ABSTRACT

Title of Dissertation: Statistical Analysis of 3D Modeling From Monocular Video Streams

Amit K. Roy Chowdhury, Doctor of Philosophy, 2002

Dissertation directed by: Professor Rama Chellappa
Department of Electrical and Computer Engineering

3D scene modeling from a video sequence is considered to be one of the most important problems in computer vision. Its successful solution has numerous possibilities in applications like multimedia communications, surveillance, virtual reality, automatic navigation, medical prognosis, etc. One of the most powerful techniques for solving this problem is known as structure from motion (SfM). Briefly, the SfM problem is about recovering the absolute or relative depth of static and moving objects using video acquired from single or multiple video cameras. The most challenging problem is when only a monocular video is present and we require a dense estimate of the depth. Successful solution of this problem requires a detailed understanding of the geometry of the 3D world and its 2D projections on the image planes. However, the motion between adjacent frames of a video sequence is usually very small, thus introducing large errors in its estimation. Hence, in order to obtain a satisfactory solution, it is important to

understand the statistics of these errors and their interaction with the geometry of the problem. The overall aim of this thesis is to show how to combine the statistics describing the quality of the input video data with an understanding of the geometry, in order to obtain an accurate 3D scene reconstruction from a video sequence using the optical flow model.

In our work, we pose the 3D reconstruction problem in an estimation-theoretic framework. We adopt the optical flow paradigm for modeling the motion between the frames of the video sequence. We show how the statistics of the errors in the input motion estimates are propagated through the 3D reconstruction algorithm and affect the quality of the output. We present a new result: that the 3D estimate is always statistically biased, and the magnitude of this bias is significant. In order to demonstrate our analysis in a practical application, we consider the problem of reconstructing a 3D model of a human face from video. An algorithm is proposed that obtains a robust 3D model by fusing two-frame estimates using stochastic approximation theory and then combines it with a generic face model in a Markov chain Monte Carlo optimization procedure. We address the question of how to automatically evaluate the quality of a 3D reconstruction from a video sequence, and present a criterion using concepts from information theory. Finally, we propose a probabilistic registration algorithm that extends the results of our work to create holistic 3D models from multiple video streams.

Statistical Analysis of 3D Modeling From Monocular Video Streams

by

Amit K. Roy Chowdhury

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2002

Advisory Committee:

Professor Rama Chellappa, Chairman/Advisor
Professor Azriel Rosenfeld
Associate Professor Adrian Papamarcou
Assistant Professor Min Wu
Professor Larry Davis

DEDICATION

To my parents

ACKNOWLEDGEMENTS

I would like to express my deep gratitude to my advisor, Dr. Rama Chellappa. He introduced me to this problem that has absorbed most my professional life over the past few years, guided me through the process that reached fruition in this thesis, and held out a beacon of hope when the path seemed like a long dark tunnel. Moreover, through his daily interactions he has given me something to live up to by his ever-friendly attitude and his emphasis on honesty and integrity.

I am also grateful to my committee members, including Dr. Larry Davis, Dr. Adrian Papamarcou, Dr. Azriel Rosenfeld and Dr. Min Wu. I would like to thank them for the time they spent discussing different technical issues that arose in the course of this work and for reading and commenting on the thesis. I am very grateful to many of my teachers at different institutions who instilled in me the quest for knowledge and the courage to probe the unknown. I would like to take this opportunity to thank many of my colleagues for their help and advice, without which this thesis would not have seen the light of day. It is impossible to acknowledge all of them individually, but I would like to particularly mention Dr. A. N. Rajagopalan, Dr. Gang Qian, Dr. Murari Srinivasan, Dr. Sridhar Srinivasan, Dr. Venu Govindu, Dr. Hankyu Moon, Mr. Shaohua Zhao, Mr.

Amit Kale, Ms. Namrata Vaswani, Mr. Kaushik Chakraborty and Mr. Damianos Karakos. I would also like to thank Mr. G.S.Rao, Mr. S. Krishnamurthy and Mr. Tai Vo for helping build the 3D face modeling system, which is based on some of the ideas contained in this thesis. I have very much enjoyed my stay at Maryland for the last four years and this is in so small part due to the excellent friends I have had: Indrajit Bhattacharya, Kaushik Chakraborty, Sanjili and Kaushik Ghosh, Ayush Gupta, Asmita and Anand Gupte, Madhumita and Bikash Koley, Harsh Mehta, Souvik Mitra and Ayan Roy Chowdhury, to name but a few.

Words cannot express my gratitude and indebtedness to my parents, who have given up so much in life in order that I could reach this stage today. Thinking about the sacrifices they made humbles me. We usually take their unconditional love for granted, but I would like to take this opportunity to say "Thank you". I would also like to thank my younger brother for his love and understanding. Last, but not the least, I would like to express my gratefulness to my wife, Sumita, for patiently bearing with me through the preparation of this thesis and supporting me throughout. She will always be a source of inspiration for me.

TABLE OF CONTENTS

List of Tables **ix**

List of Figures **x**

1 Introduction **1**
1.1 Literature Review 2
1.2 Contributions of the Thesis 5
1.3 Organization of the Thesis 7

2 Uncertainty Analysis and Propagation **9**
2.1 Introduction 9
2.2 The Basic Equations of SfM 10
2.3 Qualitative Analysis of 3D Estimates 12
2.3.1 Distribution of Depth Sub-Estimates 12
2.3.2 Time Series Analysis 14
2.4 Error Covariance of 3D Reconstruction 19
2.4.1 Proof of Error Covariance Result 21
2.4.2 Unknown FOE 23
2.4.3 Structure of $\mathbf{R_z}$ 24
2.5 Performance Analysis for Multiple Frames 27
2.5.1 Error Covariance Calculation 27
2.5.2 Significance of Multi-frame Distortion 28
2.6 Conclusions 30

3 Bias in Structure Estimate **31**
3.1 Introduction 31
3.2 Bias in Depth Reconstruction 33
3.2.1 Problem Formulation and Result 33
3.2.2 Computation of Bias Term 35
3.3 Analysis of the Bias 41
3.3.1 Bias in Multi-frame Reconstruction 42
3.3.2 Connection to the Human Visual System 43

3.4 The Generalized CRLB for SfM . . . 43
3.4.1 Computing the Fisher Information of Unbiased Structure and Motion Parameters . . . 44
3.4.2 Computing the Generalized CRLB . . . 47
3.5 Simulation Results . . . 49
3.5.1 Effect of Bias on Reconstruction . . . 49
3.5.2 Variation of Bias with Individual Camera Motion Parameters 51
3.5.3 The Generalized CRLB . . . 54
3.6 Conclusion . . . 55

4 3D Face Reconstruction Algorithm 57
4.1 Introduction . . . 57
4.1.1 Incorporating a Generic Model in an Energy Function Minimization Framework . . . 59
4.2 Estimating 3D Structure and Motion From Video . . . 61
4.2.1 Estimating 3D Depth . . . 62
4.2.2 Camera Motion Tracking: . . . 64
4.2.3 The Reconstruction Algorithm . . . 66
4.3 Incorporating the Generic Model in 3D Face Reconstruction . . . 68
4.3.1 The Optimization Function: . . . 68
4.3.2 Mesh Registration: . . . 71
4.3.3 The Generic Mesh Algorithm: . . . 72
4.4 3D Face Model Results . . . 73
4.4.1 Overview of Implementation Strategy . . . 73
4.4.2 SfM Algorithm . . . 74
4.4.3 Reconstruction Example Without Generic Model . . . 75
4.4.4 The Line Process and Neighborhood Set . . . 77
4.4.5 The Optimization Procedure . . . 79
4.4.6 Texture Mapping . . . 80
4.4.7 Face Modeling: Different Examples . . . 81
4.5 Conclusions . . . 82

5 Evaluating the Quality of 3D Reconstructions 85
5.1 Introduction . . . 85
5.1.1 Information Theoretic Concepts in Image and Video Processing . . . 87
5.2 Problem Formulation . . . 88

5.2.1 Notation . . . 90
5.2.2 System Model . . . 90
5.3 Incremental Mutual Information . . . 92
5.3.1 Estimating the Mutual Information . . . 95
5.4 A Case Study: Reconstructing With Gaussian Noise . . . 97
5.4.1 An Estimation-Theoretic Interpretation: . . . 99
5.5 Simulation Results . . . 101
5.5.1 Experiment 1 . . . 101
5.5.2 Experiment 2 . . . 101
5.5.3 Experiment 3 . . . 102
5.6 Conclusions . . . 104

6 Registration of Partial 3D Models 105
6.1 3D Registration Using Prior Models . . . 110
6.1.1 Obtaining the Partial Models . . . 110
6.1.2 Formulation of the Registration Problem . . . 110
6.1.3 Computing the Feature Correspondence Probabilities . . . 111
6.1.4 Prior Information . . . 112
6.1.5 Identifying Unpaired Features . . . 112
6.1.6 Correspondence Matrix . . . 112
6.2 Matching the Spatial Arrangement of Features . . . 113
6.3 The Correspondence Algorithm . . . 114
6.3.1 Reducing the Search Space . . . 115
6.4 Experimental Analysis and Applications . . . 115
6.4.1 Feature Selection and Prior Extraction . . . 116
6.4.2 Estimation of Posterior Probabilities . . . 116
6.4.3 Matching the Spatial Arrangement of Features . . . 118
6.4.4 Importance of Prior Information . . . 118
6.4.5 Importance of Considering the Relative Configuration of Features . . . 119
6.4.6 Application to 3D Model Alignment . . . 120
6.5 Conclusions . . . 122

7 Conclusions and Future Work 127
7.0.1 Future Work . . . 129

A Stochastic Approximation 132

B Computation of Mutual Information **135**

Bibliography **138**

LIST OF TABLES

LIST OF FIGURES

2.1 One frame from each of the two video sequences: (a) represents an image from an indoor video sequence and (b) from an outdoor video sequence. These two sequences were used in the qualitative analysis. 13
2.2 Plot of estimates of the moments and cumulants of the two-frame depth for the outdoor house sequence against the tracked feature points. Skewness = 1.1; Kurtosis = 3.2 ⇒ right skewed and peaked distribution. 15
2.3 Plot of estimates of the moments and cumulants of the two-frame depth for the face sequence against the tracked feature points. Skewness = −0.25; Kurtosis = 1.9 ⇒ left skewed and flat distribution. 16
2.4 A plot of the depth values across 50 frames for four randomly chosen points from the face sequence. It can be seen that there are isolated outliers in all four cases. 17
2.5 Plot of average distortion in reconstruction as a function of the number of frames for two different video sequences. The vertical axis is scaled down by a factor of 10^3. 29

3.1 (a), (b), (c) and (d) are plots of the reconstruction for noise variances σ^2_{x1}, σ^2_{x2}, σ^2_{x3} and σ^2_{x4} in the feature positions, where $\sigma^2_{x4} > \sigma^2_{x3} > \sigma^2_{x2} > \sigma^2_{x1}$. The plots are for the same set of ten 3D points tracked over 15 frames. The camera is moving with constant, non-zero translation and rotation. The solid lines indicate the true depth values, the dashed lines indicate the reconstruction without bias compensation, and the dashed and dotted lines indicate the reconstruction with bias compensation. 50

3.2 Plots of the variation in the bias in inverse depth with the camera motion parameters. The horizontal axes represent the following: (a): $v_x \in (0,1)$cm/frame, (b): $v_y \in (0,1)$cm/frame, (c): $v_z \in (0,1)$cm/frame, (d): $\omega_x \in (0,10)$degrees/frame, (e): $\omega_y \in (0,10)$degrees/frame, (f): $\omega_z \in (0,10)$degrees/frame. The values of the bias on the vertical axis are in percentages of the true inverse depth value. The camera motion is scaled between 0 and 1 on the horizontal axis. 51
3.3 Plots of the trajectories of feature points (top row) and the CRLB of the inverse depth as a function of the number of frames, for different camera motion parameters (bottom row). (a), (b): $x_f = 0, y_f = 10, \omega_x = \omega_y = 1$degree/frame$, \omega_z = 0$; (c), (d): $x_f = 10, y_f = 0, \omega_x = \omega_y = \omega_z = 1$degree/frame; (e), (f): $x_f = 10, y_f = 10, \omega_x = \omega_y = \omega_z = 1$degree/frame. The solid line shows the CRLB for the unbiased estimate and the dotted line for the biased one. 52
3.4 Plots of the trajectories of feature points (top row) and the CRLB of the inverse depth as a function of the number of frames, for different camera motion parameters (bottom row). (a), (b): $x_f = 1, y_f = 1, \omega_x = \omega_y = \omega_z = 0$; (c), (d): Uniform acceleration, $\omega_x = \omega_y = \omega_z = 0$; (e), (f): Uniform acceleration, $\omega_x = \omega_y = \omega_z = 1$degree/frame. The solid line shows the CRLB for the unbiased estimate and the dotted line for the biased one. 53

4.1 Block diagram of the 3D Reconstruction Framework. 62
4.2 Block diagram of the multi-frame fusion algorithm. 66
4.3 (a) and (b) represent two images from the Yosemite video sequence for which the depth was computed. The remaining figures (c) - (i) are results of 3D reconstruction from 15 frames for different viewing angles. 67
4.4 A block diagram representation of the complete 3D modeling algorithm using the generic mesh and SfM algorithm. 72
4.5 Two frames of the original video sequence which is the input to the SfM reconstruction algorithm. 75

4.6 Plot of the variance of the inverse depth for different features in a face sequence. The diameter of the circle at each feature point is proportional to the variance at that feature point. In the second plot, the diagonal elements of $\mathbf{R_h}$ are shown. 76
4.7 (a) represents the distortion of the SfM algorithm with the number of images; (b) depicts one view from the reconstructed model at this stage of the algorithm. 76
4.8 The vertices which form part of the line processes indicating a change in depth values are indicated with black 'x's. 77
4.9 Mesh representations of the 3D models obtained at different stages of the algorithm. (a) represents the generic mesh, (b) the model obtained from the SfM algorithm (the ear region is stitched on from the generic model in order to provide an easier comparison between the different models), (c) the smoothed mesh obtained after the optimization procedure, (d) a finer version of the smoothed mesh for the purpose of texture mapping. 78
4.10 Different views of the 3D model after texture mapping. 79
4.11 Two frames from the second video sequence to which we applied our algorithm. 80
4.12 Different views of the 3D model after texture mapping on the second video sequence. 81
4.13 Two frames from the third video sequence to which we applied our algorithm. 82
4.14 Different views of the 3D model after texture mapping on the third video sequence. 83

5.1 Block diagram representation of the reconstruction framework. $\mathbf{X}$ is the inverse depth that we want to estimate, $(\mathbf{H}(1), ..., \mathbf{H}(L))$ are the intermediate reconstructions (e.g. from pairs of frames), and $\hat{\mathbf{X}}$ is the final fused estimate. 89
5.2 A channel model representation of the 3D reconstruction framework. The channel is characterized by the probability distribution function $P(\mathbf{H}^{(N)}|X)$. 91
5.3 A typical plot of the mutual information in the data processing inequality of (5.2). 94

5.4 The upper plot shows the true depth values of the 3D points (the solid line) and the fused estimate from the intermediate reconstructions from all the frames (the dotted lines). The lower plot shows the decrease in the incremental information with increasing number of frames. 102
5.5 The upper plot shows the true depth values of the 3D points (the solid line) and the fused estimate from the intermediate reconstructions from all the frames (the dotted lines). The lower plot is the change in the mutual information with increasing number of frames. This is the case where the estimated reconstruction does not converge to the true value even with an increasing number of observations. 103
5.6 The above figures represent a 3D reconstruction from video using the method of measuring the incremental mutual information to judge the quality of the result. (a) is one of the images from the video along with the set of tracked features used for the reconstruction. (b) represents the change in the incremental mutual information with the number of images; (c) depicts one view from the reconstructed model. 103

6.1 The output of the corner finder algorithm on two images obtained from projections of the partial models, represented by small dots. 117
6.2 Features identified in the front and side view images by applying a k-means clustering to the output of the corner-finder. 117
6.3 Intensity blocks around the features to be matched in the front view. The numbers represent the positions of the corresponding features in the image. 118
6.4 Intensity blocks around the features to be matched in the side view. The numbers represent the positions of the corresponding features in the image. 119
6.5 The shapes of the significant image attributes in the front view around the feature point whose position in the original image is indicated on top. 120
6.6 The shapes of the significant image attributes in the side view around the feature point whose position in the original image is indicated on top. 121

6.7 The prior information (the shape representation averaged over a large number of viewing angles) which was pre-computed. 122
6.8 The posterior density matrix. 123
6.9 The *a posteriori* probabilities for each of the features in the front image, obtained from each of the rows of the correspondence matrix. 124
6.10 The *a posteriori* probabilities for each of the features in the side image, obtained from each of the columns of the correspondence matrix. 124
6.11 The probability of matching $\mathbf{X}$ against all permutations of $\mathbf{Y}$. The true value is marked with a $\Uparrow$ below the horizontal axis. . . . 125
6.12 The probability of match for each of the features in the front image, for the case where prior information is not available. 125
6.13 The probability of match for the shape of each feature in the front image against all possible combinations of the features in the side view, for the case where prior information is not available. The true value is marked with a $\Uparrow$ below the horizontal axis. 126
6.14 3D models from the front and side which are used as input to the algorithm, and two views of the 3D model obtained after the alignment. 126

Chapter 1

Introduction

Extraction of the 3D structure of a scene from a sequence of images is termed the structure from motion (SfM) problem. It has been one of the central problems in computer vision for the past two decades, because of potential applications in numerous areas like multimedia communication, virtual reality, automatic navigation, robotics, surveillance, human recognition and identification, medical diagnosis, etc. Extensive literature on the subject can be found in [1], [2], [3], [4], [5], and [6], among others.

The traditional approach to this problem is to recover the 3D structure from a pair of images of the scene. This is known as the geometric stereo approach and is based on the concept of triangulation: if two corresponding points on two images are known, then the 3D object point must lie at the intersection of the rays through those two points. While the theoretical basis of stereo is straightforward, its implementation is a non-trivial issue. It requires that we establish correspondence between points in two images, which by itself is a difficult problem.

The other approach to this problem is to reconstruct the 3D scene from a monocular video sequence using optical flow [7]. The optical flow is a model

for the motion between adjacent pairs of frames in a video sequence, which can then be used to extract the 3D structure [8]. The challenge in this approach is that the motion between adjacent frames of the video sequence is usually very small, thus making the motion estimation process extremely sensitive to noise. However, as we show in this thesis, it is possible to derive closed-form mathematical expressions for the error in the 3D estimates, which can then be used to obtain robust reconstructions.

1.1 Literature Review

Pioneered by the seminal work of Longuet-Higgins [9] and the eight-point algorithm developed independently by Tsai and Huang [10], SfM has been one of the most vibrant research areas in computer vision. Most of the earlier work concentrated on developing efficient algorithms for reconstructing 3D structure from multiple frames. The use of multiple frames was motivated by the hope that the extra information would help correct the flaws that are inevitably present in two-frame reconstructions. The problem of tracking an object across multiple frames was addressed in [11] where a known object and its past position and velocity were used to predict its new location. Broida and Chellappa investigated the use of the extended Kalman filter [12] for estimating motion and structure from a sequence of monocular images [13]. Azarbayejani and Pentland extended this work to include the estimation of the focal length of the camera, along with motion and structure [14]. Tomasi and Kanade developed an algorithm for shape and motion estimation under orthographic projection using the factorization theorem [15]. Szeliski and Kang proposed a non-linear least squares optimization scheme using the Levinburg-Marquardt method for solving the problem [16].

Oliensis developed a multi-frame algorithm under perspective projection in [17], which was extended recently in [18]. Most of these multi-frame methods can be characterized as batch processing (but not necessarily recursive), which means that the problem of estimating the motion and structure is formulated as one of minimizing an objective function defined as a sum of squares of the differences between the actual observed images and the projections of their estimated 3D locations, over all tracked positions and images (bundle adjustment). In contrast, Thomas and Oliensis proposed a fusion algorithm that computes the final reconstruction from intermediate reconstructions by analyzing the uncertainties in them, rather than from image data directly [19]. The error in the individual reconstructions was modeled as a combination of the error in the estimated camera motion and the error in tracking the image coordinates, assuming the image noise to be independent and zero mean.

Another focus of research in SfM has been on understanding the sensitivity of the solution when the input data is noisy. SfM algorithms often make assumptions about the inputs (e.g. perfect image correspondences) that are violated in practice and lead to errors in the reconstruction. In order to make our algorithms work in the presence of these errors, we might be tempted to introduce preprocessing stages to minimize their effects (e.g. design better correspondence algorithms). However, the sources of the errors are often unknown; preprocessing stages are independent research problems in their own right (the correspondence problem is a very good example of this); and incorporating these stages adds to the total computational cost of the final system. The alternative is to understand these errors in a statistical sense and account for their influence within the structure of the main algorithm.

Many researchers have analyzed the sensitivity and robustness of several of the existing algorithms for reconstructing a scene from a video sequence. The work of Weng et al. [20, 21] is one of the earliest instances of estimating the standard deviation of the error in reconstruction using first-order perturbations in the input. The Cramer-Rao lower bound on the estimation error variance of the structure and motion parameters from a sequence of monocular images was derived in [22]. Young and Chellappa derived bounds on the estimation error for structure and motion parameters from two images under perspective projection [23] as well as from a sequence of stereo images [24]. Similar results were derived in [25] and the coupling of the translation and rotation for small field of view was studied. Daniilidis and Nagel have also shown that many algorithms for three-dimensional motion estimation, which work by minimizing an objective function leading to an eigenvector solution, suffer from instabilities [26]. They examined the error sensitivity in terms of translational direction, viewing angle and distance of the moving object from the camera. Zhang's work [27] on determining the uncertainty in the estimation of the fundamental matrix is another important contribution in this area. Haralick showed how well-known estimation techniques could be used to propagate additive random perturbations through many vision algorithms [28]. Chiuso, Brockett and Soatto [29] have analyzed SfM in order to obtain provably convergent and optimal algorithms. Oliensis emphasized the need to understand algorithm behavior and the characteristics of the natural phenomenon that is being modeled [6]. Ma, Kosecka and Sastry [30] also addressed the issues of sensitivity and robustness in their motion recovery algorithm. Recently, Sun, Ramesh and Tekalp [31] proposed an error characterization of the factorization method for 3-D shape and

motion recovery from image sequences using matrix perturbation theory. Morris and Kanatani extended the covariance-based uncertainty calculations to account for geometric indeterminacies, referred to in the literature as *gauge* freedom [32].

1.2 Contributions of the Thesis

In this thesis, we consider the problem of reconstructing the 3D structure of a scene from monocular video using optical flow to model the motion between the video frames. The use of optical flow is motivated by the need to obtain a dense estimate of the structure. The overall aim of the thesis is to show how to estimate the quality of the 3D reconstruction as a function of the quality of the input video sequence, and then to use this understanding to build accurate and robust 3D models. Specifically, the thesis makes the following original contributions.

Error Covariance in Reconstruction We derive an explicit expression for the error covariance in the motion and structure estimates as a function of the error covariance in the feature positions in the images. We consider the separate cases where the focus of expansion (FOE) is known and where it is unknown. The derivation uses the implicit function theorem [33]. The result for two frames is extended to multiple frames, resulting in a plot describing the distortion in the 3D reconstruction as a function of the number of frames of the video sequence.

Structure Estimate Is Statistically Biased We prove analytically that the 3D estimate obtained from optical flow is statistically biased. The bias is a result of the fact that the feature positions can be obtained only up to a certain level of accuracy. We show, through simulations, that the

magnitude of the bias is significant compared to the true depth values. We also analyze how the bias is affected by the various camera motion parameters. An interesting observation here is that psychophysicists have noted the existence of systematic biases in observers' magnitude estimation of depth, and our analysis shows that it is also present in the standard mathematical models used to estimate 3D structure.

3D Face Reconstruction Algorithm Most of the existing work on reconstructing a 3D model of a human face from video uses a generic model to initialize the optimization algorithm. The problem with this approach is that the solution often settles to a local minimum near the initialization point, resulting in an estimate which bears the characteristics of the generic model, rather than the particular face being modeled from the video. We propose an alternative algorithm which uses the theoretical understanding regarding the quality of the reconstruction and postpones the introduction of the generic model to a later stage in the algorithm. A 3D estimate of the structure is obtained purely from the video sequence using robust statistics and stochastic approximation theory. The generic model is then introduced to correct for the persisting errors by comparing the geometric trends in the two models. The generic model is combined with the 3D estimate in a Markov chain Monte Carlo framework.

Quality Evaluation of 3D Reconstruction An important question in 3D reconstruction from video is how to automatically evaluate the quality of the final estimate. Probably more important: how do we identify situations where the quality of the video sequence is so poor that even a very large number of frames will not yield a final result with the desired fidelity?

We have tried to answer these questions using ideas from information theory. We propose a criterion termed incremental mutual information (IMI), which estimates the mutual information (MI) between the actual 3D structure and its estimate from a certain number of video frames, and computes the change in the MI as more frames are considered. The MI is computed using Monte Carlo techniques. For the Gaussian noise case, we derive an explicit closed-form expression.

Registration of Partial 3D Models 3D models of a scene are usually obtained by aligning partial models of different portions of the scene. We propose a scheme whereby two separate models, representing the front and side views of a face, are registered by taking advantage of information extracted from multiple video streams of that face or any other similar one. Since this information needs to be collected only once for each class of applications, we call it the prior information, and show that its incorporation can lead to an extremely robust solution.

