual translation is close to zero,

as often happens with hand held cameras then the method becomes unstable. These drawbacks can be overcome by using stereo image sequences. We have presented a new method for recovering camera motion and 3D shape from a moving stereo rig. Using this method it is simple to bring 3D reconstructions from various view points into the same coordinate system.

8.1 Future Work

Affine Camera Models

It would be interesting to try and extend this framework to orthographic, weak perspective and affine camera models. Recently, Quan & Kanade [54] have shown that for affine cameras there exists a constraint between the *directions* of corresponding lines in three views which takes the form of a $2 \times 2 \times 2$ trilinear tensor. It is not clear how to combine this geometric constraint with the optical flow constraint equation which provides point-line correspondences.

Reconstruction Under Changing Lighting Conditions

In this thesis we use the constant brightness assumption to derive the optical flow constraint equation. This is a good approximation to the images of non-specular surfaces captured by a moving camera. It does not hold in general, in the case of a moving rigid object in front of stationary camera. Jourjine [37] provides a theoretical extension to this framework to the case of rigid object motion assuming a Lambertian surface model and parallel lighting. In this case both surface and albedo can be recovered.

Tracking and Reconstruction for Non-rigid Motion

How do we extend this framework to allow for more complex motion models such as smooth non-rigid motion. For example, suppose we are trying to reconstruct the head of someone talking or a person moving. We could use multiple cameras to provide us with synchronous images from multiple views (binocular video streams). How do we combine sequences of such views to form an extended scene reconstruction as was done in Chapter 5?

Where to Look Next?

Site reconstruction, the creation of 3D model of the surroundings, provides an exciting challenge beyond geometry and photometry, and deeper into the realm of artificial intelligence. An active vision system for site reconstruction has to reason about the current 3D model and decide where to look (and where to move) to efficiently add information to the 3D model.

In the 3D reconstruction from Chapter 5, shown again in Figure 8-1, there are noticeable holes: behind the trees and other obstacles. Some of these holes could have

been filled in, if the camera was able to actively look down while passing overhead. Filling in other gaps would require actively changing the camera path. A complete system should also be able to detect regions of interest and sample those areas with higher resolution in order to correctly determine the shape of objects. It is an engineering detail to decide if this is to be done by a 'foveated' camera (a camera with non-uniform spatial sampling) or by a separate camera with a longer focal length.

The capability to reason about the 3D world and make active changes in the behavior is not only useful for site reconstruction but also for search and rescue operations. The system should be able to determine where an object of certain size might be hidden from view, and then determine how to move to search those areas.