1.3 Organization of the Thesis

The thesis is organized along the lines of the previous section. In Chapter 2, we derive the results relating the error covariance of the 3D estimate to the error covariances in the input feature positions. The results on the bias in the 3D estimate and its links to the human vision system are derived in Chapter 3. The 3D reconstruction algorithm, incorporating the generic model, is presented in Chapter 4. Chapter 5 describes the information-theoretic criterion for quality evaluation and how it can be computed. The extension of our work to multiple

cameras in the form of registration of partial 3D models is discussed in Chapter 6. Finally, we conclude and outline future extensions and applications of this thesis.

Chapter 2

Uncertainty Analysis and Propagation

2.1 Introduction

Structure from motion algorithms reconstruct the camera motion and 3D depth either from a set of feature points tracked over a number of frames of the video sequence or from the estimated optical flow between pairs of frames. In the discrete case, the optical flow represents the motion estimate at each pixel of a video frame and is mathematically modeled as tracking a feature set, where each pixel is a feature point. Naturally, the quality of the reconstruction is affected by the preciseness with which the features can be tracked. There are two kinds of tracking errors [27]:

Location errors: This is when a point is poorly localized. Usually, the error is small (within a few pixels) and can be assumed to exhibit a Gaussian behavior.

False matches: This is when a particular feature point in one image maps to a completely different feature point in another image. The errors in this case are large and cannot be suitably modeled by Gaussian distributions.

Since the localization errors are usually small, their effects can be captured by the second-order statistics. In this chapter, we analyze how the error covariance in feature positions due to the localization errors affects the error covariance of motion and structure estimates. In Chapter 4, we will explain how we deal with errors due to false matches using robust statistics.

We start by outlining the basic equations for recovering 3D structure using optical flow. Next, we perform an experimental study of the quality of the reconstruction obtained from pairs of frames, which gives us a qualitative idea of the statistics of the errors in the estimates. In Section 2.4, we derive a closed-form analytical expression relating the error covariance in feature positions to the error covariance of the motion and structure estimates. We assume that the covariance matrix of the feature correspondences is known. We will briefly explain how this can be obtained. The extension of the covariance calculations to multiple frames of the video sequence is addressed in the last section of this chapter.

2.2 The Basic Equations of SfM

Consider a coordinate frame attached rigidly to a camera with the origin at the center of perspective projection and the z-axis perpendicular to the image plane. Assume that the camera is in motion with respect to a single rigid-body imaged scene with translational velocity $V = [v_x, v_y, v_z]$ and rotational velocity $\Omega = [\omega_x, \omega_y, \omega_z]$. We assume that the camera motion between two consecutive frames in the video sequence is small, and use the small-motion approximation to the perspective projection model for motion field analysis. If $p(x, y)$ and $q(x, y)$ are the horizontal and vertical velocity fields of a point (x, y) in the image plane,

they are related to the 3D object motion and scene depth by [34]

$$\begin{aligned} p(x,y) &= (xv_z - fv_x)/z(x,y) + \frac{1}{f}xy\omega_x - (f + \frac{1}{f}x^2)\omega_y + y\omega_z \\ q(x,y) &= (yv_z - fyv_y)/z(x,y) + (f + \frac{1}{f}y^2)\omega_x - \frac{1}{f}xy\omega_y - x\omega_z, \end{aligned} \quad (2.1)$$

where f is the focal length of the camera. Analysis of the above pair of equations reveals that only the translational component of the image velocity depends on the 3D location of the scene point; the rotational component depends only on the image position (x, y). Also, the image velocity field is invariant under equal scaling of the depth z and the translational velocity vector V. This is known as the scale ambiguity for 3D reconstruction. It follows that we can determine the relative motion and scene structure only up to a scale factor. Since only the direction of the translational motion can be obtained from (2.1), the equations can be re-written as

$$\begin{aligned} p(x,y) &= (x - fx_f)h(x,y) + \frac{1}{f}xy\omega_x - (f + \frac{1}{f}x^2)\omega_y + y\omega_z \\ q(x,y) &= (y - fy_f)h(x,y) + (f + \frac{1}{f}y^2)\omega_x - \frac{1}{f}xy\omega_y - x\omega_z, \end{aligned} \quad (2.2)$$

where $(x_f, y_f) = (\frac{v_x}{v_z}, \frac{v_y}{v_z})$ is known as the *focus of expansion* (FOE), and $h(x,y) = \frac{v_z}{z(x,y)}$ is the inverse scene depth. For N such corresponding points, the equations can be written in more compact matrix notation. Let us define [8]

$$\begin{aligned} \mathbf{h} &= (h_1, h_2, ..., h_N)^T_{N\times 1} \\ \mathbf{u} &= (p_1, q_1, p_2, q_2, ..., p_N, q_N)^T_{2N\times 1} \\ \mathbf{r}_i &= (x_i y_i, -(1 + x_i^2), y_i)^T_{3\times 1} \\ \mathbf{s}_i &= (1 + y_i^2, -x_i y_i, -x_i)^T_{3\times 1} \\ \mathbf{\Omega} &= (w_x, w_y, w_z)^T_{3\times 1} \\ \mathbf{Q} &= \begin{bmatrix} r_1 & s_1 & r_2 & s_2 & ... & r_N & s_N \end{bmatrix}^T_{2N\times 3} \end{aligned}$$

$$
\begin{aligned}
\mathbf{P} &= \begin{bmatrix} x_1 - x_f & 0 & \cdots & 0 \\ y_1 - y_f & 0 & \cdots & 0 \\ 0 & x_2 - x_f & \cdots & 0 \\ 0 & y_2 - y_f & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & x_N - x_f \\ 0 & 0 & \cdots & y_N - y_f \end{bmatrix}_{2N \times N} \\
\mathbf{B} &= [\mathbf{P} \quad \mathbf{Q}]_{2N \times (N+3)} \\
\mathbf{z} &= \begin{bmatrix} \mathbf{h} \\ \mathbf{\Omega} \end{bmatrix}_{(N+3) \times 1} .
\end{aligned} \tag{2.3}
$$

Then (2.2) can be written as

$$\mathbf{Bz} = \mathbf{u}. \tag{2.4}$$

We want to compute $\mathbf{z}$ from $\mathbf{u}$.

2.3 Qualitative Analysis of 3D Estimates

Our experiments in understanding the properties of two-frame reconstructions will have two parts: statistical distribution of the intermediate depth reconstructions (we will also refer to them as sub-estimates) and time-series analysis of these sub-estimates. Our experiments are conducted on two different image sequences, an indoor sequence of a person's face and an outdoor sequence of a house. One frame from each of these two sequences in shown in Figure 2.1.

2.3.1 Distribution of Depth Sub-Estimates

To obtain a good fused estimate from sub-estimates, one should know how to weight the sub-estimates and their uncertainties in order that the final esti-

(a) (b)

Figure 2.1: One frame from each of the two video sequences: (a) represents an image from an indoor video sequence and (b) from an outdoor video sequence. These two sequences were used in the qualitative analysis.

mate accurately reflects this information. In a general fusion problem, this involves computing the likelihood function [35]. Traditional fusion methods like Kalman filtering provide a computational method for this likelihood function and work well under Gaussian approximation. This typically happens when the sub-estimates have small uncertainties and a Gaussian approximation to reflect their variances is adequate. However, when the sub-estimates have large errors, it is inadequate to approximate their likelihoods as Gaussian.

A standard test for Gaussianity of observations is to analyze their higher-order statistics. It is well known that for Gaussian random variables, all odd central moments are identically zero (this is actually true for any symmetric distribution) and all cumulants of order greater than two are zero [36]. Figures 2.2 and 2.3 show plots of the estimates of the central moments and cumulants of two-frame depth against the feature points. Analysis of these plots reveals that there is significant non-Gaussianity in the distribution function of the depth. For the indoor face sequence, the estimated skewness is -0.25 and the kurtosis is 1.9, while for the outdoor house sequence, the values are 1.1 and 3.2 respectively

(averaged over all features). Knowing that the skewness of a standard normal distribution ($\mathcal{N}(0,1)$) is 0 and the kurtosis is 3 [37], we can infer that the distribution of the depth sub-estimates for the face sequence is left skewed (negative skewness) and flat (kurtosis less than 3), while the same distribution function for the house sequence is right skewed (positive skewness) and peaked (kurtosis greater than 3). What these figures emphasize is that the distribution function of the depth sub-estimates which need to be fused is significantly non-Gaussian and it varies widely depending on the data (in fact, it is impossible to even infer whether the distribution is sub-Gaussian or super-Gaussian). However, it is not possible to infer anything more about the distribution function, thus making it impossible to write down the likelihood function. These observations should be taken into consideration in designing the optimization strategy to be adopted for multi-frame fusion, as will be explained in Chapter 4.

2.3.2 Time Series Analysis

Figure 2.4 shows a plot of the depth values (obtained for pairs of frames) across 50 frames for four randomly chosen points in the face image sequence. It can be seen that there are isolated outliers in all four cases. It is difficult to ascertain the exact cause of the outliers; however, the general reasons for their occurrence can be inferred. Application of least squares estimation techniques in the presence of such outliers will severely affect the estimates. One bad point is often enough to perturb least squares completely. In fact, regression analysis shows that least squares is vulnerable to outliers in both the independent or *explanatory variables* as well as the observations or *response variables* [38]. In our case, the observations are the two-frame depth values which depend on the image correspondences

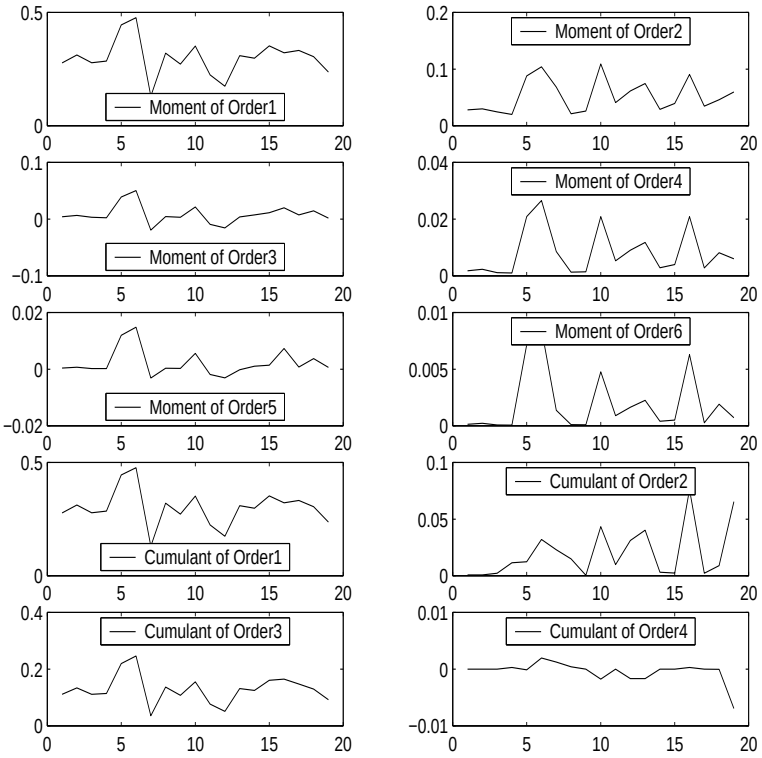


Figure 2.2: Plot of estimates of the moments and cumulants of the two-frame depth for the outdoor house sequence against the tracked feature points. Skewness = 1.1; Kurtosis = 3.2 $\Rightarrow$ right skewed and peaked distribution.

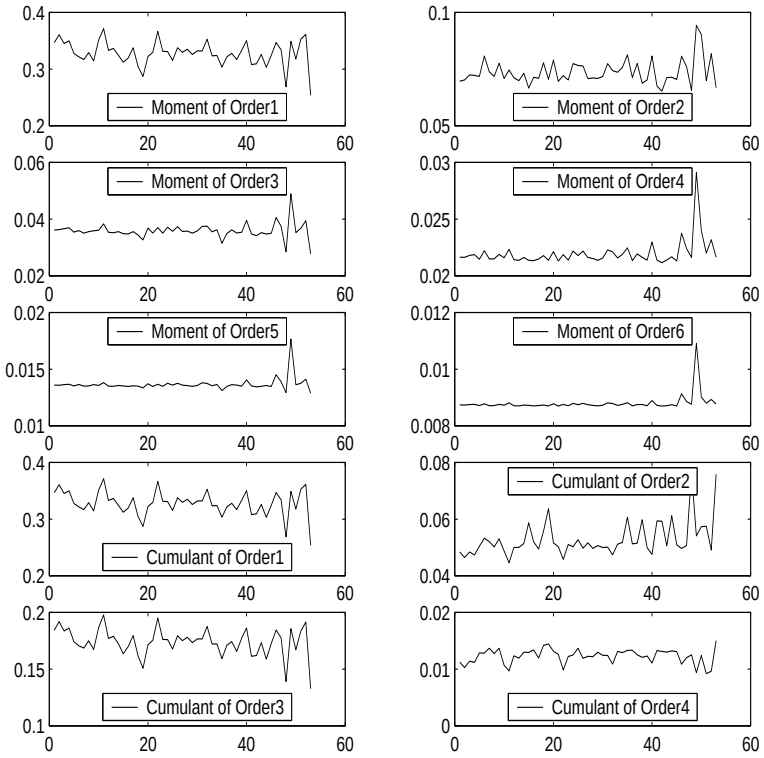


Figure 2.3: Plot of estimates of the moments and cumulants of the two-frame depth for the face sequence against the tracked feature points. Skewness $= -0.25$; Kurtosis $= 1.9 \Rightarrow$ left skewed and flat distribution.

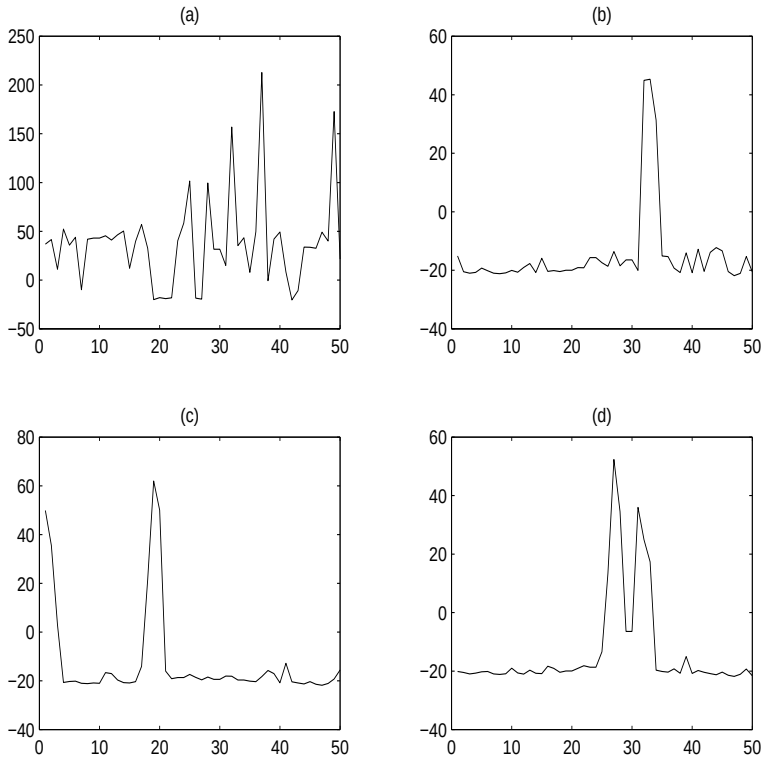


Figure 2.4: A plot of the depth values across 50 frames for four randomly chosen points from the face sequence. It can be seen that there are isolated outliers in all four cases.

(p, q), and which therefore are the explanatory variables. Thus there are outliers in both of the variables and least-squares techniques will perform poorly.

Numerous papers have been published in the statistics literature over the last two decades on designing robust estimators [38]. The two most popular robust methods are *M-estimators* and the *least-median-of-squares* (LMedS) method. A good review of these methods in general and as applied to vision in particular can be found in [1], [2], [39], [40], [41]. While we address the issue of designing a robust 3D reconstruction system in Chapter 4, we will introduce the use of LMedS, which estimates the parameters by solving the nonlinear minimization

of the residual r_i,

$$\min_i \text{median} r_i^2. \tag{2.5}$$

The median is a preferred estimator as it has a high breakdown point. In fact, experiments prove that this method is very robust to outliers due to either bad localization or false matches [1]. However, unlike M-estimators, the LMedS problem cannot be reduced to a weighted least-squares problem, thus complicating its computation. It is also a well-known fact that the efficiency of LMedS is low in the presence of Gaussian noise [39] [1]. As discussed before, the noise in the structure estimates deviates appreciably from Gaussianity and thus LMedS is a good choice for our application.

Given that the two-frame depth observations are non-Gaussian, a linear estimator like the least squares or the linear mean square error estimators is no longer optimal (in the minimum variance sense). We must therefore search over a larger class of non-linear estimators. However, rather than search for a general non-linear estimator, we restrict our search to those estimators which minimize the median of squares. The question now is, is it possible to develop a recursive strategy for this optimization taking into account the statistics of the observations we mentioned previously? Before we can answer that question (in Chapter 4), we need to obtain a quantitative idea of the errors in the reconstruction, which we will now address.

[1] The efficiency of an estimator is defined as the ratio of the lowest achievable variance of the estimated parameters (obtained from the inverse of the Fisher information matrix) and the actual variance obtained from the given method.

2.4 Error Covariance of 3D Reconstruction

In this section, we will derive an expression for the error covariance in 3D reconstruction as a function of the error covariance in the 2D motion estimates. We will use the implicit function theorem to derive a very general result and then introduce different assumptions (like Gaussianity and independence) to calculate simpler expressions.

Recall equation (2.4). Let $\mathbf{z} = \psi(\mathbf{u})$. Expanding ψ in a Taylor series around $E[\mathbf{u}]$,

$$\psi(\mathbf{u}) = \psi(E[\mathbf{u}]) + D_\psi(E[\mathbf{u}])(\mathbf{u} - E[\mathbf{u}]) + \mathcal{O}(\mathbf{u} - E[\mathbf{u}])^2, \tag{2.6}$$

where $\mathcal{O}(x^2)$ denotes terms of order 2 or higher in $\mathbf{x}$ and $D_\psi(\mathbf{x}) = \frac{\partial \psi}{\partial \mathbf{x}}$. Up to a first-order approximation,

$$\psi(\mathbf{u}) - \psi(E[\mathbf{u}]) = D_\psi(E[\mathbf{u}])(\mathbf{u} - E[\mathbf{u}]). \tag{2.7}$$

The covariance of $\mathbf{z}$ can then be written as

$$\begin{aligned} \mathbf{R_z} &= E[(\psi(\mathbf{u}) - E[\psi(\mathbf{u})])(\psi(\mathbf{u}) - E[\psi(\mathbf{u})])^T] \\ &= E[D_\psi(E[\mathbf{u}])(\mathbf{u} - E[\mathbf{u}])(\mathbf{u} - E[\mathbf{u}])^T (D_\psi(E[\mathbf{u}]))^T] \\ &= D_\psi(E[\mathbf{u}])\mathbf{R_u} D_\psi(E[\mathbf{u}])^T \end{aligned} \tag{2.8}$$

where $\mathbf{R_u}$ is the covariance matrix of $\mathbf{u}$ and we have used the first-order approximation that $E[\mathbf{z}] = \psi(E[\mathbf{u}])$. Now consider the cost function

$$\begin{aligned} C &= \frac{1}{2}||\mathbf{Bz} - \mathbf{u}||^2 \\ &= \frac{1}{2}\sum_{i=1}^{n=2N}(u_i - \sum_{j=1}^{N+3} b_{ij} z_j)^2 \\ &= \frac{1}{2}\sum_{i=1}^{n} C_i^2(u_i, \mathbf{z}) \\ &= \frac{1}{2}\sum_{i=1}^{N}(C_{pi}^2 + C_{qi}^2), \end{aligned} \tag{2.9}$$

where C_{pi} and C_{qi} are the components of the cost function corresponding to the p and q components of the motion and b_{ij} is the $(i,j)^{\text{th}}$ element of $\mathbf{B}$.

We will first consider the case where the FOE is known, resulting in a linear system of equations in (2.4) . We will state a result which gives a precise relationship between the error in the image correspondences $\mathbf{R_u}$ and the error in the depth and motion estimates $\mathbf{R_z}$. We will then show how the results can be extended to the case where the FOE is unknown.

Theorem 1 *Define*

$$\begin{aligned}
A_{\bar{i}p} &= [0 \ \cdots \ 0 \ -(x_{\bar{i}} - x_f) \ 0 \ \cdots \ 0 \ -x_{\bar{i}}y_{\bar{i}} \ (1+x_{\bar{i}}^2) \ -y_{\bar{i}}], \\
&= [-(x_{\bar{i}} - x_f)\mathbf{I}_{\bar{i}}(N)| - \mathbf{r}_{\bar{i}}] = [A_{\bar{i}ph}|A_{\bar{i}pm}] \\
A_{\bar{i}q} &= [0 \ \cdots \ 0 \ -(y_{\bar{i}} - y_f) \ 0 \ \cdots \ 0 \ -(1+y_{\bar{i}}^2) \ x_{\bar{i}}y_{\bar{i}}(N) \ x_{\bar{i}}], \\
&= [-(y_{\bar{i}} - y_f)\mathbf{I}_{\bar{i}}(N)| - \mathbf{s}_{\bar{i}}] = [A_{\bar{i}qh}|A_{\bar{i}qm}] \qquad (2.10)
\end{aligned}$$

where $\bar{i} = \lceil i/2 \rceil$ is the upper ceiling of i ($\bar{i}$ will then represent the number of feature points N and $i = 1, ..., n = 2N$) and $\mathbf{I}_n(N)$ denotes a 1 in the n^{th} position of the array of length N and zeros elsewhere. The subscript p in $A_{\bar{i}p}$ and q in $A_{\bar{i}q}$ denotes that the elements of the respective vectors are derived from the p^{th} and q^{th} components of the motion in (2.2). Then

$$\mathbf{R_z} = \mathbf{H}^{-1}\left(\sum_i \frac{\partial C_i^T}{\partial \mathbf{z}}\frac{\partial C_i}{\partial \mathbf{u}}\mathbf{R_u}\frac{\partial C_i^T}{\partial \mathbf{u}}\frac{\partial C_i}{\partial \mathbf{z}}\right)\mathbf{H}^{-T} \qquad (2.11)$$

$$= \mathbf{H}^{-1}\left(\sum_{\bar{i}=1}^{N}\left({A_{\bar{i}p}}^T A_{\bar{i}p} R_{u\bar{i}p} + {A_{\bar{i}q}}^T A_{\bar{i}q} R_{u\bar{i}q}\right)\right)\mathbf{H}^{-T}, \qquad (2.12)$$

and

$$\mathbf{H} = \sum_{\bar{i}=1}^{N}\left({A_{\bar{i}p}}^T A_{\bar{i}p} + {A_{\bar{i}q}}^T A_{\bar{i}q}\right). \qquad (2.13)$$

2.4.1 Proof of Error Covariance Result

We will use the implicit function theorem [33] to prove the above result. The approach is similar to the derivation of the uncertainty in the fundamental matrix [2]. However, it is possible to derive explicit and elegant results for the error covariance in terms of the parameters of (2.2), which would be extremely cumbersome for the case of the fundamental matrix.

Implicit Function Theorem The implicit function theorem states that if f is a continuously differentiable mapping, $f(x, y) = 0$ can be solved uniquely for y in terms of x under certain conditions. We state the theorem precisely as described by Rudin in [33].