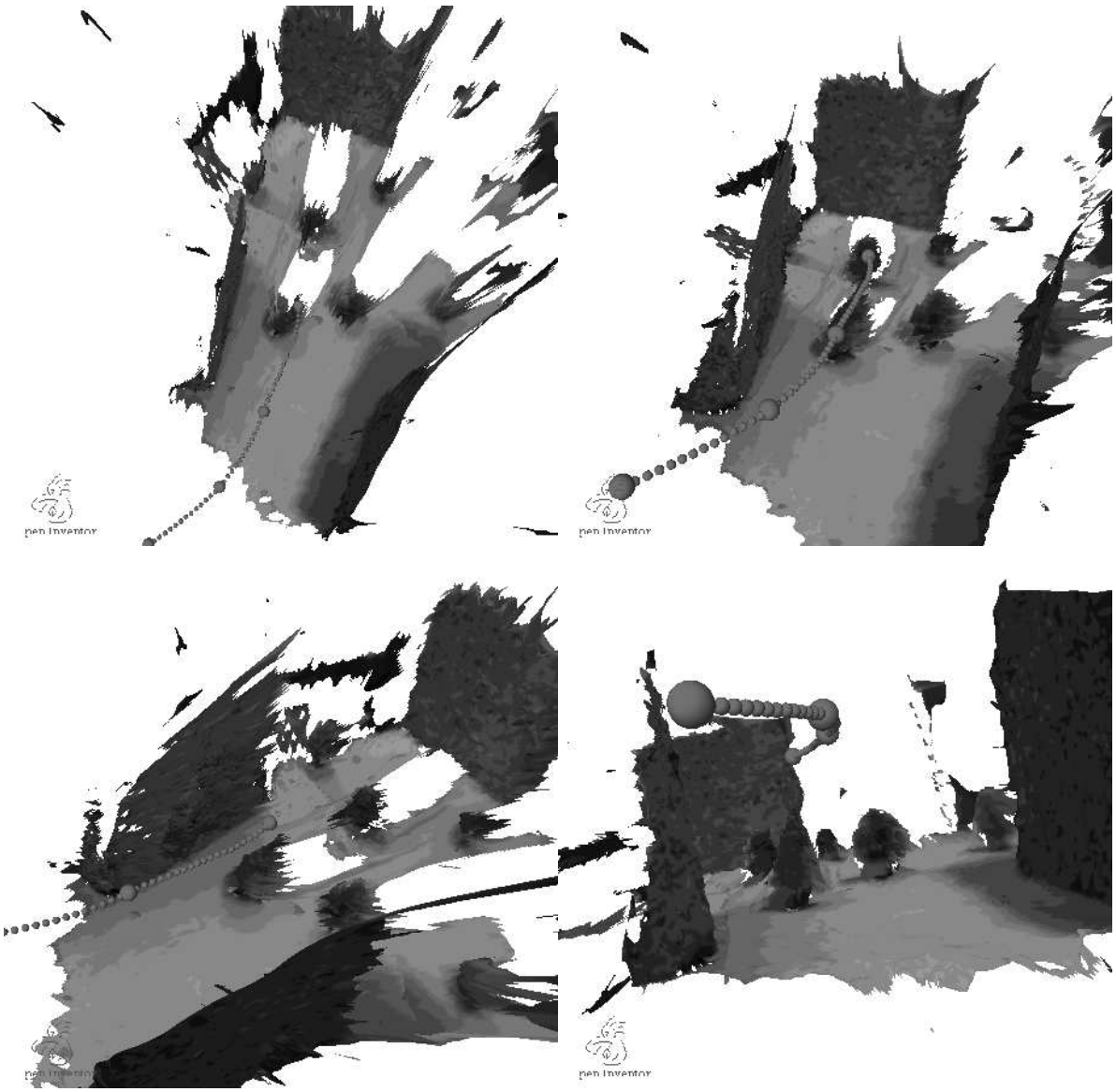


Figure 8-1: *Euclidean reconstruction of the extended scene created by aligning four separate reconstructions into one coordinate system using the recovered camera motion. The small spheres show the camera path. The large spheres show the camera positions of the four Euclidean reconstructions. There are noticeable holes in the reconstruction: behind the trees and other obstacles. Some of these holes could be filled in the camera was able to actively look down while passing overhead. Other holes would require changing the camera path.*

Bibliography

[1] M. Abdel-Mottaleb, R. Chellappa, and A. Rosenfield. Binocular motion stereo using map estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 321–327, New York, NY, June 1993.

[2] S. Avidan and A. Shashua. Tensorial transfer: On the representation of $n > 3$ views of a 3D scene. In *Proceedings of the ARPA Image Understanding Workshop*, Palm Springs, CA, February 1996.

[3] S. Avidan and A. Shashua. View synthesis in tensor space. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Puerto Rico, June 1997.

[4] A. Azarbayejani and A. P. Pentland. Recursive estimation of motion, structure and focal length. *pami*, 17(6):562–575, June 1995.

[5] J. R. Bergen, P. Anandan, K. J. Hanna, and R. Hingorani. Hierarchical model-based motion estimation. In *Proceedings of the European Conference on Computer Vision*, Santa Margherita Ligure, Italy, June 1992.

[6] J. Bergen, P. Anandan, K. Hanna, and R. Hingorani. Hierarchical model-based motion estimation. In *Proceedings of the European Conference on Computer Vision*, Santa Margherita Ligure, Italy, June 1992.

[7] J. Bergen and R. Hingorani. Hierarchical motion-based frame rate conversion. Technical report, David Sarnoff Research Center, 1990.

[8] D. C. Brown. Close-range camera calibration. *Photogrammetric Engineering*, 37:855–866, 1971.

[9] T. Buchanan. On the critical set for photogrammetric reconstruction using line tokens in p3(c). *Geometriae Dedicata*, 44:223–232, 1992.

[10] P. J. Burt and E. H. Adelson. The laplacian pyramid as a compact image code. *IEEE Transactions on Communications*, 31:532–540, 1983.

[11] R. Deriche and G. Giraudon. Accurate corner detection: An analytical study. In *Proceedings of the International Conference on Computer Vision*, pages 66–70, Osaka, Japan, December 1990.

[12] F. Devernay and O. D. Faugeras. Automatic calibration and removal of distortion from scenes of structured environments. In *Proceedings of SPIE Conference*, San Diego, CA, July 1995.