Let $\mathbf{f}$ be a $\mathcal{C}'$ mapping of an open set $E \subset \Re^{n+m}$ into $\Re^n$, such that $\mathbf{f}(\mathbf{a}, \mathbf{b}) = \mathbf{0}$ for some point $(\mathbf{a}, \mathbf{b}) \in E$. Put $A = \mathbf{f}'(\mathbf{a}, \mathbf{b})$ and assume that A_x (the derivative matrix of $\mathbf{f}$ with respect to its first argument $\mathbf{x} \in \Re^n$) is invertible. Then there exist open sets $U \in \Re^{n+m}$ and $W \in \Re^m$, with $(\mathbf{a}, \mathbf{b}) \in U$ and $\mathbf{b} \in W$, having the following property: To every $\mathbf{y} \in W$ there corresponds a unique $\mathbf{x}$ such that $\mathbf{f}(\mathbf{g}(\mathbf{y}), \mathbf{y}) = \mathbf{0}$ and

$$\mathbf{g}'(\mathbf{b}) = -(A_x)^{-1} A_y . \diamond \tag{2.14}$$

For our problem, we desire to obtain our parameter of interest $\mathbf{z}$ by minimizing C. Choosing $\mathbf{a} = E[\mathbf{z}]$ and $\mathbf{b} = E[\mathbf{u}]$, let

$$\phi = \frac{\partial C^T}{\partial \mathbf{z}}, \qquad \text{and} \qquad \mathbf{H} = \frac{\partial \phi}{\partial \mathbf{z}}. \tag{2.15}$$

ϕ is a $m \times 1$ vector and $\mathbf{H}$ is a $m \times m$ matrix. Then from the implicit function theorem

$$D_\psi(\mathbf{u}) = -\mathbf{H}^{-1} \frac{\partial \phi}{\partial \mathbf{u}}. \tag{2.16}$$

Thus (2.8) becomes

$$\mathbf{R_z} = \mathbf{H}^{-1}\frac{\partial\phi}{\partial\mathbf{u}}\mathbf{R_u}\frac{\partial\phi^T}{\partial\mathbf{u}}\mathbf{H}^{-T}. \tag{2.17}$$

Then from (2.9) and (2.15),

$$\begin{aligned}
\phi &= \frac{\partial C}{\partial \mathbf{z}}^T = \sum_i C_i \frac{\partial C_i^T}{\partial \mathbf{z}} \\
\mathbf{H} &= \frac{\partial \phi}{\partial \mathbf{z}} = \sum_i \frac{\partial C_i^T}{\partial \mathbf{z}}\frac{\partial C_i}{\partial \mathbf{z}} + \sum_i \frac{\partial^2 C_i^T}{\partial \mathbf{z^2}} \\
&\approx \sum_i \frac{\partial C_i^T}{\partial \mathbf{z}}\frac{\partial C_i}{\partial \mathbf{z}} \\
\frac{\partial\phi}{\partial\mathbf{u}} &\approx \sum_i \frac{\partial C_i^T}{\partial \mathbf{z}}\frac{\partial C_i}{\partial \mathbf{u}}.
\end{aligned} \tag{2.18}$$

Thus equation (2.17) becomes

$$\mathbf{R_z} = \mathbf{H}^{-1}\left(\sum_{ij}\frac{\partial C_i^T}{\partial \mathbf{z}}\frac{\partial C_i}{\partial \mathbf{u}}\mathbf{R_u}\frac{\partial C_j^T}{\partial \mathbf{u}}\frac{\partial C_j}{\partial \mathbf{z}}\right)\mathbf{H}^{-T}, \tag{2.19}$$

which gives a precise relationship between the uncertainty of the image correspondences $\mathbf{R_u}$ and the uncertainty of the depth and motion estimates $\mathbf{R_z}$.

Substituting our cost function from (2.9), we get

$$\frac{\partial C_i}{\partial \mathbf{z}} = \begin{cases} A_{\bar{i}p}, & i \text{ odd} \\ A_{\bar{i}q}, & i \text{ even} \end{cases}, \tag{2.20}$$

as a 1 × (N+3)-dimensional vector and

$$\begin{aligned}
\frac{\partial C_i}{\partial \mathbf{u}} &= [\tfrac{\partial C_i}{\partial p_1} \quad \tfrac{\partial C_i}{\partial q_1} \quad \dots \quad \tfrac{\partial C_i}{\partial p_N} \quad \tfrac{\partial C_i}{\partial q_N}], \\
&= \mathbf{I}_i(2N),
\end{aligned} \tag{2.21}$$

as a 1 × 2N-dimensional array. Hence the Hessian from (2.18) becomes

$$\mathbf{H} = \sum_{\bar{i}=1}^{N}\left(A_{\bar{i}p}{}^T A_{\bar{i}p} + A_{\bar{i}q}{}^T A_{\bar{i}q}\right). \tag{2.22}$$

Assuming that the feature points as well as the components of the motion vector at each feature point are independent of each other, $\mathbf{R_u} = \mathrm{diag}[R_{u\bar{i}p}, R_{u\bar{i}q}]_{\bar{i}=1,\dots N}$.

(Note that this condition is weaker than the one required to prove the optimality of the least squares criterion according to the Gauss-Markov theorem [37].) Then we can obtain a simpler relationship for the error covariances in (2.19):

$$\begin{aligned}\mathbf{R_z} &= \mathbf{H}^{-1}\left(\sum_i \frac{\partial C_i^T}{\partial \mathbf{z}}\frac{\partial C_i}{\partial \mathbf{u}}\mathbf{R_u}\frac{\partial C_i^T}{\partial \mathbf{u}}\frac{\partial C_i}{\partial \mathbf{z}}\right)\mathbf{H}^{-T} \\ &= \mathbf{H}^{-1}\left(\sum_{\bar{i}=1}^{N}\left(A_{\bar{i}p}{}^T A_{\bar{i}p}R_{u\bar{i}p} + A_{\bar{i}q}{}^T A_{\bar{i}q}R_{u\bar{i}q}\right)\right)\mathbf{H}^{-T}. \qquad (2.23)\end{aligned}$$

Equations (2.22) and (2.23) prove the statement of Theorem 1. If we make the even stronger assumption that the components of $\mathbf{R_u}$ are all identical (with variance r^2), i.e. $\mathbf{R_u} = r^2\mathbf{I}_{2N\text{X}2N}$, then (2.23) simplifies to

$$\begin{aligned}\mathbf{R_z} &= \mathbf{H}^{-1}(r^2\mathbf{H})\mathbf{H}^{-T} \\ &= r^2\mathbf{H}^{-1}. \qquad (2.24)\end{aligned}$$

2.4.2 Unknown FOE

When the focus of expansion in (2.2) is unknown, the linear form of (2.4) is lost. The unknown vector $\mathbf{z} = [\mathbf{h}, x_f, y_f, \mathbf{\Omega}]^T = [\mathbf{h}, \mathbf{m}]^T$ and the cost function is $C = \frac{1}{2}\sum_{i=1}^{n} C_i^2 = \frac{1}{2}\sum_{i=1}^{n} < u_i - \hat{u}_i(\mathbf{z}), u_i - \hat{u}_i(\mathbf{z}) >$, where $\hat{u}_i$ is the estimate of the 2D motion vector obtained by projecting the reconstructed scene according to (2.2). However, our method of deriving the error covariances using the implicit function theorem allows us to use the same method to derive the error covariances in this general case. The derivation presented above remains exactly the same except that we need to redefine the two vectors $A_{\bar{i}p}$ and $A_{\bar{i}q}$ as follows:

$$\begin{aligned}A_{\bar{i}p} &= [-(x_{\bar{i}} - x_f)\mathbf{I}_{\bar{i}}(N) \;\; | \;\; h_{\bar{i}} \;\; 0 \;\; -\mathbf{r}_{\bar{i}}], \\ &= [A_{\bar{i}ph} | A_{\bar{i}pm}], \\ A_{\bar{i}q} &= [-(y_{\bar{i}} - y_f)\mathbf{I}_{\bar{i}}(N) \;\; | \;\; 0 \;\; h_{\bar{i}} \;\; -\mathbf{s}_{\bar{i}}],\end{aligned}$$

$$= [A_{\bar{i}qh} | A_{\bar{i}qm}] \tag{2.25}$$

A very important distinction between the unknown FOE case and the known FOE case is that $A_{\bar{i}p}$ and $A_{\bar{i}q}$ are now functions of the inverse depth estimates h_i.

2.4.3 Structure of $\mathbf{R_z}$

The $\mathbf{R_z}$ thus obtained has an interesting structure as a result of our partitioning the vectors $A_{\bar{i}p}$ and $A_{\bar{i}q}$ into structure and motion components. From (2.22),

$$\mathbf{H} = \begin{bmatrix} \mathbf{H_h} & \mathbf{H_{hm}} \\ \mathbf{H_{hm}^T} & \mathbf{H_m} \end{bmatrix} \tag{2.26}$$

where

$$\begin{aligned} \mathbf{H_h} &= \sum_{\bar{i}=1}^{N} \left({A_{\bar{i}ph}}^T A_{\bar{i}ph} + {A_{\bar{i}qh}}^T A_{\bar{i}qh} \right) \\ \mathbf{H_{hm}^T} &= \sum_{\bar{i}=1}^{N} \left({A_{\bar{i}pm}}^T A_{\bar{i}ph} + {A_{\bar{i}qm}}^T A_{\bar{i}qh} \right) \\ \mathbf{H_m} &= \sum_{\bar{i}=1}^{N} \left({A_{\bar{i}pm}}^T A_{\bar{i}pm} + {A_{\bar{i}qm}}^T A_{\bar{i}qm} \right). \end{aligned} \tag{2.27}$$

$$\tag{2.28}$$

Thus

$$\mathbf{H_h} = \begin{bmatrix} (x_1 - x_f)^2 + (y_1 - y_f)^2 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & (x_N - x_f)^2 + (y_N - y_f)^2 \end{bmatrix} \tag{2.29}$$

and

$$\mathbf{H_m} = \sum_{\bar{i}=1}^{N} \left(\mathbf{r}_{\bar{i}}^T \mathbf{r}_{\bar{i}} + \mathbf{s}_{\bar{i}}^T \mathbf{s}_{\bar{i}} \right). \tag{2.30}$$

Then the inverse of $\mathbf{H}$ (assuming it exists) is [42]

$$\mathbf{H}^{-1} = \begin{bmatrix} \mathbf{Q} & \mathbf{S} \\ \mathbf{S}^T & \mathbf{G} \end{bmatrix} \tag{2.31}$$

with

$$\begin{aligned} \mathbf{Q} &= (\mathbf{H_h} - \mathbf{H_{hm}}\mathbf{H_m^{-1}}\mathbf{H_{hm}^T})^{-1} \\ \mathbf{G} &= (\mathbf{H_m} - \mathbf{H_{hm}^T}\mathbf{H_h^{-1}}\mathbf{H_{hm}})^{-1} \\ \mathbf{S} &= -\mathbf{Q}\mathbf{H_{hm}}\mathbf{H_m^{-1}}. \end{aligned} \tag{2.32}$$

From (2.23)

$$\begin{gathered} \textstyle\sum_{\bar{i}=1}^{N} \left({A_{\bar{i}p}}^T A_{\bar{i}p} R_{u\bar{i}p} + {A_{\bar{i}q}}^T A_{\bar{i}q} R_{u\bar{i}q} \right) = \\ \begin{bmatrix} \sum_{\bar{i}=1}^{N} \left({A_{\bar{i}ph}}^T A_{\bar{i}ph} R_{u\bar{i}p} + {A_{\bar{i}qh}}^T A_{\bar{i}qh} R_{u\bar{i}q} \right) & \sum_{\bar{i}=1}^{N} \left({A_{\bar{i}ph}}^T A_{\bar{i}pm} R_{u\bar{i}p} + {A_{\bar{i}qh}}^T A_{\bar{i}qm} R_{u\bar{i}q} \right) \\ \sum_{\bar{i}=1}^{N} \left(A_{\bar{i}pm} {A_{\bar{i}ph}}^T R_{u\bar{i}p} + A_{\bar{i}qm} {A_{\bar{i}qh}}^T R_{u\bar{i}q} \right) & \sum_{\bar{i}=1}^{N} \left({A_{\bar{i}pm}}^T A_{\bar{i}pm} R_{u\bar{i}p} + {A_{\bar{i}qm}}^T A_{\bar{i}qm} R_{u\bar{i}q} \right) \end{bmatrix} \\ \triangleq \begin{bmatrix} \mathbf{A} & \mathbf{C} \\ \mathbf{C}^T & \mathbf{B} \end{bmatrix} \end{gathered} \tag{2.33}$$

with

$$\textstyle\sum_{\bar{i}=1}^{N} \left({A_{\bar{i}ph}}^T A_{\bar{i}ph} R_{u\bar{i}p} + {A_{\bar{i}qh}}^T A_{\bar{i}qh} R_{u\bar{i}q} \right) =$$

$$\begin{bmatrix} (x_1 - x_f)^2 \sigma_{p1}^2 + (y_1 - y_f)^2 \sigma_{q1}^2 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & (x_N - x_f)^2 \sigma_{pN}^2 + (y_N - y_f)^2 \sigma_{qN}^2 \end{bmatrix}, \tag{2.34}$$

where $\sigma_{p\bar{i}}^2$ and $\sigma_{q\bar{i}}^2$ are the variances of the p and q motion components for the i-th feature point (i.e. $R_{u\bar{i}p} = \sigma_{p\bar{i}}^2$ and $R_{u\bar{i}q} = \sigma_{q\bar{i}}^2$). Then substituting (2.29) and (2.33) into (2.23), we obtain a partition for $\mathbf{R_z}$ as

$$\mathbf{R_z} = \begin{bmatrix} \mathbf{R_h} & \mathbf{R_{hm}} \\ \mathbf{R_{hm}^T} & \mathbf{R_m} \end{bmatrix} \tag{2.35}$$

$$= \begin{bmatrix} \mathbf{Q} & \mathbf{S} \\ \mathbf{S}^T & \mathbf{G} \end{bmatrix} \begin{bmatrix} \mathbf{A} & \mathbf{C} \\ \mathbf{C}^T & \mathbf{B} \end{bmatrix} \begin{bmatrix} \mathbf{Q} & \mathbf{S} \\ \mathbf{S}^T & \mathbf{G} \end{bmatrix}^T \tag{2.36}$$

Under the simplifying assumptions of equation (2.24), the partition of $\mathbf{R_z}$ can be obtained from the partition of $\mathbf{H}$ directly. Thus

$$\mathbf{R_h} = r^2 \begin{bmatrix} \mathbf{Q} & \mathbf{S} \\ \mathbf{S}^T & \mathbf{G} \end{bmatrix} \tag{2.37}$$

This is precisely the expression for the covariance and Cramer-Rao lower bound (CRLB) derived in [23] under an IID Gaussian noise assumption. This should be the case since the least squares technique is optimal under these conditions (the Gauss-Markov theorem [37]).

Estimating the Covariance of the Feature Points: The covariance of the feature points is in principle a function of the tracking algorithm, its parameters and the image intensity function in the neighborhood of the tracked points. We try to estimate the covariance of the feature points due to measurement errors caused primarily due to localization of the points. We use the standard method for estimating the error covariance using the inverse of the Hessian matrix of the second partial derivatives of the intensity along the x and y axes [31]. If $\mathbf{x} = [u(i,j), v(i,j)]^T$ represents the motion estimate in the x and y directions respectively at a point (i,j), then the error covariance at that point can be estimated by the inverse of the Hessian matrix as

$$\mathbf{R_u} = \begin{bmatrix} \frac{\partial^2 I(i,j)}{\partial x^2} & \frac{\partial^2 I(i,j)}{\partial x \partial y} \\ \frac{\partial^2 I(i,j)}{\partial x \partial y} & \frac{\partial^2 I(i,j)}{\partial y^2} \end{bmatrix}^{-1}, \tag{2.38}$$

where $I(i,j)$ is the intensity at the point (i,j).

2.5 Performance Analysis for Multiple Frames

In this section, we extend our covariance computation results to multiple frames and obtain an expression for the error as a function of the number of frames in the video sequence. We will discuss the importance of such a criterion in automatically evaluating the quality of a reconstruction.

2.5.1 Error Covariance Calculation

Let the two-frame inverse depth values for a particular feature point be denoted by $X^1, X^2, ..., X^N$ and let $\bar{X}$ be the mean. Now $E[\bar{X}] = \frac{1}{N}\sum_{i=1}^{N} E[X^i]$ and $\text{Cov}[\bar{X}] = E[(\bar{X} - X^*)^2] = E[\bar{X}^2] - E[\bar{X}]^2$.

$$\begin{aligned} E[\bar{X}]^2 &= \left(E\left[\frac{1}{N}\sum_{i=1}^{N} X^i\right]\right)^2 \\ &= \frac{1}{N^2}\left(\sum_{i=1}^{N} E[X^i]\right)^2 \\ &= \frac{1}{N^2}\left[\sum_{i=1}^{N} E[X^i]^2 + \sum_{i=1}^{N}\sum_{j=1}^{N} E[X^i]E[X^j]\right], i \neq j \end{aligned} \tag{2.39}$$

and

$$\begin{aligned} E[\bar{X}^2] &= E\left[(\frac{1}{N}\sum_{i=1}^{N} X^i)^2\right] \\ &= \frac{1}{N^2}E\left[\sum_{i=1}^{N} X^{i2} + \sum_{i=1}^{N}\sum_{j=1}^{N} X^i X^j\right], i \neq j \\ &= \frac{1}{N^2}\left[\sum_{i=1}^{N} E[X^{i2}] + \sum_{i=1}^{N}\sum_{j=1}^{N} E[X^i X^j]\right], i \neq j \end{aligned} \tag{2.40}$$

which yields the expression for the covariance of the estimator as

$$\text{Cov}[\bar{X}] = \frac{1}{N^2}\left[\sum_{i=1}^{N} \text{Cov}[X^i] + \sum_{i=1}^{N}\sum_{j=1}^{N}(E[X^i X^j] - E[X^i]E[X^j])\right] i \neq j. \tag{2.41}$$

The first summation, $\mathrm{Cov}[X^i]$, is the variance of the two-frame depth estimates obtained from $\mathbf{R}^i_{\mathbf{z}}$ in (2.24). Under the assumption of independence of the two-frame observations, the second term of (2.41) vanishes and we obtain a closed-form expression for the variance of the estimator for the N-frame SfM algorithm.The covariance of the estimate of the j-th feature point is

$$\mathrm{Cov}[\bar{X}_j] = \frac{1}{N^2}\left[\sum_{i=1}^{N} \mathbf{R}^i_{\mathbf{h}}(j,j)\right] \tag{2.42}$$

where $\mathbf{R}^i_{\mathbf{h}}(j,j)$ is the j-th diagonal term obtained from (2.36) for the i-th and $(i+1)$-st frames. Under the assumption of IID Gaussian noise of [23] for the two-frame algorithm, (2.42) simplifies to the following form:

$$\begin{aligned} \mathrm{Cov}[\bar{X}_j] &= \frac{1}{N^2}\left[\sum_{i=1}^{N} r^{i^2}\mathbf{Q}^i(j,j)\right], \\ &= \frac{1}{N^2}\left[\sum_{i=1}^{N} r^{i^2}(\mathbf{H}^i_{\mathbf{h}} - \mathbf{H}^i_{\mathbf{hm}}\mathbf{H}^{i^{-1}}_{\mathbf{m}}\mathbf{H}^{iT}_{\mathbf{hm}})^{-1}\right] \end{aligned} \tag{2.43}$$

where the terms are defined in (2.31) and (2.32). The expressions are valid for both the known and unknown FOE cases with $A_{\bar{i}p}$ and $A_{\bar{i}q}$ appropriately defined.

The average distortion in the reconstruction over M feature points is

$$\begin{aligned} E_{M,N}[(\bar{X} - E[\bar{X}])^2] &= E_M[E_N[(\bar{X} - E[\bar{X}])^2 | \bar{X} = \bar{X}_j]] \\ &= \frac{1}{MN^2}\sum_{j=1}^{M}\sum_{i=1}^{N} \mathbf{R}^i_{\mathbf{h}}(j,j) \\ &= \frac{1}{MN^2}\sum_{i=1}^{N} \mathrm{trace}(\mathbf{R}^i_{\mathbf{h}}). \end{aligned} \tag{2.44}$$

2.5.2 Significance of Multi-frame Distortion

Figure 2.5 plots the covariance of the estimator for the inverse depth as a function of frame number using (2.42) and (2.44) for two video sequences used in our experiments. A few interesting observations regarding these curves can now be made.

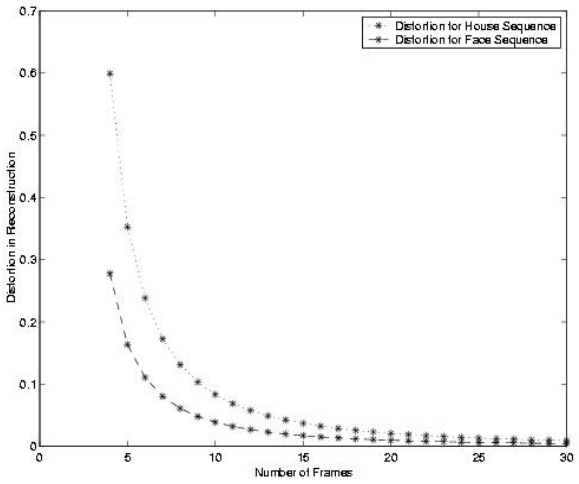


Figure 2.5: Plot of average distortion in reconstruction as a function of the number of frames for two different video sequences. The vertical axis is scaled down by a factor of 10^3.

- Given a particular tolerable level of distortion, each of these curves specifies the minimum number of frames necessary to achieve that level of distortion.

- The errors in SfM are due to a number of reasons, the effects of which are impossible to quantify separately. These curves give a compact representation for understanding the effects of these various sources of errors on the final estimate.

- The curve identifies an operating point of a MFSfM algorithm as a trade-off between tolerable reconstruction error and the computational cost of considering more frames.

- The curves depend on the covariance of the image correspondences only, if the FOE is known. In situations where the FOE does not change appreciably over the image sequence of interest, it is possible to plot these curves after the first pair of frames itself (after estimating the FOE).

- Though the average distortion over all features is plotted here, the curves can also be obtained for each individual feature point using (2.42). Since the uncertainty of the depth estimates is a function of the feature point (since the variance of the image correspondences will depend on the particular feature), the curves can be used to identify points which are more prone to reconstruction errors and thus would require greater numbers of frames to achieve a tolerable distortion.

- The nature of the plots for the unknown FOE case will remain the same, with $A_{\bar{i}p}$ and $A_{\bar{i}q}$ defined appropriately as in (2.25) (and now depending on h). However, they can no longer be computed without first estimating h. Hence the distortion in (2.44) needs to be estimated as the algorithm progresses.

2.6 Conclusions

In this chapter, we have presented our method for computing the error covariance of the depth and motion estimates as a function of the error covariance in tracking the feature points. We also showed how the results can be extended to consider multiple frames in the video sequence. These results provide a quantitative understanding of the quality of the reconstruction as a function of the quality of the video sequence, embedding in them the effects of several factors like lighting, camera distortion, algorithmic shortcomings, etc. In a later chapter, we will show how to use these mathematical results to obtain accurate and robust 3D reconstructions from a video sequence.

Chapter 3

Bias in Structure Estimate

3.1 Introduction

In the previous chapter, we dealt with the question of estimating the error covariance in the depth and motion estimates as a function of the error in tracking the feature points. A different source of error which has not received much attention in the computer vision community, but has been noted by psycho-physicists, is the fact "that it is hard to explain ... the existence of systematic biases in observers' magnitude estimation of perceived depth" [43]. In this chapter, we prove that the depth estimate is statistically biased, derive a precise expression for it, and hypothesize that our mathematical analysis supports many of the experimental observations. Many structure from motion (SfM) algorithms that reconstruct a scene from a video sequence pose the problem in a linear least squares framework $Ax = b$. It is a well-known fact that the least squares estimate is biased if the system matrix A is noisy. In SfM, the matrix A contains the image coordinates, which are always difficult to obtain precisely; thus it is expected that the structure and motion estimates in such a formulation of the problem would be biased. Some authors, notably Weng et al. [20], Daniilidis and

Spetsakis [44] and Kanatani [45], have proved that there exists a bias in translation and rotation estimates from stereo. Using the matrix perturbation theorem [42], Kanatani has derived expressions for the bias in the rotation and translational angles. Yet, to the best of our knowledge, there has been no attempt to compute the bias in the depth reconstruction from monocular video or to analyze the effects of different motion parameters on it. We show that even with a perfect motion estimate, the depth estimate is statistically biased. Existing results on the minimum achievable variance of the estimator are extended by deriving a *generalized* Cramer-Rao lower bound. Through simulations, we demonstrate the effects of camera motion parameters on the bias and give numerical examples to highlight the importance of compensating for it.

This chapter is organized as follows. In Section 3.2, we compute an expression for the bias term. We also analyze the relationship of our results to the human visual system. Section 3.4 extends the existing results on the minimum variance of the structure estimate and derives a generalized CRLB after incorporating the effect of the bias. We analyze how the bias is affected by various physical parameters, like the camera motion and the geometrical indeterminacies. Finally, in Section 3.5 we present our experimental results and through simulations, compare the reconstructions obtained from a bias-compensated SfM algorithm with those obtained from one that ignores the bias. Also, the effects of the different motion parameters on the bias and the generalized CRLB are studied.

3.2 Bias in Depth Reconstruction

3.2.1 Problem Formulation and Result

Recall equation (2.4) of Chapter 2, $\mathbf{Bz} = \mathbf{u}$. In situations where the FOE is known or can be estimated by other means [8], the solution vector $\mathbf{x}$ can be obtained using a standard linear least squares approach. The estimate is $\hat{\mathbf{z}} = (\mathbf{B}'\mathbf{B})^{-1}\mathbf{B}'\mathbf{u}$ [36].

The system matrix $\mathbf{B}$ depends upon the image coordinates $\{x_i, y_i\}$ and the FOE (x_f, y_f). The positions of the image coordinates will always have measurement errors, which are sometimes quite large. Thus the matrix $\mathbf{B}$ may have significant noise terms. It is well known that the least-squares solution to a linear system of the form $Ax = b$ with errors in the system matrix A is biased [46], [47]. Thus we can expect that the solution to the SfM problem will also be biased.

In this chapter, we obtain an approximate expression for the bias in the estimate and analyze the significance of this estimation error on the reconstruction. In order to keep the algebraic manipulations tractable, we assume that we know the camera motion $\mathbf{\Omega}$. Then (2.4) can be expressed in the form $\mathbf{b} = \mathbf{Ah}$, with $\mathbf{A} \triangleq \mathbf{P}$ and $\mathbf{b} \triangleq [p_1 - \mathbf{r}_1\Omega', q_1 - \mathbf{s}_1\Omega', ..., p_N - \mathbf{r}_N\Omega', q_N - \mathbf{s}_N\Omega']'$, and the least squares solution (if $\mathbf{A}$ is known exactly) is $\hat{\mathbf{h}} = (\mathbf{A}'\mathbf{A})^{-1}\mathbf{A}'\mathbf{b}$. Since we assume that the FOE is also known, we consider the case where the camera motion is known. The bias term will be computed under this assumption of known camera motion. However, our knowledge of the camera motion may not be perfect and we will also need to consider the errors in its value. Even though the known camera motion assumption is introduced for simplicity of mathematical manipu-

lation, it also has the advantage that we can analyze the effect of the bias under conditions of known camera motion. This establishes the fact that the bias occurs due to lack of precision in obtaining the motion estimates from the video sequence, and not because of errors in the camera motion.

We now state the main result of this chapter, which is a precise expression for the bias in 3D reconstruction from optical flow.

Theorem 2 *Let the errors in the different variables be expressed as follows:* $x_i = \bar{x}_i + \delta x_i, y_i = \bar{y}_i + \delta y_i, p_i = \bar{p}_i + \delta p_i, q_i = \bar{q}_i + \delta q_i$, *where the over-bars represent the unknown true values of the parameters and the observed (measured) values are represented without the over-bars. For convenience, let us define* [1]

$$\begin{aligned}
\mathbf{M} \stackrel{\Delta}{=} \mathbf{A}'\mathbf{A} &= \begin{bmatrix} (x_1 - x_f)^2 + (y_1 - y_f)^2 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & (x_N - x_f)^2 + (y_N - y_f)^2 \end{bmatrix} \\
&\stackrel{\Delta}{=} \begin{bmatrix} m_{11} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & m_{NN} \end{bmatrix}, \\
&= diag\left[m_{ii}\right]_{i=1,\ldots,N}
\end{aligned} \tag{3.1}$$

$$\mathbf{v} \stackrel{\Delta}{=} \mathbf{A}'\mathbf{b} \stackrel{\Delta}{=} \begin{bmatrix} (x_1 - x_f)v_{p1} + (y_1 - y_f)v_{q1} \\ \vdots \\ (x_N - x_f)v_{pN} + (y_N - y_f)v_{qN} \end{bmatrix}$$

[1] *Diagonal matrices will be very frequently used in the calculations. A diagonal matrix of size* $N \times N$ *consisting of the diagonal terms* $a_1, ..., a_N$ *will be represented as* $diag\left[a_1, ..., a_N\right]$ *or* $diag\left[a_i\right]_{i=1,\ldots,N}$.

$$\triangleq \begin{bmatrix} v_1 \\ \vdots \\ v_N \end{bmatrix}, \tag{3.2}$$

where $v_{pi} = p_i - \mathbf{r}_i\Omega'$ *and* $v_{qi} = q_i - \mathbf{s}_i\Omega'$.

If σ_f^2 *is the variance of the measurement error of the FOE (i.e.* $E[\delta {x_f}^2] = E[\delta {y_f}^2] = \sigma_f^2$*) and* $\sigma_i^2 = E[\delta {x_i}^2] = E[\delta {y_i}^2]$ *is the variance in the image coordinate measurements, then under the assumptions of the above formulation, the bias in the inverse depth estimate of the* i^{th} *feature point is given by*

$$\begin{aligned}
[Bias]_i &= b(\hat{\mathbf{h}}_i) = \frac{v_i}{m_{ii}^2}\left(\sigma_f^2 + 2\sigma_i^2\right) \\
&+ \frac{\sigma_i^2}{m_{ii}^2}\left[(x_i - x_f)^2 r_{ix} + (y_i - y_f)^2 s_{iy}\right]\Omega' \\
&+ \frac{\sigma_i^2}{m_{ii}^2}\left[(x_i - x_f)(y_i - y_f)(s_{ix} + r_{iy})\right]\Omega' \\
&+ \frac{\sigma_i^2}{m_{ii}}\left[(x_i - x_f)\omega_y - (y_i - y_f)\omega_x - (r_{ix} + s_{iy})\Omega'\right],
\end{aligned} \tag{3.3}$$

where f_{ix} *represents the derivative of a function* f_i *with respect to* x.

3.2.2 Computation of Bias Term

We now prove the above result. Expanding $\hat{\mathbf{h}}$ in a Taylor series around the true value $\bar{\mathbf{h}}$, (i.e. the noise $N = 0$, which means that the deviations from the true values are zero) and assuming the mean deviation in that region to be zero (i.e. $E[\delta x_i] = E[\delta y_i] = E[\delta x_f] = E[\delta y_f] = 0$) and all the components $\delta x_i, \delta y_i, \delta x_f, \delta y_f$ to be mutually uncorrelated, we can express

$$\begin{aligned}
E[\hat{\mathbf{h}}] \approx\ & \bar{\mathbf{h}} + \sum_{i=1}^{N}\left[\frac{\partial^2\hat{\mathbf{h}}}{\partial\delta {x_i}^2}E[\frac{\delta {x_i}^2}{2}] + \frac{\partial^2\hat{\mathbf{h}}}{\partial\delta {y_i}^2}E[\frac{\delta {y_i}^2}{2}]\right] \\
&+ \frac{\partial^2\hat{\mathbf{h}}}{\partial\delta {x_f}^2}E[\frac{\delta {x_f}^2}{2}] + \frac{\partial^2\hat{\mathbf{h}}}{\partial\delta {y_f}^2}E[\frac{\delta {y_f}^2}{2}],
\end{aligned} \tag{3.4}$$

where all the partials are computed at $N = 0$. In the absence of any measurement noise, the expected value of the estimate obtained from the least squares solution should equal the true value $\bar{\mathbf{h}}$. However, since there exist errors in the measurement model, the estimate is biased and the sum of the last four terms on the right hand side of (3.4) represents the total bias in the estimate. Since the bias is calculated with known camera motion, it shows that *even if the camera motion is known perfectly, the depth estimate is statistically biased.* Thus the bias does not occur due to errors in the camera motion estimates and cannot be compensated by adjusting it.

In order to calculate the bias, we need to compute the derivatives in (3.4). We can compute all the derivatives using the fact that for an arbitrary matrix Q [48],

$$-\frac{\partial Q^{-1}}{\partial x} = Q^{-1}\frac{\partial Q}{\partial x}Q^{-1}. \tag{3.5}$$

To keep the computation simple, we will make the assumption that (p_i, q_i) do not depend on $(\delta x_i, \delta y_i)$ [2]. Note that $\frac{\partial \mathbf{M}}{\partial \delta p_i} = \frac{\partial \mathbf{M}}{\partial \delta q_i} = \frac{\partial \mathbf{M}}{\partial \delta \omega_x} = \frac{\partial \mathbf{M}}{\partial \delta \omega_y} = \frac{\partial \mathbf{M}}{\partial \delta \omega_z} = 0$, since $\mathbf{M}$ does not involve these variables. Also, $\frac{\partial^2 \mathbf{v}}{\partial \delta {p_i}^2} = \frac{\partial^2 \mathbf{v}}{\partial \delta {q_i}^2} = \frac{\partial^2 \mathbf{v}}{\partial \delta {x_f}^2} = \frac{\partial^2 \mathbf{v}}{\partial \delta {y_f}^2} = \frac{\partial^2 \mathbf{v}}{\partial \delta {\omega_x}^2} = \frac{\partial^2 \mathbf{v}}{\partial \delta {\omega_y}^2} = \frac{\partial^2 \mathbf{v}}{\partial \delta {\omega_z}^2} = 0$, since $\mathbf{v}$ is a linear function of these variables. Since $\hat{\mathbf{h}} = \mathbf{M}^{-1}\mathbf{v}$, we can now compute the following terms:

$$\begin{aligned}
\frac{\partial \hat{\mathbf{h}}}{\partial \delta p_i} &= -\mathbf{M}^{-1}\frac{\partial \mathbf{M}}{\partial \delta p_i}\mathbf{M}^{-1}\mathbf{v} + \mathbf{M}^{-1}\frac{\partial \mathbf{v}}{\partial \delta p_i} = \mathbf{M}^{-1}\frac{\partial \mathbf{v}}{\partial \delta p_i}, \\
\frac{\partial^2 \hat{\mathbf{h}}}{\partial \delta {p_i}^2} &= -\mathbf{M}^{-1}\frac{\partial \mathbf{M}}{\partial \delta p_i}\mathbf{M}^{-1}\frac{\partial \mathbf{v}}{\partial \delta p_i} + \mathbf{M}^{-1}\frac{\partial^2 \mathbf{v}}{\partial \delta {p_i}^2} = 0, \\
\frac{\partial \hat{\mathbf{h}}}{\partial \delta \omega_x} &= -\mathbf{M}^{-1}\frac{\partial \mathbf{M}}{\partial \delta \omega_x}\mathbf{M}^{-1}\mathbf{v} + \mathbf{M}^{-1}\frac{\partial \mathbf{v}}{\partial \delta \omega_x} = \mathbf{M}^{-1}\frac{\partial \mathbf{v}}{\partial \delta \omega_x}, \\
\frac{\partial^2 \hat{\mathbf{h}}}{\partial \delta {\omega_x}^2} &= -\mathbf{M}^{-1}\frac{\partial \mathbf{M}}{\partial \delta \omega_x}\mathbf{M}^{-1}\frac{\partial \mathbf{v}}{\partial \delta \omega_x} + \mathbf{M}^{-1}\frac{\partial^2 \mathbf{v}}{\partial \delta {\omega_x}^2} = 0,
\end{aligned} \tag{3.6}$$

[2] Since $p_i \approx \bar{x}_{i+1} + \delta x_{i+1} - \bar{x}_i - \delta x_i$ and $q_i \approx \bar{y}_{i+1} + \delta y_{i+1} - \bar{y}_i - \delta y_i$, this amounts to assuming that the motion field does not depend on the deviations of feature positions.

Following exactly similar steps, we get $\frac{\partial^2 \hat{\mathbf{h}}}{\partial \delta \omega_y{}^2} = \frac{\partial^2 \hat{\mathbf{h}}}{\partial \delta \omega_z{}^2} = 0$. Hence (3.4) simplifies to

$$\begin{aligned} E[\hat{\mathbf{h}}] \approx \bar{\mathbf{h}} + \sum_{i=1}^{N} \left[\frac{\partial^2 \hat{\mathbf{h}}}{\partial \delta x_i{}^2} E[\frac{\delta x_i{}^2}{2}] + \frac{\partial^2 \hat{\mathbf{h}}}{\partial \delta y_i{}^2} E[\frac{\delta y_i{}^2}{2}] \right] \\ + \frac{\partial^2 \hat{\mathbf{h}}}{\partial \delta x_f{}^2} E[\frac{\delta x_f{}^2}{2}] + \frac{\partial^2 \hat{\mathbf{h}}}{\partial \delta y_f{}^2} E[\frac{\delta y_f{}^2}{2}], \end{aligned} \tag{3.7}$$

which is to be expected since the system matrix $\mathbf{A}$ depends only on $\{x_i, y_i\}$ and (x_f, y_f). The derivatives with respect to the image coordinates $(\delta x_i, \delta y_i)$ are

$$\begin{aligned} \frac{\partial \hat{\mathbf{h}}}{\partial \delta x_i} &= -\mathbf{M}^{-1} \frac{\partial \mathbf{M}}{\partial \delta x_i} \mathbf{M}^{-1} \mathbf{v} + \mathbf{M}^{-1} \frac{\partial \mathbf{v}}{\partial \delta x_i} \\ \frac{\partial^2 \hat{\mathbf{h}}}{\partial \delta x_i{}^2} &= 2\mathbf{M}^{-1} \frac{\partial \mathbf{M}}{\partial \delta x_i} \mathbf{M}^{-1} \frac{\partial \mathbf{M}}{\partial \delta x_i} \mathbf{M}^{-1} \mathbf{v} - \mathbf{M}^{-1} \frac{\partial^2 \mathbf{M}}{\partial \delta x_i{}^2} \mathbf{M}^{-1} \mathbf{v} \\ &\quad -\mathbf{M}^{-1} \frac{\partial \mathbf{M}}{\partial \delta x_i} \mathbf{M}^{-1} \frac{\partial \mathbf{v}}{\partial \delta x_i} + \mathbf{M}^{-1} \frac{\partial^2 \mathbf{v}}{\partial \delta x_i{}^2} \\ \frac{\partial^2 \hat{\mathbf{h}}}{\partial \delta y_i{}^2} &= 2\mathbf{M}^{-1} \frac{\partial \mathbf{M}}{\partial \delta y_i} \mathbf{M}^{-1} \frac{\partial \mathbf{M}}{\partial \delta y_i} \mathbf{M}^{-1} \mathbf{v} - \mathbf{M}^{-1} \frac{\partial^2 \mathbf{M}}{\partial \delta y_i{}^2} \mathbf{M}^{-1} \mathbf{v} \\ &\quad -\mathbf{M}^{-1} \frac{\partial \mathbf{M}}{\partial \delta y_i} \mathbf{M}^{-1} \frac{\partial \mathbf{v}}{\partial \delta y_i} + \mathbf{M}^{-1} \frac{\partial^2 \mathbf{v}}{\partial \delta y_i{}^2}. \end{aligned} \tag{3.8}$$

The derivatives of $\mathbf{M}$ with respect to $\{\delta x_i, \delta y_i\}$ are

$$\begin{aligned} \frac{\partial \mathbf{M}}{\partial \delta x_i} &= \text{diag}\,[0, ..., 0, 2(x_i - x_f), 0, ..., 0] \\ \frac{\partial^2 \mathbf{M}}{\partial \delta x_i{}^2} &= \text{diag}\,[0, ..., 0, 2, 0, ..., 0] \\ \frac{\partial \mathbf{M}}{\partial \delta y_i} &= \text{diag}\,[0, ..., 0, 2(y_i - y_f), 0, ..., 0] \\ \frac{\partial^2 \mathbf{M}}{\partial \delta y_i{}^2} &= \text{diag}\,[0, ..., 0, 2, 0, ..., 0] \\ \mathbf{M}^{-1} \frac{\partial \mathbf{M}}{\partial \delta x_i} &= \text{diag}\left[0, ..., 0, \frac{2(x_i - x_f)}{m_{ii}}, 0, ..., 0\right] \end{aligned} \tag{3.9}$$

Similar expressions can be obtained for the derivatives with respect to δy_i, by substituting y for x.

Next, we compute the derivatives with respect to deviations from the FOE $(\delta x_f, \delta y_f)$.

$$\frac{\partial \hat{\mathbf{h}}}{\partial \delta x_f} = -\mathbf{M}^{-1} \frac{\partial \mathbf{M}}{\partial \delta x_f} \mathbf{M}^{-1} \mathbf{v} + \mathbf{M}^{-1} \frac{\partial \mathbf{v}}{\partial \delta x_f}$$

$$
\begin{aligned}
\frac{\partial^2 \hat{\mathbf{h}}}{\partial \delta {x_f}^2} &= 2\mathbf{M}^{-1}\frac{\partial \mathbf{M}}{\partial \delta x_f}\mathbf{M}^{-1}\frac{\partial \mathbf{M}}{\partial \delta x_f}\mathbf{M}^{-1}\mathbf{v} - \mathbf{M}^{-1}\frac{\partial^2 \mathbf{M}}{\partial \delta {x_f}^2}\mathbf{M}^{-1}\mathbf{v} \\
&\quad -\mathbf{M}^{-1}\frac{\partial \mathbf{M}}{\partial \delta x_f}\mathbf{M}^{-1}\frac{\partial \mathbf{v}}{\partial \delta x_f} + \mathbf{M}^{-1}\frac{\partial^2 \mathbf{v}}{\partial \delta {x_f}^2} \\
\frac{\partial^2 \hat{\mathbf{h}}}{\partial \delta {y_f}^2} &= 2\mathbf{M}^{-1}\frac{\partial \mathbf{M}}{\partial \delta y_f}\mathbf{M}^{-1}\frac{\partial \mathbf{M}}{\partial \delta y_f}\mathbf{M}^{-1}\mathbf{v} - \mathbf{M}^{-1}\frac{\partial^2 \mathbf{M}}{\partial \delta {y_f}^2}\mathbf{M}^{-1}\mathbf{v} \\
&\quad -\mathbf{M}^{-1}\frac{\partial \mathbf{M}}{\partial \delta y_f}\mathbf{M}^{-1}\frac{\partial \mathbf{v}}{\partial \delta y_f} + \mathbf{M}^{-1}\frac{\partial^2 \mathbf{v}}{\partial \delta {y_f}^2}. \qquad (3.10)
\end{aligned}
$$

Computing the derivatives of $\mathbf{M}$ with respect to $(\delta x_f, \delta y_f)$, we get

$$
\begin{aligned}
\frac{\partial \mathbf{M}}{\partial \delta x_f} &= \mathrm{diag}\left[-2(x_i - x_f)\right]_{i=1,\ldots,N}, \\
\frac{\partial^2 \mathbf{M}}{\partial \delta {x_f}^2} &= 2I_{N\times N} \\
\frac{\partial \mathbf{M}}{\partial \delta y_f} &= \mathrm{diag}\left[-2(y_i - y_f)\right]_{i=1,\ldots,N}, \\
\frac{\partial^2 \mathbf{M}}{\partial \delta {y_f}^2} &= 2I_{N\times N} \\
\frac{\partial \mathbf{v}}{\partial \delta x_f} &= [-v_{p1}, ..., -v_{pN}]', \\
\frac{\partial \mathbf{v}}{\partial \delta y_f} &= [-v_{q1}, ..., -v_{qN}]', \\
\mathbf{M}^{-1}\frac{\partial \mathbf{M}}{\partial \delta x_f} &= \mathrm{diag}\left[\frac{-2(x_i - x_f)}{m_{ii}}\right]_{i=1,\ldots,N}, \\
\mathbf{M}^{-1}\frac{\partial \mathbf{v}}{\partial \delta x_f} &= \left[\frac{-v_{p1}}{m_{11}}, ..., \frac{-v_{pN}}{m_{NN}}\right]'. \qquad (3.11)
\end{aligned}
$$

Using (3.11), we can now compute some of the terms in the expression for $\frac{\partial^2 \hat{\mathbf{h}}}{\partial \delta {x_f}^2}$ and $\frac{\partial^2 \hat{\mathbf{h}}}{\partial \delta {y_f}^2}$ in (3.10):

$$
\begin{aligned}
\mathbf{M}^{-1}\frac{\partial \mathbf{M}}{\partial \delta x_f}\mathbf{M}^{-1}\frac{\partial \mathbf{M}}{\partial \delta x_f}\mathbf{M}^{-1}\mathbf{v} &= \left[\frac{4(x_1 - x_f)^2 v_1}{m_{11}^3}, ..., \frac{4(x_N - x_f)^2 v_N}{m_{NN}^3}\right]', \\
\mathbf{M}^{-1}\frac{\partial^2 \mathbf{M}}{\partial \delta {x_f}^2}\mathbf{M}^{-1}\mathbf{v} &= \left[\frac{2v_1}{m_{11}^2}, ..., \frac{2v_N}{m_{NN}^2}\right]', \\
\mathbf{M}^{-1}\frac{\partial \mathbf{M}}{\partial \delta x_f}\mathbf{M}^{-1}\frac{\partial \mathbf{v}}{\partial \delta x_f} &= \mathrm{diag}\left[\frac{2(x_1 - x_f)v_{p1}}{m_{11}^2}, ..., \frac{2(x_N - x_f)v_{pN}}{m_{NN}^2}\right]'. \qquad (3.12)
\end{aligned}
$$

Similar expressions can be obtained for the partial derivatives with respect to δy_f. Substituting the above expressions in (3.10), we get the expression for one

of the bias terms in (3.7) as

$$\frac{\partial^2 \hat{\mathbf{h}}}{\partial \delta x_f{}^2} + \frac{\partial^2 \hat{\mathbf{h}}}{\partial \delta y_f{}^2} = \left[\frac{2v_1}{m_{11}^2}, ..., \frac{2v_N}{m_{NN}^2}\right]'. \tag{3.13}$$

Computing the partial derivatives with respect to $(\delta x_i, \delta y_i)$ requires more work. Let us denote the partial of a function $f_i = f(x_i, y_i)$ with respect to x_i by f_{ix} and the second partials by $f_{ixx}, f_{ixy}, f_{iyy}$. Thus $r_{ix} = [y_i, -2x_i, 0]\,, r_{iy} = [x_i, 0, 1]\,, s_{ix} = [0, -y_i, 1]\,, s_{iy} = [2y_i, -x_i, 0]\,, r_{ixx} = [0, -2, 0]\,, r_{iyy} = [0, 0, 0]\,, s_{ixx} = [0, 0, 0]\,, s_{iyy} = [2, 0, 0]$[3]. The derivatives of $\mathbf{v}$ with respect to $(\delta x_i, \delta y_i)$ are

$$\begin{aligned}
\frac{\partial v_i}{\partial \delta x_i} &= v_{pi} + (x_i - x_f)\frac{\partial v_{pi}}{\partial \delta x_i} + (y_i - y_f)\frac{\partial v_{qi}}{\partial \delta x_i} \\
&= v_{pi} - (x_i - x_f) r_{ix}\Omega' - (y_i - y_f) s_{ix}\Omega' \;\triangleq\; v_{ix} \\
\frac{\partial v_i}{\partial \delta y_i} &= v_{qi} + (x_i - x_f)\frac{\partial v_{pi}}{\partial \delta y_i} + (y_i - y_f)\frac{\partial v_{qi}}{\partial \delta y_i} \\
&= v_{pi} - (x_i - x_f) r_{iy}\Omega' - (y_i - y_f) s_{iy}\Omega' \;\triangleq\; v_{iy} \\
\frac{\partial^2 v_i}{\partial \delta x_i{}^2} &= -2r_{ix}\Omega' + 2(x_i - x_f)\omega_y \;\triangleq\; v_{ixx} \\
\frac{\partial^2 v_i}{\partial \delta y_i{}^2} &= -2s_{iy}\Omega' - 2(y_i - y_f)\omega_x \;\triangleq\; v_{iyy}.
\end{aligned} \tag{3.14}$$

Using (3.8) and (3.14), we now compute each of the terms in $(\frac{\partial^2 \hat{\mathbf{h}}}{\partial \delta x_i{}^2}, \frac{\partial^2 \hat{\mathbf{h}}}{\partial \delta y_i{}^2})$ in (3.7):

$$\begin{aligned}
\mathbf{M}^{-1}\frac{\partial \mathbf{M}}{\partial \delta x_i}\mathbf{M}^{-1}\frac{\partial \mathbf{M}}{\partial \delta x_i}\mathbf{M}^{-1}\mathbf{v} &= \left[0, ..., 0, \frac{4(x_i - x_f)^2 v_i}{m_{ii}^3}, 0, ..., 0\right]' \\
\mathbf{M}^{-1}\frac{\partial^2 \mathbf{M}}{\partial \delta x_i{}^2}\mathbf{M}^{-1}\mathbf{v} &= \left[0, ..., 0, \frac{2v_i}{m_{ii}^2}, 0, ..., 0\right]' \\
\mathbf{M}^{-1}\frac{\partial \mathbf{M}}{\partial \delta x_i}\mathbf{M}^{-1}\frac{\partial \mathbf{v}}{\partial \delta x_i} &= \left[0, ..., 0, \frac{2(x_i - x_f)v_{ix}}{m_{ii}^2}, 0, ..., 0\right]' \\
\mathbf{M}^{-1}\frac{\partial \mathbf{M}}{\partial \delta y_i}\mathbf{M}^{-1}\frac{\partial \mathbf{v}}{\partial \delta y_i} &= \left[0, ..., 0, \frac{2(y_i - y_f)v_{iy}}{m_{ii}^2}, 0, ..., 0\right]'
\end{aligned}$$

[3]We are using the subscripts x, y for brevity of notation, but the derivatives are actually with respect to $\{\delta x_i, \delta y_i\}$.

$$\begin{aligned}
\mathbf{M}^{-1}\frac{\partial^2 \mathbf{v}}{\partial \delta {x_i}^2} &= \left[0, ..., 0, \frac{v_{ixx}}{m_{ii}}, 0, ..., 0\right]' \\
\mathbf{M}^{-1}\frac{\partial^2 \mathbf{v}}{\partial \delta {y_i}^2} &= \left[0, ..., 0, \frac{v_{iyy}}{m_{ii}}, 0, ..., 0\right]'
\end{aligned} \tag{3.15}$$

Then the expression for the i^{th} component of $\frac{\partial^2 \hat{\mathbf{h}}}{\partial \delta {x_i}^2} + \frac{\partial^2 \hat{\mathbf{h}}}{\partial \delta {y_i}^2}$ is

$$\begin{aligned}
\left[\frac{\partial^2 \hat{\mathbf{h}}}{\partial \delta {x_i}^2} + \frac{\partial^2 \hat{\mathbf{h}}}{\partial \delta {y_i}^2}\right]_i &= \frac{8[(x_i - x_f)^2 + (y_i - y_f)^2]v_i}{m_{ii}^3} - \frac{4v_i}{m_{ii}^2} \\
&\quad - \frac{2[(x_i - x_f)v_{ix} + (y_i - y_f)v_{iy}]}{m_{ii}^2} + \frac{v_{ixx} + v_{iyy}}{m_{ii}} \\
&= \frac{4v_i}{m_{ii}^2} + \frac{2}{m_{ii}^2}\left[(x_i - x_f)^2 r_{ix} + (y_i - y_f)^2 s_{iy}\right]\Omega' \\
&\quad + \frac{2}{m_{ii}^2}\left[(x_i - x_f)(y_i - y_f)(s_{ix} + r_{iy})\right]\Omega' \\
&\quad + \frac{2}{m_{ii}}\left[(x_i - x_f)\omega_y - (y_i - y_f)\omega_x - (r_{ix} + s_{iy})\Omega'\right]
\end{aligned} \tag{3.16}$$

At this point we can appreciate the need for the approximation in computing the bias in the structure parameters only. If we compute the bias for the entire solution vector $\mathbf{x}$, the matrix $\mathbf{M}$ would no longer be diagonal and its inverse would be even more complicated. Since many algorithms proceed by first computing the camera motion and then the structure, this approximation is reasonable. Moreover, it allows us to analyze the characteristics of the solution for the structure, even if the camera motion is known.