[13] F. Du and M. Brady. Self calibration of the intrinsic parameters of cameras for active vision systems. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 477–482, New York, NY, June 1993.

[14] R. Dutta and M. Snyder. Robustness of correspondence based structure from motion. In *Proceedings of the International Conference on Computer Vision*, pages 106–110, Osaka, Japan, December 1990.

[15] O. Faugeras and T. Papadopoulo. A nonlinear method for estimating the projective geometry of three views. To be published ICCV 98, January 1998.

[16] O. D. Faugeras. *Three Dimensional Computer Vision: a Geometric Viewpoint*. The MIT Press, Cambridge, MA, 1993.

[17] O. D. Faugeras and B. Mourrain. On the geometry and algebra of the point and line correspondences between N images. In *Proceedings of the International Conference on Computer Vision*, pages 951–956, Cambridge, MA, June 1995. IEEE Computer Society Press, IEEE Computer Society Press.

[18] C. Fermuller and Y. Aloimonos. Global rigidity constraints in image displacement fields. In *Proceedings of the International Conference on Computer Vision*, Cambridge, MA, June 1995. IEEE Computer Society Press, IEEE Computer Society Press.

[19] C. Fermuller and Y. Aloimonos. On the geometry of visual correspondence. *International Journal of Computer Vision*, 21(3):223–47, February 1997.

[20] J. G. Fryer and S. O. Mason. Rapid lens calibration of a video camera. *Photogrammetric Engineering*, 55(4):437–442, April 1989.

[21] P. Fua. Reconstructing complex surfaces from multiple stereo views. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1078–1085. IEEE Computer Society Press, June 1995.

[22] E. Grosso and M. Tistarelli. Active/dynamic stereo vision. *Pattern Analysis and Machine Intelligence*, 17(11):1117–1128, November 1995.

[23] K. J. Hanna and N. E. Okamoto. Combining stereo and motion analysis for direct estimation of scene structure. In *Proceedings of the International Conference on Computer Vision*, Berlin, Germany, May 1993.

[24] R. Hartley. Lines and points in three views — a unified approach. In *Proceedings of the ARPA Image Understanding Workshop*, Monterey, CA, November 1994.

[25] R. Hartley. A linear method for reconstruction from lines and points. In *Proceedings of the International Conference on Computer Vision*, pages 882–887, Cambridge, MA, June 1995.

[26] R. Hartley. In defence of the 8-point algorithm. In *Proceedings of the International Conference on Computer Vision*, pages 1064–1070, Cambridge, MA, June 1995. IEEE Computer Society Press, IEEE Computer Society Press.

[27] R. Hartley. A linear method for reconstruction from lines and points. In *Proceedings of the International Conference on Computer Vision*, Cambridge, MA, June 1995. IEEE Computer Society Press, IEEE Computer Society Press.

[28] D. J. Heeger and A. Jepson. Simple method for computing 3d motion and depth. In *Proceedings of the International Conference on Computer Vision*, pages 96–100, Osaka, Japan, December 1990.

[29] J. Heel. Direct estimation of structure and motion from multiple frames. AI Memo 1190, Massachusetts Institute of Technolgy Artificial Intelligence Labratory, March 1990.

[30] A. Heyden. Reconstruction from image sequences by means of relative depths. In *Proceedings of the International Conference on Computer Vision*, pages 1058–1063, Cambridge, MA, June 1995. IEEE Computer Society Press, IEEE Computer Society Press.

[31] B. K. P. Horn. *Robot Vision*. MIT Press, Cambridge, Mass., 1986.

[32] B. K. P. Horn. Relative orientation revisited. *Journal of the Optical Society of America*, 8:1630–1638, 1991.

[33] B. K. P. Horn and B. G. Schunk. Determining optical flow. *Artificial Intelligence*, 17:185–203, 1981.

[34] B. K. P. Horn and E. J. Weldon, Jr. Direct methods for recovering motion. *International Journal of Computer Vision*, 2:51–76, 1988.

[35] T. Huang and C. Lee. Motion and structure from orthographic projections. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-11:536–540, 1989.

[36] M. Irani, B. Rousso, and S. Peleg. Recovery of ego-motion using image stabilization'. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 454–460, Seattle, Washington, June 1994.

[37] A. N. Jourjine. Surface and albedo from a sequence of images. Siemens Corp, April 1997.

[38] T. Kanade, Okutomi, and Nakahara. A multiple-baseline stereo method. In *Proceedings of the ARPA Image Understanding Workshop*, pages 409–426. Morgan Kaufmann, San Mateo, CA, January 1992.

[39] R. Koch. 3-d surface reconstruction from stereoscopic image sequences. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 109–114. IEEE Computer Society Press, June 1995.

[40] R. Kumar and P. Anandan. Direct recovery of shape from multiple views: A parallax based approach. In *Proceedings of the International Conference on Pattern Recognition*, Jerusalem, Israel, October 1994.

[41] H. Longuet-Higgins. A computer algorithm for reconstructing a scene from two projections. *Nature*, 293:133–135, 1981.

[42] H. Longuett-Higgins and K. Prazdny. The interpretation of a moving retinal image. In *Proceedings of the Royal Society of London B*, volume 208, pages 385–397. Royal Society of London, 1980.

[43] B. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *Proceedings IJCAI*, pages 674–679, Vancouver, 1981.

[44] S. Maybank. The critical line congruence for reconstruction from three images. *Applicable Algebra in Engineering Communication and Computing (AAECC)*, 6:89–113, 1995.

[45] I. S. McQuirk. An analog vlsi chip for estimating the focus of expansion. AITR 1577, MIT, 1996.

[46] D. Michaels. *Exploiting Continuity-in-Time in Motion Vision*. PhD thesis, Department of Electrical engineering and Computer Science, Massachusetts Institute of Technology, May 1992.

[47] J. J. More, B. S. Garbow, and K. E. Hillstrom. User guide for minpack-1. Technical Report ANL-80-74, Argonne National Laboratory, Argonne, Illinois, August 1980.

[48] J. Mundy and A. Zisserman. Appendix — projective geometry for machine vision. In J. Mundy and A. Zisserman, editors, *Geometric invariances in computer vision*. MIT Press, Cambridge, 1992.

[49] H. H. Nagel. Direct estimation of optical flow and of its derivates. In G. A. Orban and H. H. Nagel, editors, *Artificial and Biological Vision Systems*, Basic Research Series, pages 193–224. Springer-Verlag, 1992.

[50] N. Navab, O. D. Faugeras, and T. Vieville. The critical sets of lines for camera displacement estimation: A mixed euclidean-projective and constructive approach. In *Proc. 4th International Conference on Computer Vision*, Berlin, Germany, May 1993. IEEE.

[51] N. Navab, R. Deriche, and O. D. Faugeras. Recovering 3d motion and structure from stereo and 2d token tracking cooperation. In *Proceedings of the International Conference on Computer Vision*, pages 513–516, Osaka, Japan, December 1990.

[52] S. Negahdaripour and B. K. P. Horn. Direct passive navigation. *Pattern Analysis and Machine Intelligence*, 9(1):168–176, 1987.

[53] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge University Press, second edition edition, 1992.

[54] L. Quan and T. Kanade. Affine structure from line correspondences with uncalibrated affine cameras. *Pattern Analysis and Machine Intelligence*, 19(8):834–845, August 1997.

[55] J. Reiger and D. T. Lawton. Processing differential image motion. *Journal of the Optical Society of America*, 2:354–359, 1985.

[56] H. Sawhney. 3D geometry from planar parallax. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 929–934, Seattle, Washington, June 1994.

[57] A. Shashua. Algebraic functions for recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(8):779–789, 1995.

[58] A. Shashua. Trilinear tensor: The fundamental construct of multiple-view geometry and its applications. Submitted for journal publication, June 1997.

[59] A. Shashua and S. Avidan. The rank4 constraint in multiple view geometry. In *Proceedings of the European Conference on Computer Vision*, Cambridge, UK, April 1996.

[60] A. Shashua and N. Navab. Relative affine structure: Theory and application to 3D reconstruction from perspective views. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 483–489, Seattle, Washington, 1994.

[61] A. Shashua and N. Navab. Relative affine structure: Canonical model for 3D from 2D geometry and applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(9):873–883, 1996.

[62] A. Shashua and M. Werman. Trilinearity of three perspective views and its associated tensor. In *Proceedings of the International Conference on Computer Vision*, June 1995.

[63] A. Shashua and P. Anandan. Trilinear constraints revisited: Generalized trilinear constraints and the tensor brightness constraint. In *Proceedings of the ARPA Image Understanding Workshop*, Palm Springs, CA, February 1996.

[64] A. Shashua and K. J. Hanna. The tensor brightness constraints: Direct estimation of motion revisited. Technical report, Technion, Haifa, Israel, November 1995.