Substituting the expressions for the different partial derivatives obtained in (3.13) and (3.16) in (3.7) and evaluating them at $N = 0$, we can obtain an exact expression for the bias as stated in Theorem 2.

3.3 Analysis of the Bias

The fact that the depth estimate is statistically biased has implications for 3D reconstruction algorithms, as well as for our interpretation and analysis of the motion. We analyze below how the bias is affected by the geometric indeterminacies of SfM and discuss methods of compensating for it in 3D reconstruction algorithms. We also extend the two-frame result in Theorem 2 to multiple frames and study its links to the human visual system.

Effect of Scale Ambiguity It is well known that if (v, z) is a solution of (2.2), so is (sv, sz). This is known as the scale ambiguity in SfM [34]. Analysis of (3.3) shows that the bias in $\hat{\mathbf{h}}$ remains unchanged, since the FOE is not affected by the change of scale. However, since $h(x, y) = v_z/z(x, y)$, the bias in the scaled inverse depth, $\frac{1}{sz}$, would be $1/s$ times the bias in the inverse depth $\frac{1}{z}$.

Effect of Camera Motion Since the expression in (3.3) was derived under the assumption that Ω is known, we see that *even if the camera motion is known perfectly, the estimate of the inverse depth (and hence also the depth) is statistically biased.* Thus the assumption of known camera motion, under which (3.3) was derived, shows that this error is independent of the errors in camera motion estimation.

Bias Compensation Once the structure and motion estimates are obtained, the bias can be computed and subtracted out of the estimate. If $\hat{\mathbf{h}}_c = \hat{\mathbf{h}} - b(\hat{\mathbf{h}})$ is the bias-compensated estimate, then $E[\hat{\mathbf{h}}_c] = E[\hat{\mathbf{h}}] - b(\hat{\mathbf{h}}) = \bar{\mathbf{h}}$, thus leading to an unbiased estimate.

Bias and Total Least Squares (TLS) TLS has emerged as an alternative to

least squares since it is capable of handling errors in both the observations, b, and the system variables, A, in a linear system $Ax = b$ [49]. However, the TLS estimate is unbiased only if the error in estimating A is equal in variance to the error in estimating b. Such a condition would be very difficult to maintain in (2.4). Also, estimating the bias of a TLS estimate is extremely cumbersome. Also, as argued in [49], the covariance of an unbiased TLS estimate is larger than that of the LS estimate, in the first-order approximation as well as in simulations. Hence, there is no fundamental gain in choosing the TLS over the LS solution. Thus using the TLS criterion cannot be a solution to the problem of bias in the 3D estimate.

3.3.1 Bias in Multi-frame Reconstruction

Suppose now that we have L two-frame reconstructions for every consecutive pair of $(L+1)$ frames. Let $(\hat{\mathbf{h}}^1, ..., \hat{\mathbf{h}}^L)$ be the two-frame estimates aligned with respect to a particular frame of reference. Let the true value be $\bar{\mathbf{h}}$ and the bias in (3.3) be represented by $(b(\hat{\mathbf{h}}^1), ..., b(\hat{\mathbf{h}}^L))$, i.e. $E[\hat{\mathbf{h}}^i] = \bar{\mathbf{h}} + b(\hat{\mathbf{h}}^i)$. Assume that the estimates and the true value are with respect to a particular gauge $\mathcal{C}$ [32] (so that the problem of scale ambiguity does not arise). Then the least-squares estimate for the structure over all L observations is $\hat{\mathbf{h}} = \frac{1}{L}\sum_{i=1}^{L} \hat{\mathbf{h}}^i$. Taking expectations on both sides, we see that the bias in the multi-frame estimate is $b_{\mathcal{C}}(\hat{\mathbf{h}}) = \frac{1}{L}\sum_{i=1}^{L} b(\hat{\mathbf{h}}^i)$, where $b(\hat{\mathbf{h}}^i)$ is obtained from (2.2) for the i^{th} and $(i+1)^{\text{st}}$ frames.

3.3.2 Connection to the Human Visual System

No general theory exists which explains the performance of the human visual system. However, it is generally believed that our eyes receive a sequence of images and the data from a number of retinal images is combined to obtain a representation of the physical scene [50]. The early visual processing system extracts local image measurements from this sequence of images for use in further estimation processes. In our problem formulation, these measurements correspond to positions and gray values of image points. However, they can be derived only within a range of accuracy, i.e. there is noise in the measurements of the positions of the image points.

In our work, we do not attempt to model the specifics of the human visual system. Our work is concerned primarily with the limitations of the SfM equations as expressed in (2.2). However, there are certain similarities with the human visual system which can be used to explain certain visual phenomena. The inputs that we use (the positions of points over a sequence of images) are similar to those processed by the early visual system. This data is then used to interpret the structure of the scene from which the images were obtained. As we have shown, the structure estimation process is statistically biased. Thus our mathematical analysis is in accordance with the results obtained by experiments on the human visual system.

3.4 The Generalized CRLB for SfM

The CRLB for the two-frame SfM estimate has been computed by various researchers ([22], [23], [24], [27], [51],[52]). Usually the CRLB is computed as-

suming an unbiased estimate. In this section, we extend the existing results to account for the effect of the bias on the minimum variance of the reconstruction error and derive a generalized CRLB. First, we derive an expression for the Fisher information matrix (the inverse of the CRLB for an unbiased estimate) using an approach that highlights the importance of knowing the true values of the feature positions, followed by the derivation of the generalized CRLB taking into account the bias.

3.4.1 Computing the Fisher Information of Unbiased Structure and Motion Parameters

We assume that the true values of the feature positions $\{\bar{x}_i, \bar{y}_i\}$ and FOE $(\bar{x}_f, \bar{y}_f)$ are known. In oder not to make the notation more cumbersome, $\{x_i, y_i\}$ and (x_f, y_f) in this section will refer to the true values.

We assume a perspective camera model where the motion between successive frames is small enough to justify the equations in (2.2). We aim to compute the variance of the reconstruction error for the tracked feature points. Since the actual positions of the feature points are known, the estimate will be unbiased and the minimum error variance will be the CRLB.

Consider a weighted least squares cost function over two frames. We assume that the FOE is known and we estimate $\mathbf{h}$ and $\mathbf{\Omega}$. Thus

$$\begin{aligned} C &= \frac{1}{2}\sum_{i=1}^{M}\left[\frac{(p_i-\hat{p}_i)^2}{\lambda_{pi}}+\frac{(q_i-\hat{q}_i)^2}{\lambda_{qi}}\right], \\ &= \frac{1}{2}\epsilon^T\Lambda^{-1}\epsilon, \end{aligned} \tag{3.17}$$

where $\epsilon = [p_1-\hat{p}_1, q_1-\hat{q}_1, ..., p_M-\hat{p}_M, q_M-\hat{q}_M]'$ is the estimation error and $\Lambda = \text{diag}[\lambda_{p1}, \lambda_{q1}, ..., \lambda_{pM}, \lambda_{qM}]$ is a $2M \times 2M$ diagonal matrix consisting of the

weights. The prediction errors ϵ are assumed to be independent, zero mean Gaussian with a known covariance matrix $\Lambda_0 = \text{diag}[\lambda_{0p1}, \lambda_{0q1}, ..., \lambda_{0pL}, \lambda_{0qL}]$. We will assume that the FOE is known. If $\hat{\mathbf{x}} = (\hat{\mathbf{h}}, \hat{\mathbf{\Omega}})$ is the estimate obtained from two frames by the least squares minimization procedure and $\mathbf{x}^* = (\mathbf{h}^*, \mathbf{\Omega}^*)$ is the true value, it can be shown using standard linear system identification techniques that the covariance matrix is

$$P = [G(\hat{\mathbf{x}})]^{-1} Q(\hat{\mathbf{x}}) [G(\hat{\mathbf{x}})]^{-1}, \tag{3.18}$$

where $G(\hat{\mathbf{x}}) = \psi(\hat{\mathbf{x}})\Lambda^{-1}\psi^T(\hat{\mathbf{x}})$, $Q(\hat{\mathbf{x}}) = \psi(\hat{\mathbf{x}})\Lambda^{-1}\Lambda_0\Lambda^{-1}\psi^T(\hat{\mathbf{x}})$ and $\psi = -\left[\frac{d\epsilon}{d\mathbf{X}}\right]^T$ [12], [35], [53]. For M features, ψ is an $(M+3) \times 2M$ matrix and thus G, Q and P are $(M+3) \times (M+3)$ matrices.

Using (2.2), we get

$$\psi = \begin{bmatrix} (x_1 - x_f) & (y_1 - y_f) & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & (x_M - x_f) & (y_M - y_f) \\ r_1' & s_1' & r_2' & \cdots & r_N' & s_N' \end{bmatrix} \tag{3.19}$$

as a $(M+3) \times 2M$ matrix. Thus

$$\begin{aligned} G &\triangleq \begin{bmatrix} U & W \\ W' & V \end{bmatrix} \\ Q &\triangleq \begin{bmatrix} U_0 & W_0 \\ W_0' & V_0 \end{bmatrix}, \end{aligned} \tag{3.20}$$

where

$$\begin{aligned} U &= \text{diag}\left[\frac{(x_i - x_f)^2}{\lambda_{pi}} + \frac{(y_i - y_f)^2}{\lambda_{qi}}\right]_{i=1,...M}, \\ V &= \sum_{i=1}^{M} \left(\frac{r_i' r_i}{\lambda_{pi}} + \frac{s_i' s_i}{\lambda_{qi}}\right) \end{aligned}$$

$$
\begin{aligned}
W &= \left[\frac{(x_1 - x_f)r_1^{'}}{\lambda_{p1}} + \frac{(y_1 - y_f)s_1^{'}}{\lambda_{q1}}, ..., \frac{(x_M - x_f)r_M^{'}}{\lambda_{pM}} + \frac{(y_M - y_f)s_1^{'}}{\lambda_{qM}}\right] \\
U_0 &= \text{diag}\left[\frac{(x_i - x_f)^2}{\lambda_{pi}^2}\lambda_{0pi} + \frac{(y_i - y_f)^2}{\lambda_{qi}^2}\lambda_{0qi}\right]_{i=1,...M}, \\
V_0 &= \sum_{i=1}^{M}\left(\frac{r_i^{'}r_i}{\lambda_{pi}^2}\lambda_{0pi} + \frac{s_i^{'}s_i}{\lambda_{qi}^2}\lambda_{0pi}\right) \\
W_0 &= \left[\frac{(x_i - x_f)r_i^{'}}{\lambda_{pi}^2}\lambda_{0pi} + \frac{(y_i - y_f)s_i^{'}}{\lambda_{qi}^2}\lambda_{0qi}\right]_{i=1,...,M}. \qquad (3.21)
\end{aligned}
$$

The variance of the estimator can now be computed by substituting all the terms from (3.20) and (3.21) into the expression for P in (3.18). It is well known in the theory of weighted least squares that the minimum error in the estimator is obtained when $\Lambda = \Lambda_0$ [37], i.e. the actual covariance of the errors in the motion estimates is known. In that case, $G = Q$ and $P = G^{-1}$, which is approximately the inverse of the Hessian of the cost function [54]. Since the errors were assumed to be Gaussian, we can now invoke the Gauss-Markov theorem [37] to prove that the minimum variance (CRLB for the unbiased estimator) in the SfM reconstruction from two frames is P. Since we assumed that the feature point positions $\{x_i, y_i\}$ are known exactly, the estimate is unbiased. If $I(\mathbf{x})$ is the Fisher information matrix of $\mathbf{x}$, then using the fact that for an unbiased efficient estimator the Fisher information is the inverse of the CRLB, we get $I(\mathbf{x}) = P^{-1}$.

Fisher Information for Multi-frame Reconstruction: Let $(\hat{\mathbf{h}}^1, ..., \hat{\mathbf{h}}^L)$ be a sequence of unbiased two-frame estimates, assumed to be statistically independent and each corrupted by zero-mean additive noise. Let the true value be $\bar{\mathbf{h}}$. All the values are obtained with respect to a particular gauge $\mathcal{C}$. This can be done by fixing the scale factor across all the L observations for a particular feature and maintaining that scale for all the other features. Then the maximum likelihood (ML) estimate for the structure over all L observations is

$\hat{\mathbf{h}} = \frac{1}{L}\sum_{i=1}^{L} \hat{\mathbf{h}}^i$. Let the covariance matrix P in (3.18) derived from the i and $(i+1)^{\text{st}}$ frames be denoted by P^i. Let P_h^i be the upper $M \times M$ sub-matrix of P^i representing the covariance of $\hat{\mathbf{h}}^i$. Since the ML estimate is unbiased (as we calculated it assuming that the actual values of the features are known), i.e. $E[\hat{\mathbf{h}}] = \bar{\mathbf{h}}$, we can write

$$\begin{aligned} \text{Cov}[\hat{\mathbf{h}}] &= E\left[(\hat{\mathbf{h}} - \bar{\mathbf{h}})(\hat{\mathbf{h}} - \bar{\mathbf{h}})'\right] \\ &= \frac{1}{L^2}\sum_{i=1}^{L} \text{Cov}\left[\hat{\mathbf{h}}^i\right] \\ &= \frac{1}{L^2}\sum_{i=1}^{L} P_h^i \\ &\triangleq P_L. \end{aligned} \tag{3.22}$$

Since the ML estimate is efficient and unbiased, the covariance is the inverse of the Fisher information matrix. Thus the Fisher information matrix of $\hat{\mathbf{h}}$, denoted by $I_{\mathcal{C}}(\bar{\mathbf{h}})$, is

$$I_{\mathcal{C}}(\bar{\mathbf{h}}) = P_L^{-1}. \tag{3.23}$$

The reason we required the assumption that the true values of the coordinates are known, was to obtain an unbiased estimate. Only then can we invert the covariance matrix to obtain the Fisher matrix. It is worth noting that the CRLB computations presented in the literature implicitly assume that the exact feature positions are known, and thus are at best approximations.

3.4.2 Computing the Generalized CRLB

The expression for the CRLB that is often used in practice assumes the estimate to be unbiased. This is because it is difficult to know the bias of an estimator. The general expression for the CRLB after incorporating the bias in the estimate

and under the proper regularity assumptions is [37]

$$\Sigma_\theta(g) \geq b_\theta(g) b_\theta(g)^T + (I + \nabla_\theta b_\theta(g)) M^{-1}(\theta)(I + \nabla_\theta b_\theta(g))^T, \tag{3.24}$$

where g is the estimate of the parameter θ, b is the bias of the estimate, and M is the Fisher information matrix. ∇_θ is the gradient with respect to θ and I is an identity matrix of suitable size.

The expression for the CRLB as derived above in (3.23) also assumes the estimate to be unbiased. However, as we have shown, this is not a valid assumption. In the next section, we will prove through simulations that the magnitude of the bias is significant compared to the true depth. Hence it is important to account for the effect of the bias in the CRLB as it is a measure of the minimum error (measured by the variance) in the estimate. Since we know the expression for the bias, we can obtain a more accurate expression for the minimum variance that we can expect to obtain. Let $\hat{\mathbf{h}}$ denote the estimate of $\bar{\mathbf{h}}$ (the true value) with respect to a particular gauge $\mathcal{C}$ [55]. Let the bias in the multi-frame estimate be denoted by $b_{\mathcal{C}}(\hat{\mathbf{h}})$ (Section 3.3). Since the bias does not depend on $\bar{\mathbf{h}}$, $\nabla_{\bar{\mathbf{h}}} b_{\mathcal{C}}(\hat{\mathbf{h}}) = 0$. Let the FI matrix for multi-frame reconstruction, as derived in Section 3.4.1, be denoted by $I_{\mathcal{C}}(\bar{\mathbf{h}})$. This derivation is based on the assumptions that the feature positions are exactly known and the motion estimates $\{p_i, q_i\}$ are corrupted by additive white Gaussian noise. Since the estimate in this case is unbiased, the FI matrix can be inverted to obtain the CRLB. Thus the variance in the biased estimate $\hat{\mathbf{h}}$, represented as $\Sigma_{\bar{\mathbf{h}}}(\hat{\mathbf{h}}) = E\left[(\hat{\mathbf{h}} - \bar{\mathbf{h}})(\hat{\mathbf{h}} - \bar{\mathbf{h}})^T\right]$ must justify the following inequality (from (3.24)):

$$\Sigma_{\bar{\mathbf{h}}}(\hat{\mathbf{h}}) \geq b_{\mathcal{C}}(\hat{\mathbf{h}}) b_{\mathcal{C}}^T(\hat{\mathbf{h}}) + I_{\mathcal{C}}^{-1}(\bar{\mathbf{h}}), \tag{3.25}$$

This is the minimum variance of the structure estimate obtained from a 3D

reconstruction algorithm using optical flow.

3.5 Simulation Results

In this section, we describe the results of experiments which we conducted in order to get an idea about the importance of the bias term in the overall reconstruction, as well as to understand the effects of the different motion parameters on the bias.

3.5.1 Effect of Bias on Reconstruction

In this set of experiments, we plotted the actual reconstruction estimate, with and without bias compensation, against the true depth. A set of 10 random 3D points were generated and their 2D projections were computed at different camera positions. The set of feature points were tracked across a few frames. The depths from each pair of frames were obtained (by solving the least squares problem in (2.2)) and then combined to get the ML estimate over the entire sequence. To fix the scale of the reconstruction, the depth at the first point was used. In these experiments we considered the case of non-zero but constant linear and angular camera motion. The effect of measurement noise was studied by adding different levels of noise to the feature positions. Figures 3.1 (a), (b), (c) and (d) are for four different noise variances, σ^2_{x1}, σ^2_{x2}, σ^2_{x3} and σ^2_{x4}, where $\sigma^2_{x4} > \sigma^2_{x3} > \sigma^2_{x2} > \sigma^2_{x1}$. It can be seen that bias compensation makes the estimate closer to the true value in all the cases and gives significant advantages for some of the points.

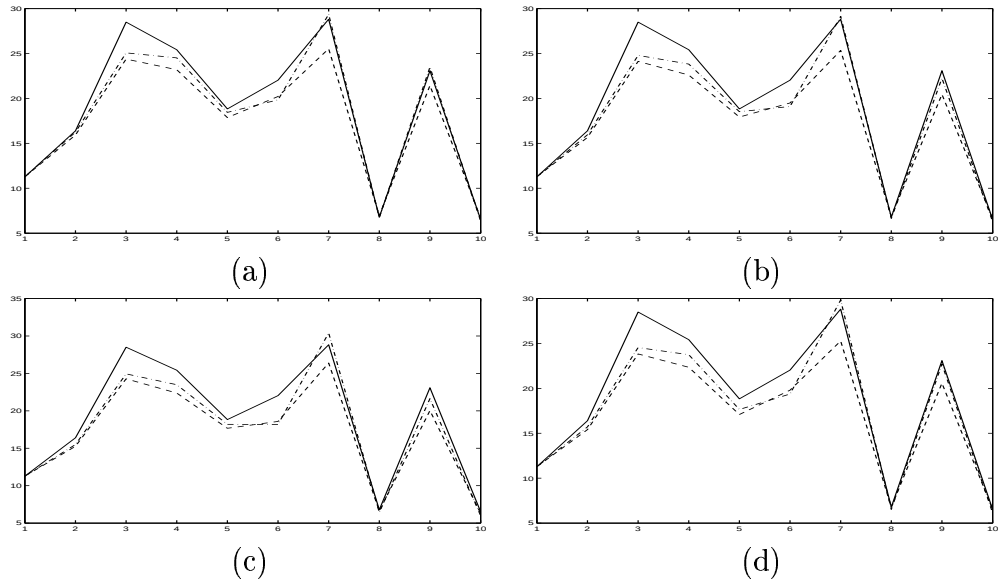


Figure 3.1: (a), (b), (c) and (d) are plots of the reconstruction for noise variances σ_{x1}^2, σ_{x2}^2, σ_{x3}^2 and σ_{x4}^2 in the feature positions, where $\sigma_{x4}^2 > \sigma_{x3}^2 > \sigma_{x2}^2 > \sigma_{x1}^2$. The plots are for the same set of ten 3D points tracked over 15 frames. The camera is moving with constant, non-zero translation and rotation. The solid lines indicate the true depth values, the dashed lines indicate the reconstruction without bias compensation, and the dashed and dotted lines indicate the reconstruction with bias compensation.

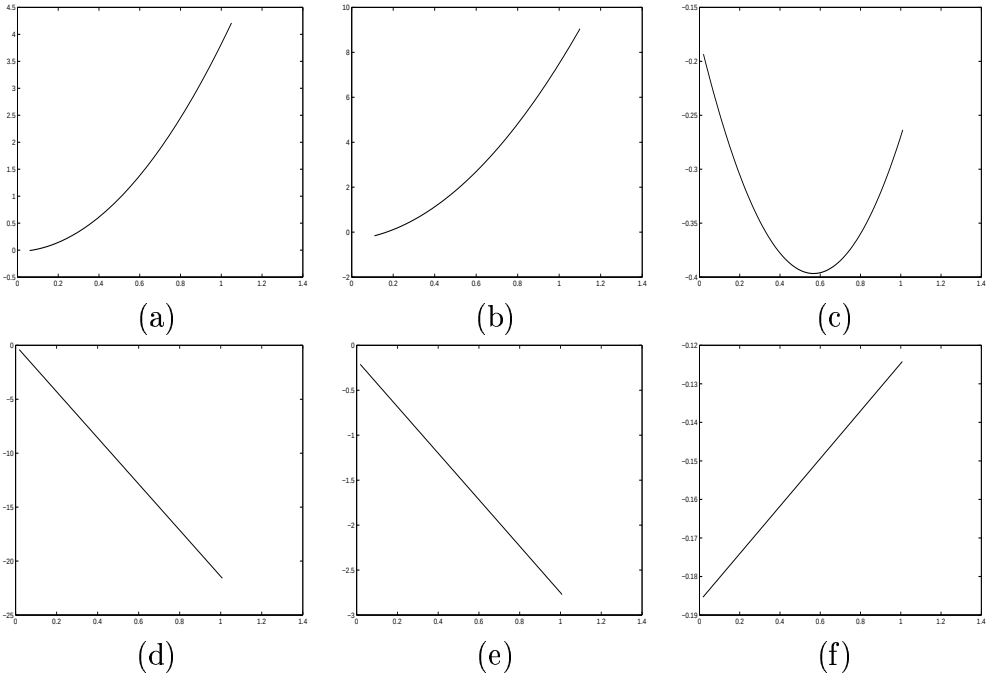


Figure 3.2: Plots of the variation in the bias in inverse depth with the camera motion parameters. The horizontal axes represent the following: (a): $v_x \in (0, 1)$cm/frame, (b): $v_y \in (0, 1)$cm/frame, (c): $v_z \in (0, 1)$cm/frame, (d): $\omega_x \in (0, 10)$degrees/frame, (e): $\omega_y \in (0, 10)$degrees/frame, (f): $\omega_z \in (0, 10)$degrees/frame. The values of the bias on the vertical axis are in percentages of the true inverse depth value. The camera motion is scaled between 0 and 1 on the horizontal axis.

3.5.2 Variation of Bias with Individual Camera Motion Parameters

Given the rather complicated form of (3.3), it is difficult to obtain analytical expressions for the effects of the various camera motion parameters on the reconstruction bias. In this set of experiments, we analyzed the effects of the camera motion through numerical simulations. Each of the six motion parameters were varied over a certain range of values, keeping all the others fixed. While fixing the range over which to vary the camera motion, it should be borne in mind that the basic SfM equations in (2.2) are valid only for small camera

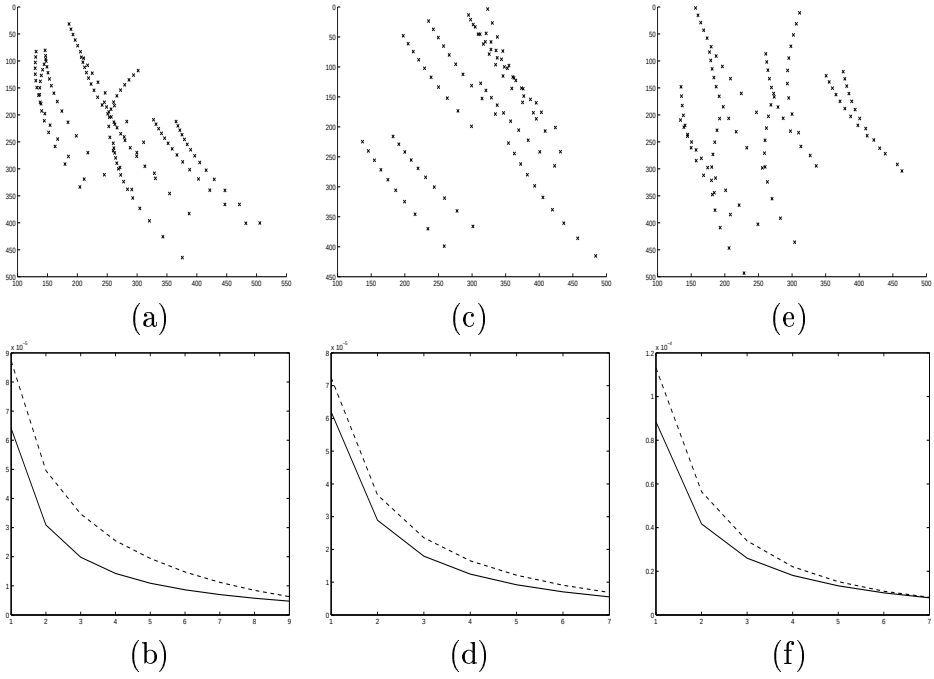


Figure 3.3: Plots of the trajectories of feature points (top row) and the CRLB of the inverse depth as a function of the number of frames, for different camera motion parameters (bottom row). (a), (b): $x_f = 0, y_f = 10, \omega_x = \omega_y = 1\text{degree/frame}, \omega_z = 0$; (c), (d): $x_f = 10, y_f = 0, \omega_x = \omega_y = \omega_z = 1\text{degree/frame}$; (e), (f): $x_f = 10, y_f = 10, \omega_x = \omega_y = \omega_z = 1\text{degree/frame}$. The solid line shows the CRLB for the unbiased estimate and the dotted line for the biased one.