[65] A. Shashua and M. Werman. Trilinearity of three perspective views and its associated tensor. In *Proceedings of the International Conference on Computer Vision*, pages 920–925, Cambridge, MA, June 1995. IEEE Computer Society Press, IEEE Computer Society Press.

[66] C. Slama, editor. *Manual of Photogrammetry*. American Society of Photogrammetry, 4 edition, 1980.

[67] S. Soatto and P. Perona. Motion from fixation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 817–824, San Fransico, CA, June 1996.

[68] M. Spetsakis and J. Aloimonos. Structure from motion using line correspondences. *International Journal of Computer Vision*, 4(3):171–183, 1990.

[69] M. Spetsakis and J. Aloimonos. A unified theory of structure from motion. In *Proceedings of the ARPA Image Understanding Workshop*, 1990.

[70] M. Spetsakis and J. Aloimonos. A unified theory of structure from motion. In *Proceedings of the ARPA Image Understanding Workshop*, 1990.

[71] G. P. Stein. Internal camera calibration using rotation and geometric shapes. Technical Report AITR-1426, Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge, MA, 1993.

[72] G. P. Stein. Accurate internal camera calibration using rotation, with analysis of sources of error. In *Proceedings of the International Conference on Computer Vision*, pages 230–236. IEEE Computer Society Press, June 1995.

[73] G. P. Stein. Lens distortion calibration using point correspondences. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Puerto Rico, June 1997.

[74] G. P. Stein and A. Shashua. Model based brightness constraints: On direct estimation of structure and motion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Puerto Rico, June 1997.

[75] R. Szeliski and S. Kang. Recovering 3d shape and motion from image streams using non-linear least squares. Technical Report CRL 93/3, DEC CRL, March 1992.

[76] R. Szeliski and S. B. Kang. Direct methods for visual scene reconstruction. In *Proceedings IEEE Workshop on Representation of Visual Scenes*, Cambridge, MA, June 1995.

[77] T. Y. Tian, C. Tomasi, and D. J. Heeger. Comparison of approaches to egomotion computation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 315–320, San Francisco, California, June 1996.

[78] A. P. Tirumalai, B. G. Schunk, and R. C. Jain. Dynamic stereo with self calibration. In *Proceedings of the International Conference on Computer Vision*, pages 466–470, Osaka, Japan, December 1990.

[79] C. Tomasi and T. Kanade. Shape and motion from image streams – a factorization method. *International Journal of Computer Vision*, 9(2):137–154, 1992.

[80] B. Triggs. Matching constraints and the joint image. In *Proceedings of the International Conference on Computer Vision*, pages 338–343, Cambridge, MA, June 1995. IEEE Computer Society Press, IEEE Computer Society Press.

[81] S. Ullman and R. Basri. Recognition by linear combination of models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-13:992—1006, 1991. Also in M.I.T AI Memo 1152, 1989.

[82] A. M. Waxman and J. H. Duncan. Binocular image flows: steps towards stereo-motion fusion. *Pattern Analysis and Machine Intelligence*, 8(6):715–729, November 1986.

[83] J. Weng, N. Ahuja, and T. Huang. Two view matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 64–73, Tampa, Florida, December 1989.

[84] J. Weng, T. Huang, and N. Ahuja. Motion and structure from line correspondences: Closed form solution, uniqueness and optimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(3), 1992.

[85] J. Weng, P. Cohen, and N. Rebibo. Motion and structure estimation from stereo image sequnces. *IEEE Transactions on Robotics and Automation*, 8(3):362–382, June 1992.

[86] J. Weng et al. Camera calibration with distortion models and accuracy evaluation. *Pattern Analysis and Machine Intelligence*, 14:965–980, 1992.

[87] G. Wolberg. *Digital Image Warping*. IEEE Computer Society Press, Los Alamitos, California, 1990.

[88] S. Xuening and M. E. Spetsakis. A comparison of weighted ls methods with ls methods in 3d motion estimation from stereo image sequences. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 200–205, New York, NY, June 1993.

[89] J. W. Young. *Projective Geometry*. The Mathematical Society of America, 1930.

[90] Z. Zhang. On the epipolar geometry between two images with lens distortion. In *Proceedings of the International Conference on Pattern Recognition*, volume 1, pages 407–411, Vienna, Austria, August a996.

[91] Z. Zhang and O. D. Faugeras. Three dimensional motion computation and segementation in a long sequnce of stereo frames. *International Journal of Computer Vision*, 7(3):211–241, 1992.