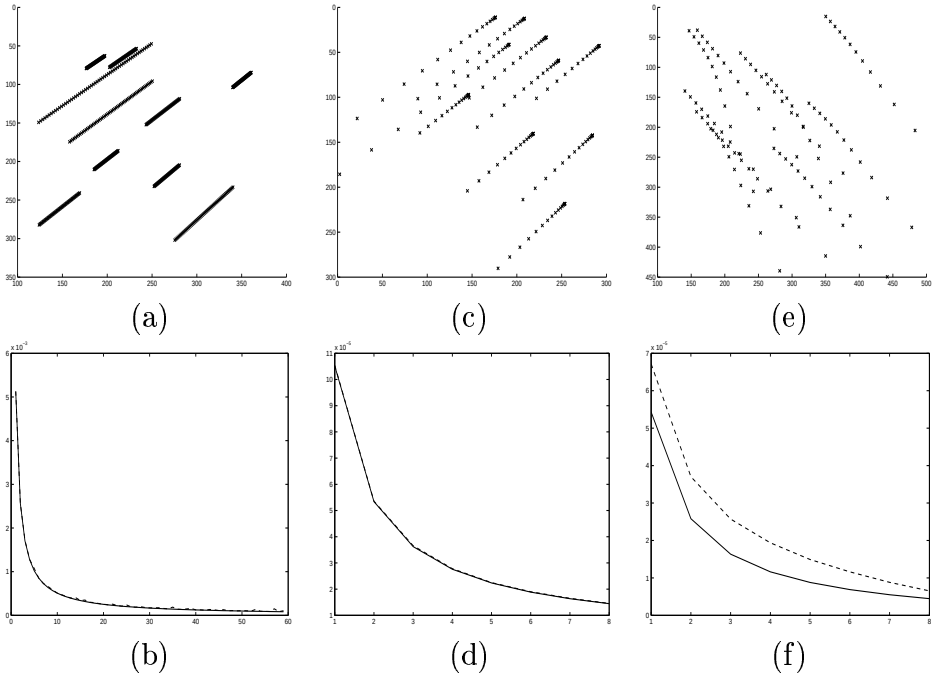


Figure 3.4: Plots of the trajectories of feature points (top row) and the CRLB of the inverse depth as a function of the number of frames, for different camera motion parameters (bottom row). (a), (b): $x_f = 1, y_f = 1, \omega_x = \omega_y = \omega_z = 0$; (c), (d): Uniform acceleration, $\omega_x = \omega_y = \omega_z = 0$; (e), (f): Uniform acceleration, $\omega_x = \omega_y = \omega_z = 1$degree/frame. The solid line shows the CRLB for the unbiased estimate and the dotted line for the biased one.

motion. Thus it does not make sense to study the behavior of the bias term over a large range of camera motion parameters.

The plot of the bias for various values of the camera motion parameters is shown in Figure 3.2. The bias is plotted on the vertical axis as a percentage of the true depth values. The motion terms which affect the bias most are $(v_x, v_y, \omega_x, \omega_y)$. The effect of (v_z, ω_z) on the bias is almost negligible. Also, the variation of the bias with v_x and v_y follows an approximate square law, while the variation with ω_x, ω_y is almost linear.

3.5.3 The Generalized CRLB

Next we present the effect of the bias term on the CRLB. A set of 3D points was generated in space. Different trajectories were generated by varying the camera motion parameters. The perspective projections of the 3D points onto the image plane were tracked across a sequence of 15 frames. Random noise was added to the positions of the points in the images. We present here the effects of the different motion parameters on the CRLB.

Figures 3.3 and 3.4 plot the trajectories of the feature points and the CRLB of the structure estimates. The trajectories of the points tracked across all the frames are shown in the top row, while the CRLB for the unbiased estimator and the generalized CRLB for the biased one are plotted in the bottom row as a function of the number of frames. The plots are for various values of the camera motion parameters $x_f, y_f, \omega_x, \omega_y, \omega_z$. We will now briefly analyze the different cases.

1. *Constant linear velocity, non-zero angular velocity:* This is the case in all the plots in Figure 3.3. It can be seen that the effects of different values of the linear velocity term on the bias in the structure estimate are not drastically different. Whether one of the components (v_x or v_y) is zero or both of them are non-zero, the bias is significant. There is a distinct upward shift in the minimum reconstruction error in all these cases.

2. *No Rotation:* This is the case of particular interest. When the rotational velocity is zero, the bias term is negligible; the difference in the CRLB is too small to represent in the plots of Figures 3.4(b) and 3.4(d)). This is the case irrespective of whether (i) V is constant but non-zero; (ii) V is constant but either v_x or v_y is zero; (iii) there is an acceleration in the

linear component of the velocity. The reason for this can be understood from the expression for the bias in (3.3). In the first term, the absolute value of the numerator v_i is extremely small compared to the denominator m_{ii}^2. The other terms which multiply Ω are zero. Thus the bias is also small unless the variances of the FOE and of the image correspondences are large enough to compensate for the small numerator.

3. *Linear Acceleration, Constant Rotation:* This is the case in Figure 3.4(f). Again, we see that the bias is significant once the rotational velocity is non-zero.

The conclusion that can be drawn from this analysis is that the parameters which affect the bias most are the camera angular motion values. *For small rotation, the bias in the estimate is negligible.* While this can be understood mathematically from the expression for the bias as derived above, we do not yet have a physical interpretation for it.

3.6 Conclusion

The analysis of the accuracy of 3D reconstruction has usually focused on the error covariance of the estimate. In this chapter, we have pointed out that there is another source of error in the SfM problem, namely the bias in the estimate. This has been observed in psychophysical experiments, and our mathematical analysis supports that fact. Our derivation of the bias term was based on the fact that the solution of a least-squares estimation problem with noisy system matrix is statistically biased. The system matrix in SfM contains the positions of the features, which can never be obtained exactly. A generalized Cramer-Rao

lower bound for SfM is proposed after incorporating the bias term. Simulations were carried out in order to show the effects of the different camera motion parameters on the bias. It was observed that the bias is negligibly small if the camera angular motion is small. A comparison between a bias-compensated SfM reconstruction with one in which the bias is neglected was presented.

Chapter 4

3D Face Reconstruction Algorithm

4.1 Introduction

Reconstructing 3D models from video sequences is an important problem in computer vision with applications to recognition, medical diagnosis, video communications, etc. Though numerous algorithms exist which can reconstruct a 3D scene from two or more images using structure from motion (SfM) [1, 3], the quality of such reconstructions is often unsatisfactory. In the previous two chapters, we analyzed the errors which affect the reconstruction quality. In this chapter, we show how to use the theoretical analysis to build accurate 3D models from a video sequence. One particularly interesting application of 3D reconstruction from 2D images is in the area of modeling a human face from video. The successful solution of this problem has immense potential for applications in face recognition, surveillance, multimedia, etc. A few algorithms exist which attempt to solve this problem using a generic model of a face [56, 57]. Typically, these methods initialize the reconstruction algorithm with this generic model. The difficulty with this approach is that the algorithm often converges to a solution very near the initial value, resulting in a reconstruction which has the characteristics

of the generic model, rather than that of the particular face in the video which needs to be modeled. This method may give very good results when the generic model has significant similarities with the particular face being reconstructed. However, if the features of the generic model are different from those of the face being reconstructed, the solution obtained using this approach may be highly erroneous.

We propose an alternative way of reconstructing a 3D model of a face. Our method also incorporates a generic model; however, we do so *after* obtaining the estimate using the SfM algorithm. The SfM algorithm reconstructs purely from the video data after computing the optical flow. We adopt a multi-frame reconstruction strategy whereby two-frame reconstructions are fused together after evaluating the quality of each such intermediate reconstruction. The intermediate estimates are combined together using a robust least median of squares (LMedS) estimator [40, 38]. The fusion is done recursively using the Robbins-Monro stochastic approximation (RMSA) [12, 58] algorithm. Quality evaluation is done by estimating the statistical error covariance analytically from the optical flow equations as explained in Section 2.4, and the progress of the fusion algorithm is continuously evaluated using the distortion function (Section 2.5) in order to determine the number of such intermediate reconstructions which are required. This reconstruction is then combined with the generic model using an energy function minimization framework [59, 60]. A cost function is proposed that compares local regions where there are no sharp depth discontinuities and corrects for errors in those regions. Optimization is done in a Markov Chain Monte Carlo (MCMC) [61, 62] framework using a Metropolis-Hastings sampler [63, 64]. The advantage of this method is that the particular characteristics of

the face that is being modeled are not lost since the SfM algorithm does not incorporate the generic model. However, any errors in the reconstruction are corrected in the energy function minimization process by comparison with the generic model.

4.1.1 Incorporating a Generic Model in an Energy Function Minimization Framework

The 3D reconstruction framework based on the uncertainty calculations provides a depth estimate using the input video data only. However, localized errors still remain which were not detected using the error correction strategy described above. The reason is that while the error covariance calculations can identify and correct for small errors, they are unable to correct the larger errors due to outliers. We use a generic face model in order to overcome such errors. A regularization approach to incorporating the generic model is proposed by imposing smoothness constraints on the final 3D reconstruction. A pertinent question to ask here is: why do we need the error correction strategy (in the SfM algorithm) and the generic model? Is it not sufficient to have a simple multi-frame reconstruction algorithm without the error correction strategies, followed by the generic model to correct for all the errors? The answer is negative, because if we use the generic model to correct for all the errors, we run into the problem of over-smoothing the 3D structure estimate. This is similar to the situation when the generic model is incorporated at the beginning of the SfM algorithm, as explained before. The aim here is to obtain as precise a 3D model as possible from SfM and then use the generic model to correct for the errors that remain.

The idea of using energy functions (also known as variational methods, regu-

larization theory, and relaxation methods) [59] to impose smoothness constraints has been very influential in vision [65, 66, 67]. Regularization theory works by minimizing a functional $E[f(x)]$ with respect to a function $f(x)$. It usually contains one term (a consistency or fidelity term) which ensures that the smoothed solution is close to the data, and a second term (a regularization term) which imposes smoothness constraints on the solution. In most implementations, where the data is specified on a regular lattice, the energy function is discretized as $E[f_i]$. The energy minimization/regularization approach can be incorporated directly into a Bayesian statistical framework using the Gibbs distribution by defining a probability $P(f) = \frac{1}{Z}\exp(-\beta E(f))$, where β is a constant and Z is a normalization term [68]. The use of Gibbs distributions on discretized energy functions leads to a mathematical structure known as Markov Random Fields (MRF) [69]. An MRF consists of a probability distribution over a set of variables $\{f_i\}$, with a neighborhood $\mathcal{N}_i$, such that $P(f_i|f_j, j \in \mathcal{N}_i) = P(f_i|f_j, \text{for all } j)$. One of the most influential formulations of the vision problem in terms of MRFs has been the work of Geman and Geman [68] on image segmentation, in which images are smoothed except at places where the image values are rapidly changing.

In our problem, the 3D estimate obtained from the multi-frame reconstruction algorithm needs to be smoothed in local regions where there are errors. These regions are identified with the help of the generic model. After the 3D depth estimate and the generic model have been aligned, the boundaries where there are sharp depth discontinuities are identified from the generic model. Each vertex of the triangular mesh representing the model is assigned a binary random variable (defined as a line process, following the terminology of [68]) depending

upon whether or not it is part of a depth boundary. Regions which are inside these boundaries are smoothed. The energy function consists of two terms which determine the closeness of the final smoothed solution to either the generic model or the 3D depth estimate, and a third term which determines whether or not a particular vertex of the mesh should be smoothed based on the value of the random variable representing the line process for that vertex. The combinatorial optimization problem is solved using simulated annealing and a Markov Chain Monte Carlo sampling strategy [63, 60, 62].

In the next section, we will explain our method for estimating 3D structure purely from the video data using stochastic approximation (SA) techniques. The incorporation of the generic model along with the 3D estimate will be explained in Section 4.3. Experimental results of the 3D models obtained using the method will be presented in Section 4.4. An overview of stochastic approximation is presented in Appendix A.

4.2 Estimating 3D Structure and Motion From Video

In this section, we explain the first part of our face reconstruction algorithm, i.e. estimating the 3D structure using SfM. Recall equations (2.22) and (2.23) from Section 2.4. There we showed that

$$\begin{aligned} \mathbf{H} &= \sum_{\bar{i}=1}^{N} \left(A_{\bar{i}p}{}' A_{\bar{i}p} + A_{\bar{i}q}{}' A_{\bar{i}q} \right) \\ \mathbf{R_z} &= \mathbf{H}^{-1} \left(\sum_{\bar{i}=1}^{N} \left(A_{\bar{i}p}{}' A_{\bar{i}p} R_{u\bar{i}p} + A_{\bar{i}q}{}' A_{\bar{i}q} R_{u\bar{i}q} \right) \right) \mathbf{H}^{-T}, \end{aligned} \tag{4.1}$$

which can be partitioned as

$$\mathbf{R_z} = \begin{bmatrix} \mathbf{R_h} & \mathbf{R_{hm}} \\ \mathbf{R_{hm}^T} & \mathbf{R_m} \end{bmatrix}. \tag{4.2}$$

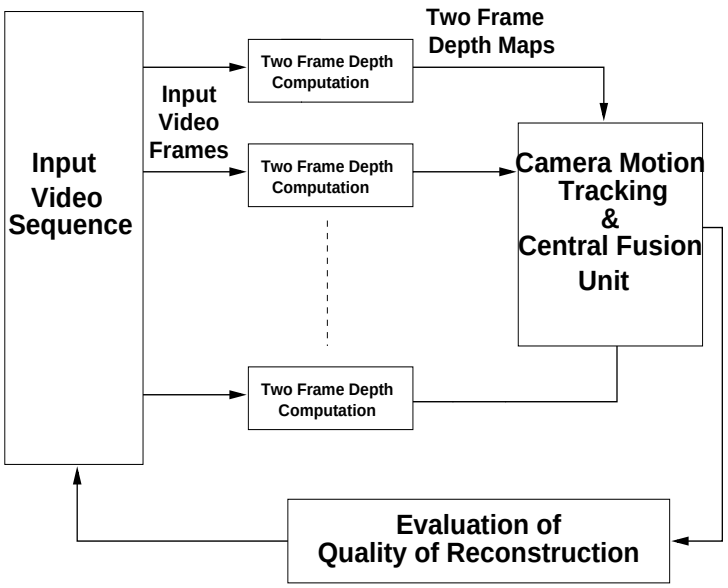


Figure 4.1: Block diagram of the 3D Reconstruction Framework.

Also recall the multi-frame distortion function (2.44). We will use these expressions to evaluate the quality of our 3D reconstruction algorithm.

4.2.1 Estimating 3D Depth

Figure 4.1 shows a block-diagram schematic of the complete 3D face reconstruction framework using SfM. The input is a video sequence. We choose an appropriate two-frame depth reconstruction strategy [8]. The depth maps are aligned to a single frame of reference and the aligned depth maps are fused together using stochastic approximation. We use the distortion function in (2.44) to evaluate the quality of the final reconstruction. This is used to optimize the fusion strategy and design a stopping criterion.

Let s^i represent the structure[1], computed for a particular point, from the

[1]In our description, subscripts will refer to feature points and superscripts will refer to frame numbers. Thus x_i^j refers to the variable x for the i-th feature point in the j-th frame.

i-th and $(i+1)$-st frame, $i = 1, ..., K$, where the total number of frames is $K+1$. Let the fused structure sub-estimate at the i-th frame be denoted by S^i.[2] Let $\mathbf{\Omega}^i$ and $\mathbf{V}^i$ represent the rotation and translation of the camera between the i-th and $(i+1)$-st frames. Note that the camera motion estimates are valid for all the points in the object in that frame. The 3×3 rotation matrix $\mathbf{P}^i$ describes the change of coordinates between times i and $i+1$, and is orthonormal with positive determinant. When the rotational velocity $\mathbf{\Omega}$ is held constant between time samples, $\mathbf{P}$ is related to $\mathbf{\Omega}$ by $\mathbf{P} = e^{\hat{\mathbf{\Omega}}}$.[3] The fused sub-estimate S^i can now be transformed as $T^i(S^i) = \mathbf{P}^i S^i + \mathbf{V}^{iT}$. But in order to do this, we need to estimate the motion parameters $\mathbf{V}$ and $\mathbf{\Omega}$. Since we can determine only the direction of translational motion $(v_x/v_z, v_y/v_z)$, we will represent the motion components by the vector $\mathbf{m} = [\frac{v_x}{v_z}, \frac{v_y}{v_z}, \omega_x, \omega_y, \omega_z]$. To keep the notation simple, m will be used to denote each of the components of $\mathbf{m}$. Thus, the problem at stage $(i+1)$ will be to i) reliably track the motion parameters obtained from the two-frame solutions, and ii) fuse s^{i+1} and $T^i(S^i)$. If $\{l^i\}$ is the transformed sequence of inverse depth values with respect to a common frame of reference, then the optimal value of the depth at the point under consideration is obtained as

$$u^* = \arg\min_u \text{median}_i \left(w_l^i (l^i - u)^2 \right), \tag{4.4}$$

where $w_l^i = (\mathbf{R}_l^i)^{-1}$, where $\mathbf{R}_l^i$ is the covariance of the inverse depth l^i as obtained

---

[2] s^i and S^i are position vectors in $\mathbf{R}^3$.

[3] For any vector $\mathbf{a} = [a_1, a_2, a_3]$, there exists a unique skew-symmetric matrix

$$\hat{a} = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix}. \tag{4.3}$$

The operator $\hat{a}$ performs the vector product on $\mathbf{R}^3$: $\hat{a}X = \mathbf{a} \times X, \forall X \in \mathbf{R}^3$.
With an abuse of notation, the same variable is used for the random variable and its realization.

in (2.19). However, since we are using a recursive strategy, it is not necessary to align all the depth maps to a common frame of reference *a priori*. We will use a Robbins-Monro stochastic approximation (RMSA) algorithm where it is enough to align the fused sub-estimate and the two-frame depth for each pair of frames and proceed as more images become available.

For each feature point, we compute $X^i(u) = w_l^i(l^i - u)^2, u \in \mathcal{U}$. Our aim is to compute the median (say θ) of $X^0, ..., X^K$, i.e. to obtain θ such that $g(\theta) = F_X(\theta) - 0.5 = 0$, where $F_X(\theta)$ is the distribution function of θ. Define $Y^k(\hat{\theta}^k) = p^k(\hat{\theta}^k) - 0.5$, where $p^k(\hat{\theta}^k) = \mathbf{I}_{[X^k \leq \hat{T}^k(\hat{\theta}^k)]}$ ($\mathbf{I}$ represents the indicator function, $\hat{T}^k$ is the estimate of the camera motion, and $\hat{\theta}^k$ is the estimate obtained at the k^{th} stage). Then

$$\begin{aligned} E[Y^k(\hat{\theta}^k)|\hat{\theta}^k] &= E[p^k(\hat{\theta}^k)|\hat{\theta}^k] - 0.5 \\ &= E[\mathbf{I}_{[X^k \leq \hat{T}^k(\hat{\theta}^k)]}] - 0.5 \\ &= P(X^k \leq \hat{T}^k(\hat{\theta}^k)) - 0.5 \\ &= F_X(\hat{\theta}^k) - 0.5 = g(\hat{\theta}^k). \end{aligned}$$

Then the RM recursion for the problem is [70]

$$\hat{\theta}^{k+1} = \hat{T}^k(\hat{\theta}^k) - a^k(p^k(\hat{\theta}^k) - 0.5), \tag{4.5}$$

where a^k is determined by (A.3). When $k = K$, we obtain the fused inverse depth $\hat{\theta}^{K+1}$, from which we can get the fused depth value S^{K+1}.

4.2.2 Camera Motion Tracking:

Since depth and motion computation are dependent on each other, there is every reason to be suspicious of the camera motion values also. However, experimental

analysis has shown that the camera motion is less prone to outliers than the depth estimates. A possible reason for this is that the camera motion is obtained using a larger number of feature points in the image and thus is less susceptible to input errors in some of the features. Our camera motion estimator is a smoothing filter which tracks the motion across the frames and removes any sharp unwanted variations. The discrete-time dynamical model of the camera motion is

$$\begin{aligned} \mathbf{m}^i &= \mathbf{m}^{i-1} + \mathbf{w}^i, \\ \mathbf{y}^i &= \mathbf{m}^i + \mathbf{v}^i. \end{aligned} \tag{4.6}$$

$\mathbf{w}$ is modeled as a zero-mean white noise process with $E[\mathbf{w}^i \mathbf{w}^j] = \mathbf{Q}^i \delta(i,j)$. The observations $\mathbf{y}^i$ of the camera motion (output of the two-frame algorithm) are corrupted by a zero-mean noise process $\mathbf{v}^i$ with a diagonal covariance matrix $\mathbf{V}^i$. $\mathbf{v}$ and $\mathbf{w}$ are assumed to be mutually uncorrelated across all instants of time, i.e. $E[\mathbf{v}^i \mathbf{w}^j] = 0$ for all (i,j), and are also independent of the parameter $\mathbf{m}^i$ at all time instants. We are interested in designing a linear mean square error (LMSE) estimator of the camera motion $\mathbf{m}^t$ based on the observations $\mathbf{y} = [\mathbf{y}^t, \mathbf{y}^{t-1}, ..., \mathbf{y}^{t-k+1}]'$. Let $\hat{\mathbf{m}}^{t|s}$ denote the estimate of $\hat{\mathbf{m}}^t$ based on the observations $[\mathbf{y}^1, ..., \mathbf{y}^s]$ and $\Sigma^{t|s} = E[(\mathbf{m}^t - \hat{\mathbf{m}}^{t|s})(\mathbf{m}^t - \hat{\mathbf{m}}^{t|s})']$. Then the LMSE estimate can be obtained from the Kalman filtering algorithm as follows. Re-indexing the observation vector $\mathbf{y}$ as $[\mathbf{y}^k, ..., \mathbf{y}^1]$, the Kalman filter is given by the following recursion [35]:

$$\begin{aligned} \hat{\mathbf{m}}^{k|k} &= \hat{\mathbf{m}}^{k|k-1} + K^k(y^k - \hat{\mathbf{m}}^{k|k}) \\ \hat{\mathbf{m}}^{k|k-1} &= \hat{\mathbf{m}}^{k-1|k-1} \\ K^k &= \Sigma^{k|k-1}[\mathbf{V}^k + \Sigma^{k|k-1}]^{-1} \\ \Sigma^{k|k-1} &= \Sigma^{k-1|k-1} + Q^k. \end{aligned} \tag{4.7}$$

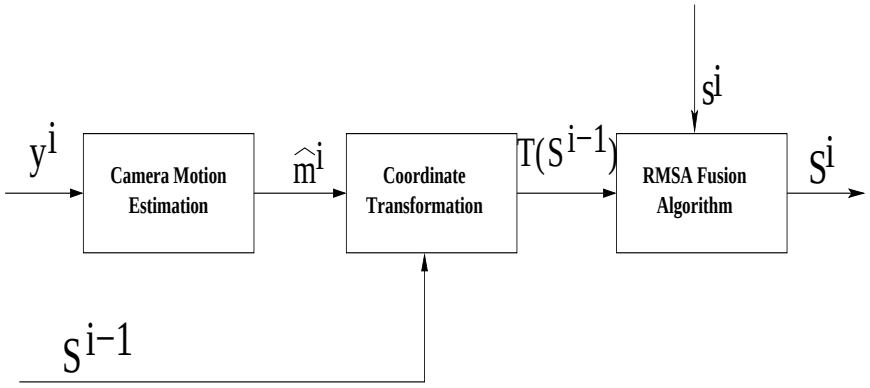


Figure 4.2: Block diagram of the multi-frame fusion algorithm.

Then $\Sigma_{\mathbf{y}^k} = E[(\mathbf{y}^k - E[\mathbf{y}^k])(\mathbf{y}^k - E[\mathbf{y}^k])'] = E[(\mathbf{m}^k + \mathbf{v}^k - \mu_{\mathbf{m}})(\mathbf{m}^k + \mathbf{v}^k - \mu_{\mathbf{m}})'] = E[(\mathbf{m}^k - \mu_{\mathbf{m}})(\mathbf{m}^k - \mu_{\mathbf{m}})'] + \mathbf{V}^k = \mathbf{R}^k_{\mathbf{m}}$, where $\mu_{\mathbf{m}} = E[\mathbf{m}^i] = E[\mathbf{m}^{i-1}] = E[\mathbf{y}^i]$. Thus the observation noise covariance can be estimated from (2.36) and the camera motion filter is derived.

Why Kalman Filter? Since the system dynamics of the camera motion are time-varying, SA techniques are not guaranteed to converge. (One often chooses the step size a_k in (A.2) to be a small positive number as a tradeoff between tracking capability and noise sensitivity [12].) Also, the presence of outliers in two-frame camera motion estimates is less pronounced than in the depth sub-estimates; hence least squares is a good criterion for tracking camera motion. The difficulty of incorporating time-varying dynamics into the SA approach, coupled with the suitability of a least squares criterion, dictates the choice of the Kalman filter for camera motion estimation.

4.2.3 The Reconstruction Algorithm

Assume that we have the fused 3D structure S^i obtained from i frames and the two-frame depth map s^{i+1} computed from the i-th and $(i+1)$-st frames. Figure

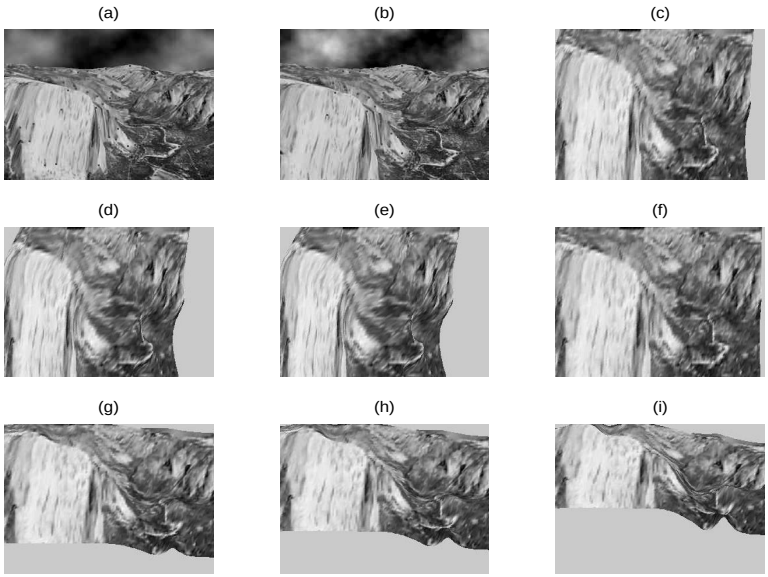


Figure 4.3: (a) and (b) represent two images from the Yosemite video sequence for which the depth was computed. The remaining figures (c) - (i) are results of 3D reconstruction from 15 frames for different viewing angles.

4.2 shows a block diagram of the multi-frame fusion algorithm. The main steps of the algorithm are:

Track Estimate the camera motion according to the camera motion tracking algorithm.

Transform Transform the previous model S^i to the new reference frame.

Update Update the transformed model using s^{i+1} to obtain S^{i+1} from (4.5).

Evaluate Reconstruction Compute a performance measure for the fused reconstruction from (2.44).

Iterate Decide whether to stop on the basis of the performance measure. If not, set $i = i + 1$ and go back to Track.

Although our primary application is to obtain 3D models of human faces, we will present the result of applying the reconstruction algorithm described above to a completely different video sequence known as the Yosemite sequence. Figure 4.3 shows the reconstruction of the 3D scene from this sequence. Fifteen frames were used for this reconstruction. Figures 4.3(a) and 4.3(b) represent two frames from the original sequence. The depth was reconstructed using a two-frame algorithm [8] and our fusion strategy and the 3D model was constructed. Figures 4.3(c) to 4.3(i) represent views of the 3D model from different angles.

4.3 Incorporating the Generic Model in 3D Face Reconstruction

For the 3D face reconstruction problem, we can take advantage of the fact that the general structure of most faces is similar. In this section, we will explain our method for combining the generic model with the 3D estimate obtained from the video sequence using the SfM algorithm described in the previous section. We propose an optimization framework for combining the two models in such a way that the errors in the 3D estimate are corrected by comparison with the generic model.

4.3.1 The Optimization Function:

Both the generic model and the 3D estimate have a triangular mesh representation with N vertices and the depth at each of these vertices is known. By depth, we mean the z coordinate of the vertex represented by (x, y, z). The (x, y) plane is considered parallel to the image plane. Let $\{d_{g_i}, i = 1, ..., N\}$ be the set of

depth values of the generic mesh for each of the N vertices of the triangles of the mesh. Let $\{d_{s_i}, i = 1, ..., N\}$ be the corresponding depth values from the SfM estimate. We wish to obtain a set of values $\{f_i, i = 1, ..., N\}$ which are a smoothed version of the SfM model, after correcting the errors on the basis of the generic mesh.

Since we want to retain the specific features of the face we are trying to model, our error correction strategy works by comparing local regions in the two models and smoothing those parts of the SfM estimate where the *trend* of the depth values is significantly different from that in the generic model, e.g. a sudden peak on the forehead will be detected as an outlier after the comparison and smoothed. This is where our work is significantly different from previous work [56, 57], since we do not intend to fuse the depth in the two models but to correct errors based on local geometric trends. Towards this goal, we introduce a line process on the depth values. The line process indicates the borders where the depth values have sudden changes, and is calculated on the basis of the generic mesh, since it is free from errors. For each of the N vertices, we assign a binary number indicating whether or not it is part of the line process. This concept of the line process is borrowed from the seminal work of Geman and Geman [68] on stochastic relaxation algorithms in image restoration.

The optimization function we propose is

$$\begin{aligned} E(f) &= \sum_{i=1}^{N} (f_i - d_{g_i})^2 + (1-\mu) \sum_{i=1}^{N} (f_i - d_{s_i})^2 + \\ &\quad \mu \sum_{i=1}^{N} (1 - l_i) \sum_{j \in \mathcal{N}_i} (f_i - f_j)^2 \mathbf{I}_{d_s \neq d_g}, \end{aligned} \tag{4.8}$$

where $l_i = 1$ if the i^{th} vertex is part of a line process and μ is a combining factor which controls the extent of the smoothing. $\mathcal{N}_i$ is the set of vertices which are

neighbors of the i^{th} vertex. $\mathbf{I}_{d_s \neq d_g}$ represents the indicator function which is 1 if $d_s \neq d_g$, else 0. In order to understand the importance of (4.8), consider the third term. When $l_i = 1$, the i^{th} vertex is part of a line process and should not be smoothed on the basis of the values in $\mathcal{N}_i$; hence this term is switched off. Any errors in the value of this particular vertex will be corrected on the basis of the first two terms, which control how close the final smoothed mesh will be to the generic one and the SfM estimate. When $l_i = 0$, indicating that the i^{th} vertex is not part of a line process, its final value in the smoothed mesh is determined by the neighbors as well as its corresponding values in the generic model and SfM estimate. The importance of each of these terms is controlled by the factor $0 < \mu < 1$. In the case (largely academic) where $d_s = d_g$, the smoothed mesh can be either d_s or d_g and this is taken care of in the indicator function in the third term in (4.8).

In order to solve the optimization problem in (4.8), we use the technique of simulated annealing built upon a Markov Chain Monte Carlo (MCMC) framework [60, 62, 71]. The MCMC optimizer is essentially a Monte Carlo integration procedure in which the random samples are produced by evolving a Markov chain. Let $T_1 > T_2 > ... > T_k > ...$ be a sequence of monotone decreasing temperatures in which T_1 is reasonably large and $\lim_{T_k \to \infty} = 0$. At each such T_k, we run N_k iterations of a Metropolis-Hastings (M-H) sampler [63, 64] with the target distribution $\pi_k(f) \propto \exp\{-E(f)/T_k\}$.[4] As k increases, π_k puts more

[4]For any given target probability distribution $\pi(x)$, the Metropolis-Hastings algorithm prescribes a transition rule for a Markov chain so that the equilibrium distribution of the chain is $\pi(x)$. To start the algorithm, one needs to choose an arbitrary, but easy to sample from, transition function $T(x, y)$ (also called a *proposal distribution*). Then, given a current state $x^{(t)}$,

- Draw y from the transition function $T(x^{(t)}, y)$.

and more of its probability mass (converging to 1) in the vicinity of the global maximum of E. Since minimizing $E(f)$ is equivalent to maximizing $\pi(f)$, we will almost surely be in the vicinity of the global optimum if the number of iterations N_k of the M-H sampler is sufficiently large. The steps of the algorithm are:

- Initialize at an arbitrary configuration f_0 and initial temperature level T_1.
- For each k, run N_k steps of MCMC iterations with $\pi_k(f)$ as the target distribution. Pass the final configuration of x to the next iteration.
- Increase k to $k+1$.

4.3.2 Mesh Registration:

The optimization procedure described above requires a one-to-one mapping of the vertices $\{d_{s_i}\}$ and $\{d_{g_i}\}$. Once we obtain the estimate from the SfM algorithm, a set of corresponding points between this estimate and the generic mesh is identified manually (as in [56, 57]). This is then used to obtain a registration between the two models. Thereafter, using proper interpolation techniques, the depth values of the SfM estimate are generated corresponding to the (x, y) coordinates of the vertices of the triangles in the generic model. By this method, we obtain meshes with the same set of N vertices, i.e. the same triangulation.

- Draw $U \sim$ Uniform$[0, 1]$ and update

$$x^{(t+1)} = \begin{cases} y & \text{if } U \leq \rho(x^{(t)}, y) \\ x^{(t)} & \text{else.} \end{cases} \tag{4.9}$$

Various functions have been suggested for ρ [62]. Metropolis [72] and Hastings [73] suggested

$$\rho(x, y) = \min\left\{1, \frac{\pi(y)T(x, y)}{\pi(x)T(x, y)}\right\}. \tag{4.10}$$

4.3.3 The Generic Mesh Algorithm:

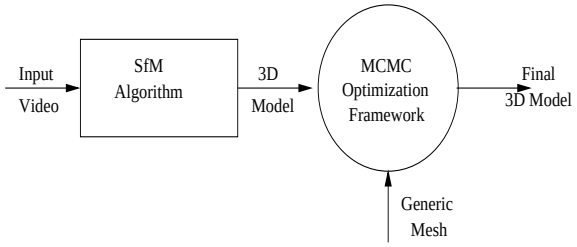


Figure 4.4: A block diagram representation of the complete 3D modeling algorithm using the generic mesh and SfM algorithm.

The main steps of the algorithm for incorporating the generic mesh are as follows.

1. Obtain the 3D estimate from the given video sequence using SfM (output of the reconstruction algorithm of Section 4.2).

2. Register this 3D model with the generic mesh and obtain the depth estimate whose vertices are in one-to-one correspondence with the vertices of the generic mesh.

3. Compute the line processes and to each vertex i assign a binary value l_i.

4. Obtain the smoothed mesh f_i from the optimization function in (4.8).

5. Map the texture onto f from the video sequence.

The Final 3D Model: The complete 3D reconstruction paradigm is composed of a sequential application of the two algorithms (3D Reconstruction Algorithm and Generic Mesh Algorithm) that we described in Sections 4.2 and 4.3. Figure 4.4 represents a block diagram of the complete 3D modeling algorithm.

4.4 3D Face Model Results

4.4.1 Overview of Implementation Strategy

In this section we present results obtained from our algorithm. Video sequences were captured from a hand-held or tripod-mounted video camera. The output is a 3D model of the scene. A MATLAB implementation of the multi-frame fusion algorithm is available. We also have an end-to-end system for 3D reconstruction of a face on a Pentium PC for demonstrations.

Points were tracked across the entire video sequence using a KLT tracker [74]. The set of tracked feature points for every pair of images was given as the input to the two-frame SfM algorithm described in [8]. The output is the depth at these points and the motion of the camera between these frames. For each pair of frames, the covariance of the error in the structure and motion estimates was computed according to (2.23). The depth maps from two consecutive pairs of frames were aligned on the basis of the camera motion estimates as explained in the reconstruction algorithm of Section 4.2. The aligned depth maps were then fused using the recursive RMSA fusion algorithm. The multi-frame distortion curve for the entire video sequence was computed at each step for the individual feature points using (2.42) and for their average representation using (2.44). When the distortion was below an acceptable level, the computation was terminated. If a particular feature was lost after a few frames (occlusion, etc.) as indicated by the KLT, the distortion for that feature was used to decide whether to include it in model building or not. In all the experiments, the FOE was estimated from the first two or three frames and assumed constant thereafter. Comparison with the estimates obtained from every adjacent pair of

frames showed that this was a justified assumption. The depth map obtained at this stage was used to build a 3D model using the Graphics toolbox of MATLAB. The feature points were used to create a Delaunay triangulation. The depth values were assigned to each of the vertices of the triangle in order to create a mesh to which the texture was mapped to create the final 3D model. The method of building the 3D model of the scene is simplistic; it is used only as a means to represent the results of the algorithm. Advanced techniques in computer graphics can produce much better models of the scene; that, however, is not the goal of this work. After the 3D model is obtained, it is combined with the generic model. A set of corresponding points is identified and the two models are registered. Then, the depth values of the 3D estimate at the (x, y) coordinates of the generic model are computed, so that the vertices of the two models are in one-to-one correspondence. The smoothed model can then be obtained by the optimization procedure described in the previous section.

4.4.2 SfM Algorithm

Figure 4.5 shows two images from the video sequence which is the input to the SfM algorithm. We use a two-frame algorithm that computes the structure from the optical flow [8] using two consecutive frames and then integrate over the video sequence using the robust estimation techniques of Section 4.2. The errors in the motion estimates were computed by tracking a set of features over the first few frames of the video sequence, which were not used in the reconstruction. The technique is similar to the gradient-based method of [31], except that, for more accurate results, it was repeated for each of these initial frames and a final estimate was obtained using bootstrapping techniques [75]. Assuming that

the statistics remain stationary over the frames used in the reconstruction, the errors estimated from the first few frames were used for obtaining $\mathbf{R_u}$ in (2.23). The variance in the inverse depth computed using our theoretical analysis of Section 4.2 is shown in Figure 4.6. The diameters of the circles indicate the variances in the motion estimates for the points which were tracked across the video sequence. A plot of the covariance matrix is also shown in the same figure so that it is possible to compute the relative magnitudes of the errors. It was assumed that the noise in the feature points is statistically independent. The quality evaluation of the fusion algorithm was done using the distortion function of (2.44). Figure 4.7(a) plots the average distortion curve for 30 frames of the video sequence.

(a) (b)

Figure 4.5: Two frames of the original video sequence which is the input to the SfM reconstruction algorithm.

4.4.3 Reconstruction Example Without Generic Model

Figure 4.7(b) shows one particular view of the reconstructed model obtained after completion of the 3D reconstruction algorithm using SfM but without the generic model. The output, without texture mapping, of the multi-frame SfM algorithm is also shown in Figure 4.9(b), where the model is represented using a triangular mesh. The model shown is obtained after the registration process,

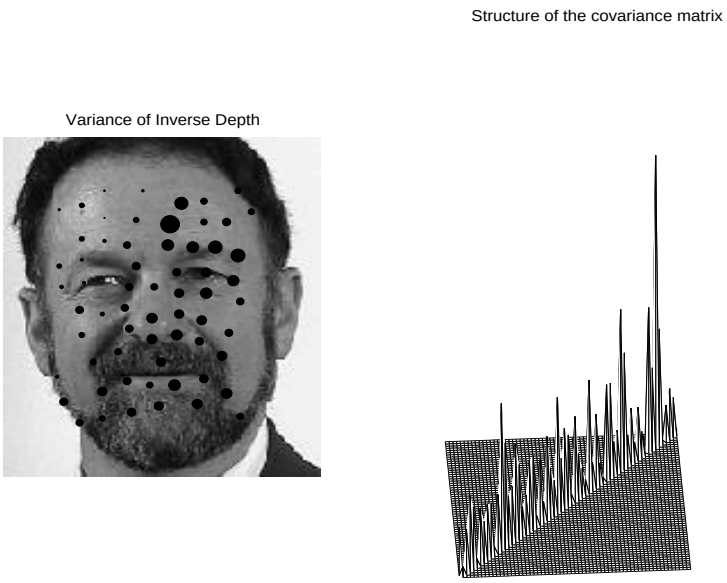
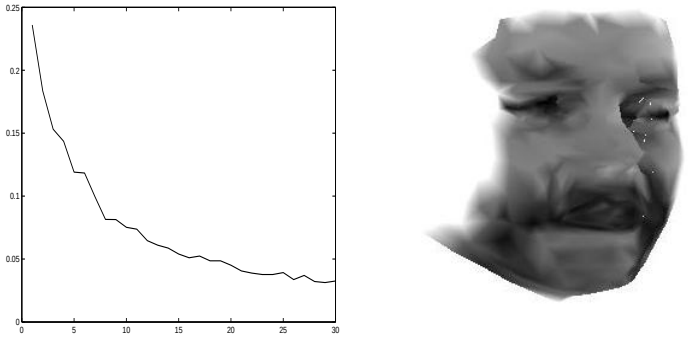


Figure 4.6: Plot of the variance of the inverse depth for different features in a face sequence. The diameter of the circle at each feature point is proportional to the variance at that feature point. In the second plot, the diagonal elements of $\mathbf{R_h}$ are shown.



(a) (b)

Figure 4.7: (a) represents the distortion of the SfM algorithm with the number of images; (b) depicts one view from the reconstructed model at this stage of the algorithm.

which was explained in Section 4.3.[5] It is evident from these plots that the general characteristics of the face are represented; however, it is also clear that a pure SfM algorithm is not enough for a completely satisfactory reconstruction of the face. We now introduce the generic model into the reconstruction strategy.

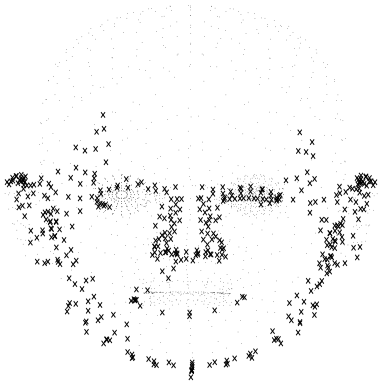


Figure 4.8: The vertices which form part of the line processes indicating a change in depth values are indicated with black 'x's.

4.4.4 The Line Process and Neighborhood Set

Figure 4.9(a) represents the generic model. The line process was calculated on the basis of this generic mesh. In Figure 4.8, the vertices of the generic mesh that indicate the boundaries between regions with sharp changes in depth are marked with black x's. For these vertices, $l_i = 1$ (in (4.8)). The local directional derivatives were calculated at each of the vertices of the generic mesh. The vertices at which there was a sharp change in the magnitude of the depth were selected to indicate that they belong to a line process. Thus, the line processes form the boundaries between regions having different depth values and divide the set of vertices into different equivalence classes.

[5]The ear region was not obtained from the SfM algorithm but was later stitched on for easy comparison with the other models.

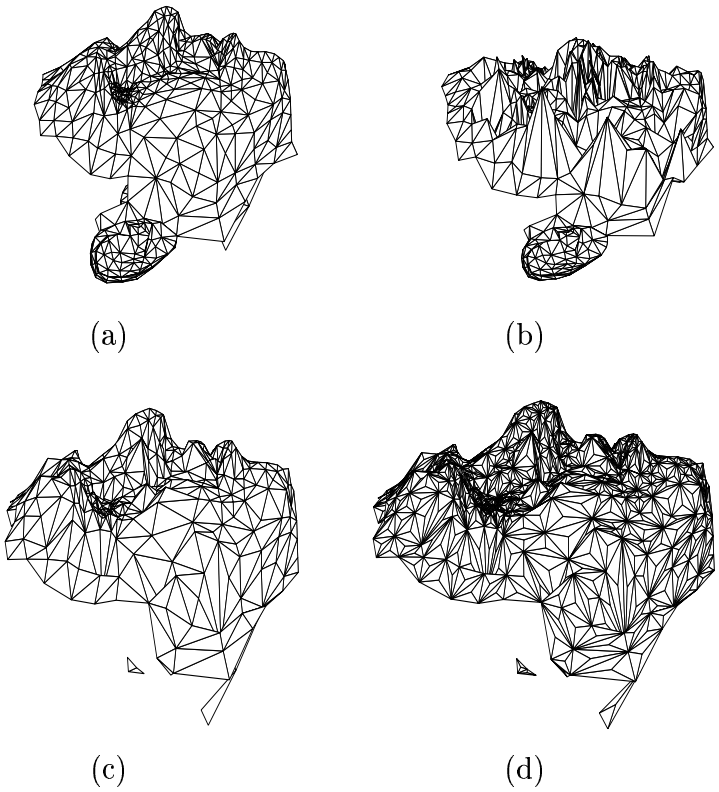


Figure 4.9: Mesh representations of the 3D models obtained at different stages of the algorithm. (a) represents the generic mesh, (b) the model obtained from the SfM algorithm (the ear region is stitched on from the generic model in order to provide an easier comparison between the different models), (c) the smoothed mesh obtained after the optimization procedure, (d) a finer version of the smoothed mesh for the purpose of texture mapping.

For each vertex, we need to identify a neighborhood set of vertices for the optimization function in (4.8). The vertices which are within a certain radial distance are identified as belonging to the neighborhood set of the central vertex. However, if a line process is encountered within this region, only those vertices which are in the same equivalence class as the central one are retained in the neighborhood. Since the entire process of determining the line processes and neighborhood sets is done on the generic mesh, it need not be done separately

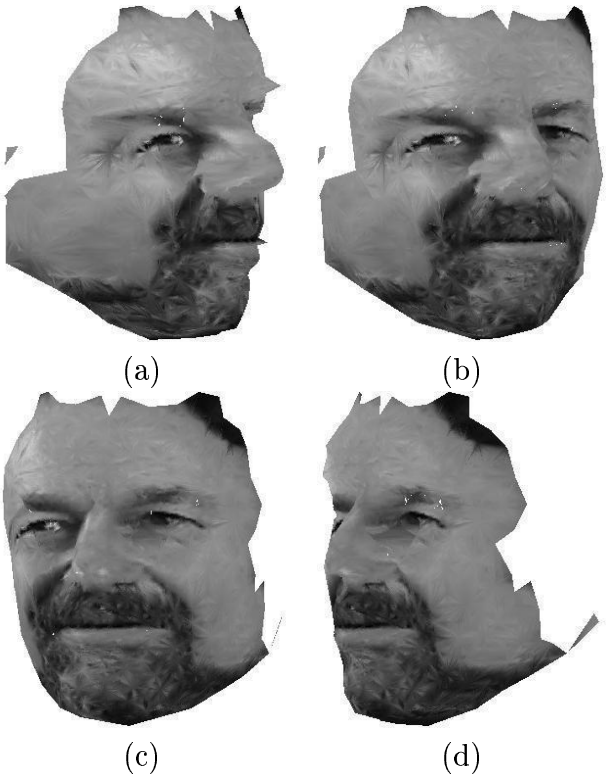


Figure 4.10: Different views of the 3D model after texture mapping.

for each 3D model.

4.4.5 The Optimization Procedure

The combinatorial optimization function in (4.8) was implemented using the simulated annealing procedure based on a Metropolis-Hastings sampler. At each temperature we carried out 100 iterations and this was repeated for a decreasing sequence of 20 temperatures. Although this is much below the optimal annealing schedule suggested by Geman and Geman [68] (whereby the temperature T_k should decrease sufficiently slowly, as $\mathcal{O}(\log(\sum_{i=1}^{k} N_i)^{-1}$, N_i being the total number of iterations at temperature T_i), it does give a satisfactory result for our face modeling example. We used a value of $\mu = 0.7$ in (4.8). The final smoothed

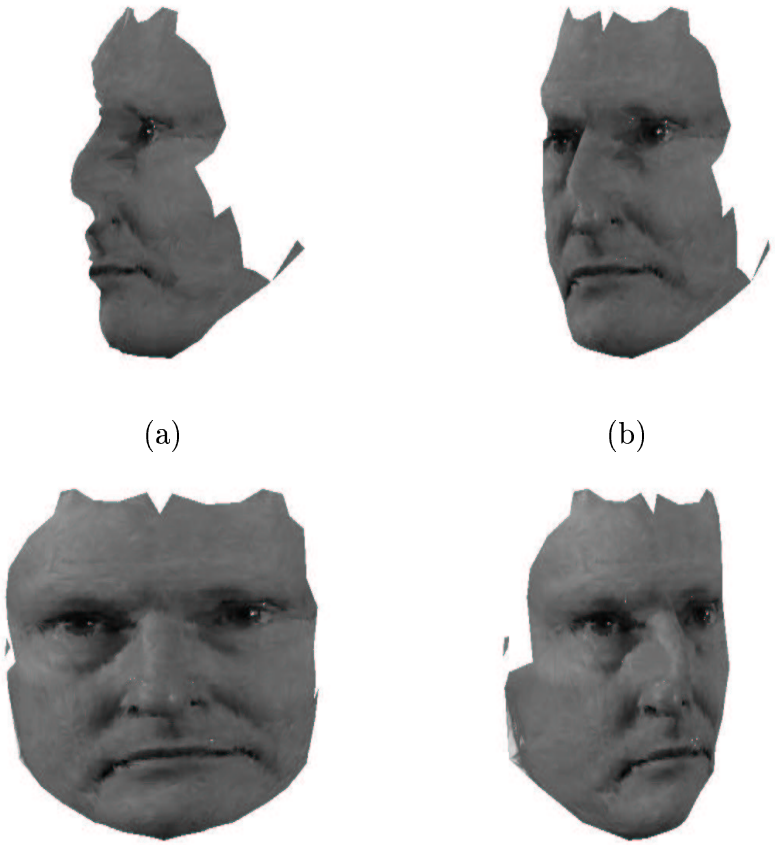
model is shown in Figure 4.9(c).

(a) (b)

Figure 4.11: Two frames from the second video sequence to which we applied our algorithm.

4.4.6 Texture Mapping

Next, we need to map the texture onto the smoothed model in Figure 4.9(c). Direct mapping of the texture from the video sequence is not possible since the large size of the triangles smears the texture over its entire surface. In order to overcome this problem, we split each of the triangles into smaller ones. This is done only at the final texture mapping stage. The initial number of triangles is enough to obtain a good estimate of the depth values, but not to obtain a good texture mapping. This splitting at the final stage helps us save a lot of computation time, since the depth at the vertices of the smaller triangles is obtained by interpolation, not by the optimization procedure. The final mesh onto which the texture is mapped is shown in Figure 4.9(d). Different views of the 3D model after the texture mapping are shown in Figure 4.10.



(a) (b)

(c) (d)

Figure 4.12: Different views of the 3D model after texture mapping on the second video sequence.

4.4.7 Face Modeling: Different Examples

We have applied our algorithm to several video sequences. We present here the results on two video sequences different from the one on which the detailed experimental results have been shown. Figure 4.11 shows two frames from the second video sequence. The set of procedures described above were carried out for this example also. Since the line processes and the neighborhood set are calculated from the generic model, the pre-computed results from the previous model were used for this experiment also. Figure 4.12 shows four views of the

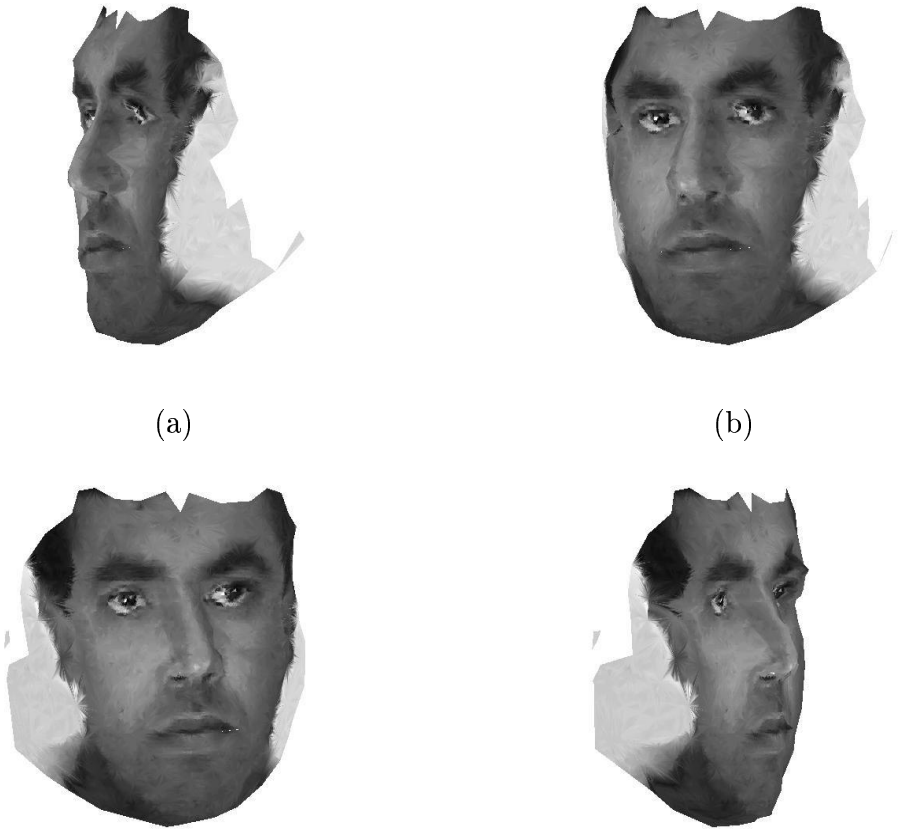
(a) (b)

Figure 4.13: Two frames from the third video sequence to which we applied our algorithm.

final 3D model reconstructed from this video sequence. Two frames of the third video sequence on which we present our results in this thesis are shown Figure 4.13. Projections from the 3D model reconstructed using our algorithm are shown in Figure 4.14.

4.5 Conclusions

In this chapter, we have presented a novel method of 3D modeling of a face from a video sequence using an SfM algorithm and a generic face model. In previous approaches, the generic model was used to initialize the SfM algorithm. The problem with this approach was that the final solution often converged very close to the initial value, resulting in a reconstruction which had the characteristics of the generic model rather than those of the particular face in the video which needs to be modeled. The main contribution of our work lies in the fact we incorporated the generic model *after* the SfM algorithm, which obtains the 3D estimate purely from the input video sequence. Also, instead of combining the depth values of the two models, we used an optimization framework whereby



(a) (b)

(c) (d)

Figure 4.14: Different views of the 3D model after texture mapping on the third video sequence.

the local *trends* in the 3D structure between the two models are compared and errors in the specific model are corrected for. The 3D structure estimation process was based on fusing the estimates obtained from pairs of frames from the video sequence, after computing the uncertainties of the two-frame solutions. The quality of the fusion algorithm was tracked using a distortion function. In order to combine the generic model with this 3D estimate, we used an energy function minimization procedure. Optimization was done using a Metropolis-Hastings sampling strategy. The results of our method at different stages of the

algorithm were presented.

Chapter 5

Evaluating the Quality of 3D Reconstructions

5.1 Introduction

In Chapters 2 and 3 we analyzed methods of quantifying the errors in 3D reconstruction from a video sequence and derived expressions for the first and second order statistics of the errors. We developed a robust 3D face reconstruction algorithm in Chapter 4. We now turn our attention to automatically evaluating the quality of 3D reconstructions in the most general setting.

The accuracy of SfM solutions is limited by various factors which can be broadly classified into inherent geometric indeterminacies [2],[3] and statistical inaccuracies [21],[23],[27]. Our work deals with the statistical aspect of the error in the 3D estimates. The main reason for the unacceptable quality of the reconstructions is the poor quality of the input images and the lack of robustness in reconstruction algorithms to deal with this issue [1, 6]. Therefore, many application systems process more images than necessary, hoping to minimize the effect of the errors because of the redundancy in the processed input data. For such cases, in order to obtain an optimal 3D reconstruction system, it is important to understand how the quality of the 3D estimates is affected by the number of

images processed. Is it possible to obtain a quantitative measure of the quality as a function of the number of images and to recognize situations where the input data is so poor that it is not possible to obtain a 3D estimate of the desired fidelity?

This is the question this chapter will address. We pose the SfM problem in a classical information-theoretic framework and propose a cost function for quality evaluation based on computing the mutual information (MI) between the scene structure and its estimates. We track the change in MI, which we term incremental MI (IMI), with an increasing number of input images. The underlying idea is the following: as more images are considered, the change in the MI between the estimate obtained from these images and the scene structure decreases. We propose methods for estimating the MI using statistical sampling techniques. Using the example of reconstructing a scene from video using optical flow [1, 7] under a Gaussian noise distribution, we show how the incremental MI can be computed from first principles in terms of the input parameters.

This chapter is organized as follows. We start with a brief survey of the use of information theory in computer vision. Section 5.2 provides a formal problem description. Section 5.3 introduces the incremental MI criterion and provides a motivation for its use. We also show how it can be computed in the most general setting using Monte-Carlo sampling techniques. In Section 5.4, we consider an example of reconstructing a 3D scene with video corrupted by Gaussian noise and derive the incremental MI from first principles. Finally, in Section 5.5, we provide the results of experiments on both simulated and real data.

5.1.1 Information Theoretic Concepts in Image and Video Processing

Recently, information-theoretic concepts have been used in various problems in image processing and computer vision, like image registration [76], object recognition [77],[78], and feature extraction and clustering [79],[80]. One of the earliest applications of MI in vision was in the performance evaluation of relaxation labeling algorithms [81]. In [76], the authors propose a method for aligning two images by maximizing the MI between them and use a stochastic optimization algorithm to perform the maximization. The underlying continuous pdfs (probability distribution functions) were represented using Parzen window densities [82]. In [77], the MI (termed "transinformation") was used to optimally place receptive fields over the object of interest. This was extended to include sequential decision processes in [78]. A slightly different technique using the "average loss of entropy" was used in [83], [84] for viewpoint selection. In the area of feature extraction, an information-theoretic approach using Fano's inequality for the error rate in classification was proposed in [79]. Information theory was used in clustering and other pattern recognition problems by Watanabe [85], [86] and a few other authors [80], [87]. In [80], the authors developed a clustering algorithm based on a sample-by-sample estimate of Renyi's entropy [88].

We are not aware of any previous work on the use of information-theoretic ideas for the quality evaluation of 3D reconstruction algorithms from video. The closest reference we can draw to our work is the Geometric Information Criterion (GIC) of Kanatani [89], which deals with model selection for geometric data. Model order selection is an important area of research in statistics [90]. Among the earliest and most influential ideas in this area are the Bayesian information

criterion (BIC) [91], the Akaike information criterion (AIC) [92], and the principle of Minimum Description Length (MDL) [93], which discriminate between competing models based on the complexity of their descriptions. The idea of fitting models to geometric data was formalized by Kanatani using a Geometric Information Criterion (GIC) [89]. We will show later that our criterion, the incremental mutual information, for evaluating the quality of 3D reconstructions is related to the idea of reduction of uncertainty in the reconstructions, which, in turn, is conceptually related to the MDL principle.

5.2 Problem Formulation

Theoretically speaking, it is possible to solve for the scene structure and camera motion from two images of the scene [34]. From (2.2) and (2.4) in Section 2.2, we see that for N corresponding points in two frames, we can write $2N$ equations relating the horizontal and vertical components of the image plane motion at each point to the depth at the point and the camera motion between the two frames. The number of unknowns is $N+5$: the depths at N points, three camera rotation parameters and two camera translation parameters (since we can get only the translation direction because of the scale ambiguity [34]). Thus it is possible to solve for the unknowns from the motion equations in a least-squares framework.

Since the motion between adjacent frames of a video sequence is usually small, the SfM equations based on motion estimates from optical flow [7] are typically valid. However, since the motion is small, even a small amount of error in the motion estimates can lead to large errors in the structure estimates. This is the classical low signal to noise ratio case in signal processing. In our experiments,

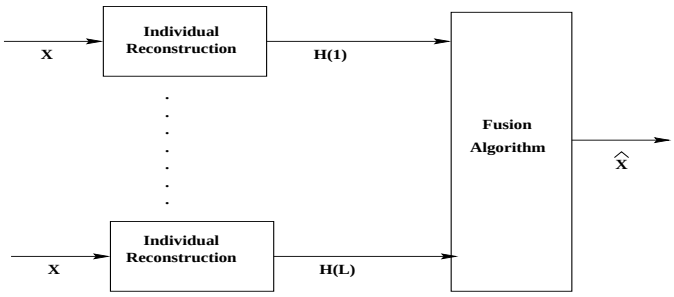


Figure 5.1: Block diagram representation of the reconstruction framework. $\mathbf{X}$ is the inverse depth that we want to estimate, $(\mathbf{H}(1), ..., \mathbf{H}(L))$ are the intermediate reconstructions (e.g. from pairs of frames), and $\hat{\mathbf{X}}$ is the final fused estimate.

we have observed that the error can often be as large as (sometimes even larger than) the actual motion between two corresponding points. Hence, in order to obtain accurate solutions to 3D structure estimation problems, it is necessary to understand the nature of these errors and their effects. In the previous chapters, we studied the first-order and second-order statistical effects of these errors. We also explored the use of robust estimation techniques from statistical approximation theory to deal with errors that cannot be suitably modeled, i.e. outliers. We showed (and so have many other authors [6],[51]) that one of the ways to reduce the effects of these errors is to integrate the estimates obtained from pairs or triples of frames over the entire video sequence. In this chapter, we try to understand how the quality of the final reconstruction is affected by the number of images in the video sequence. We pose the SfM reconstruction problem in an information-theoretic framework and use the mutual information between the unknown scene structure and the 3D estimate to get a precise idea of the quality of the reconstruction.

5.2.1 Notation

Figure 5.1 is a block-diagram representation of the 3D structure estimation algorithm. $\{\mathbf{H}(i), i = 1, ..., L\}$ represents the inverse depths [1] from individual reconstructions, which in our case are the structure estimates from pairs of frames from the video sequence. We assume that all the depth values are aligned to a common frame of reference. Feature points will be represented by subscripts, separate reconstructions will be within parentheses. Thus $H_i(k)$ represents the estimate of the i^{th} feature point for the k^{th} reconstruction. Unless required for purposes of clarity, the subscript will often be omitted from the notation. The vector of estimates of the inverse depth $[H_i(1), ..., H_i(N)]'$ will be denoted by $\mathbf{H}_i^{(N)}$. The boldface notation $\mathbf{H}(i)$ will represent all the features in the i^{th} reconstruction. The final estimate $\hat{\mathbf{X}}$ of $\mathbf{X} = [X_1, ..., X_M]'$ is obtained by fusing the individual reconstructions $(\mathbf{H}(1), ..., \mathbf{H}(L))$. Our analysis will assume that the noise in the feature points is independent and each of them will be treated separately. Hence, we will use the notation $\mathbf{H}^{(N)}$ to denote all the reconstructions for a particular feature point, which we do not represent explicitly. Similarly, X will represent the inverse depth at a particular unspecified point.

5.2.2 System Model

We assume that the individual estimates are corrupted by additive noise, i.e.

$$H(i) = X + V(i), \tag{5.1}$$

[1] The inverse depth is used throughout this chapter since it is the quantity that is estimated from the SfM equations for reconstruction from optical flow and its statistics can be obtained in an analytic form more easily than the statistics of the depth.

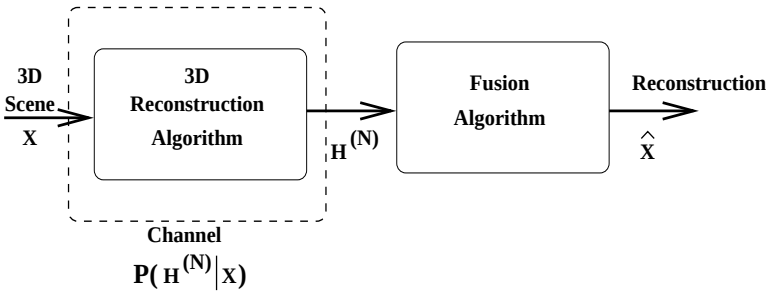


Figure 5.2: A channel model representation of the 3D reconstruction framework. The channel is characterized by the probability distribution function $P(\mathbf{H}^{(N)}|X)$.

where X is the inverse depth value of the particular feature. A more abstract representation of Figure 5.1 is shown in Figure 5.2, where the 3D reconstruction strategy is represented in a channel model. The input to the channel is the unknown 3D scene in the form of a video sequence. The output is the sequence of inverse depths of the scene (aligned to a particular frame of reference), represented by $\mathbf{H}^{(N)}$. The channel is a conceptual representation of the 3D reconstruction strategy comprising the video sequence, the correspondence algorithm, and the two-frame SfM algorithm. It is characterized by the probability distribution function $P(\mathbf{H}^{(N)}|X)$, which is assumed to be known. If the components of $\mathbf{H}^{(N)}$ are statistically independent, $P(\mathbf{H}^{(N)}|X) = \prod_{i=1}^{N} P(H(i)|X)$. In a later section, we will show how the channel characteristic can be estimated in terms of known parameters of the input video sequence.

The fusion algorithm is treated as a post-processing stage, separate from the channel. From Figure 5.2, it is clear that X, $\mathbf{H}^{(N)}$ and $\hat{X}$ form a Markov chain, i.e. $X \rightarrow \mathbf{H}^{(N)} \rightarrow \hat{X}$. Representing by $I(X,Y)$, the mutual information between two random variables X and Y, we can use the data processing inequality [94] and obtain

$$I(X,\hat{X}) \leq I(X,\mathbf{H}^{(N)}). \tag{5.2}$$

This allows us to use the mutual information between an unknown scene structure and its intermediate estimates as a criterion for evaluating the reconstruction quality, since we are assured that the mutual information between the final reconstruction and the actual scene depth will always be lower than or equal to it.

5.3 Incremental Mutual Information

Consider the channel model representation of the reconstruction strategy in Figure 5.2 and the data processing inequality of (5.2). A typical representation of the mutual information $I(X, \hat{X})$ and $I(X, \mathbf{H}^{(N)})$ is shown in Figure 5.3, which is a diagrammatic representation of the data processing inequality as a function of the number of frames, n.

Most algorithms address the issue of evaluating the quality of a reconstruction by considering the error covariance of the final estimate. The usual practice is to estimate it from the data. In Chapter 2, we showed that it is possible to obtain the error covariance analytically when we use the optical flow equations for the motion, which is usually a valid assumption when considering a monocular video sequence. In Chapter 3, we showed that it is not enough to consider the error covariance only because the estimate of the structure from optical flow is statistically biased, even when the camera motion is exactly known. However, the bias and covariance capture the first-order and second-order characteristics of the error in reconstruction. The advantage of using a criterion based on mutual information is that it is able to take into account the effect of the entire probability distribution, e.g. the effect of outliers, which is usually manifest in the higher-order statistics.

The data processing inequality allows us to evaluate the quality of the reconstruction even before the final estimate, $\hat{X}$, has been obtained. This enables us to understand the effect of intermediate reconstructions and the fusion strategy separately. Since our evaluation criterion is based on $I(X, \mathbf{H}^{(N)})$, we can decide whether considering more images from the video sequence will add to the quality of the final reconstruction. Thus it is possible to monitor the progress of a multi-frame 3D reconstruction algorithm as it proceeds by using more and more images.

Our criterion for evaluating the quality of the reconstruction depends on estimating the difference in mutual information for the two sets of observations, $\mathbf{H}^{(N)}$ and $\mathbf{H}^{(N-1)}$. We term this the *incremental mutual information*, i.e.

$$\Delta I(N) = I(X, \mathbf{H}^{(N)}) - I(X, \mathbf{H}^{(N-1)}). \tag{5.3}$$

The term gives us an idea of the contribution of the N^{th} observation to the reconstruction strategy with respect to the previous $(N-1)$ observations. As the number of observations increases, the effect of an additional observation decreases and approaches zero in the limit.

Using the relationship between mutual information and entropy, it is possible to obtain a different interpretation of the incremental mutual information. Denoting by $h(X)$ the entropy of the random variable X, we know that [94]

$$\begin{aligned} I(X;Y) &= h(X) - h(X|Y) \\ &= h(Y) - h(Y|X) \\ &= h(X) + h(Y) - h(X,Y). \end{aligned} \tag{5.4}$$

Thus $\Delta I(N)$ in (5.3) can be written as

$$\Delta I(N) = I(X;\mathbf{H}^{(N)}) - I(X;\mathbf{H}^{(N-1)})$$

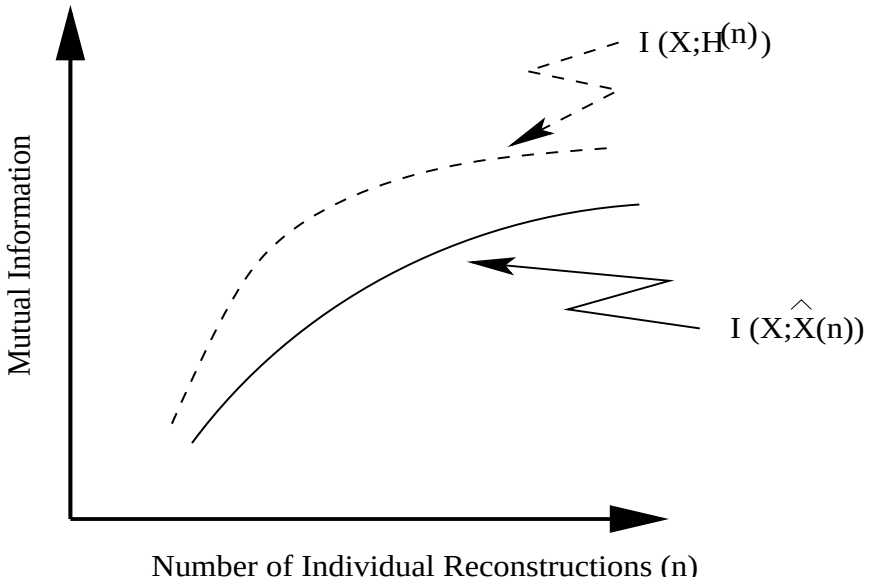


Figure 5.3: A typical plot of the mutual information in the data processing inequality of (5.2).

$$= \quad h(X|\mathbf{H}^{(N-1)}) - h(X|\mathbf{H}^{(N)}). \tag{5.5}$$

The quantity defined as the incremental mutual information can also be referred to as the incremental conditional entropy. Since the entropy of a random variable is a measure of its uncertainty, ΔI measures the reduction in the uncertainty as we add an extra observation. Since the incremental mutual information tends to zero in the limit, the difference in the conditional entropy also approaches zero. Thus we will consider more and more images from the video sequence until the uncertainty in the final structure estimate can be reduced no further. This is the intuitive idea behind our criterion in (5.3).

The rate at which the incremental mutual information decreases is also an important measure of the progress of the algorithm. An extremely slow rate of fall indicates that more images will be necessary to achieve an acceptable level of quality. Since there is motion between adjacent frames of the video, a

particular point will move out of the field of view of the camera after a certain amount of time. A very slow rate of fall of ΔI might mean that the quality of the reconstruction is not good enough even when the point is no longer visible. The rate of change of ΔI can be obtained as

$$\begin{aligned} \Delta^2 I(N) &= \Delta I(N) - \Delta I(N-1) \\ &= I(X, \mathbf{H}^{(N)}) + I(X, \mathbf{H}^{(N-2)}) - 2I(X, \mathbf{H}^{(N-1)}). \end{aligned} \tag{5.6}$$

Combining (5.3) and (5.6), we can state that an acceptable reconstruction quality has been achieved when both the following conditions are satisfied simultaneously:

$$\begin{aligned} \Delta^2 I(N) &\leq 0, \quad \forall N > N_0, \\ \Delta I(N) &< \tau, \end{aligned} \tag{5.7}$$

where N_0 is a constant and τ is a threshold defining an acceptable quality of reconstruction. Since $\Delta I(N)$ is monotone non-increasing for $N > N_0$ and is bounded below by zero, the monotone convergence theorem [33] applied to (5.5) implies that $h(X|\mathbf{H}^{(N-1)}) \to h(X|\mathbf{H}^{(N)}) \to h_0$ for some $N > N_0$, in the case where an acceptable quality of reconstruction has been reached. Thus, h_0 is the minimum level of uncertainty in a scene described by N observations.

5.3.1 Estimating the Mutual Information

We now turn our attention to estimating the IMI from the data. This requires a knowledge of the probability density functions of the random variables, which we do not know *a priori* and have to estimate from samples. The entropy of a random variable z can be expressed as the expectation of the negative logarithm

of the probability density $p(z)$, i.e.

$$h(z) = E_z[-\ln p(z)]. \tag{5.8}$$

Thus, if we can estimate the probability densities we can obtain the MI using (5.4).

We assume that the channel characteristic, $P(\mathbf{H}^{(N)}|X)$, is known. Using the observation model of (5.1) and assuming that the noise process $\{V(i)\}_{i=1}^{N}$ is independent of X, we can write

$$\begin{aligned} P(\mathbf{H}^{(N)}|X) &= P(\{X + V(i)\}_{i=1}^{N}|X) \\ &= P(\mathbf{V}^{(N)}). \end{aligned} \tag{5.9}$$

Thus knowledge of the channel characteristic implies that we know the joint distribution of the noise process. If $\{V(i)\}$ is an independent sequence of random variables, the joint distribution is simply the product of the noise distributions in the individual reconstructions. In the next section, we show by an example how the channel characteristic can be estimated from first principles starting with the basic equations of SfM from optical flow. Alternatively, the noise process can be assumed stationary and the probability distribution estimated from the initial few frames using histogram techniques. A method of estimating the probability distributions and mutual information using statistical sampling techniques can be found in [95].

Once $P(\mathbf{H}^{(N)}|X)$ is known, we can obtain

$$\begin{aligned} P(\mathbf{H}^{(N)}) &= \int_{\mathcal{X}} P(\mathbf{H}^{(N)}|X) p_X(x) dx \\ &\approx \sum_{x_i \in \mathcal{X}} P(\mathbf{H}^{(N)}|x_i) p_X(x_i), \end{aligned} \tag{5.10}$$

where $p_X(x_i)$ is the probability that the random variable $X = x_i$. Knowing $p_X(x)$ implies that we have an *a priori* statistical model on the scene structure X.

Expressing the MI in terms of the entropies, we can write

$$I(X, \mathbf{H}^{(N)}) = h(\mathbf{H}^{(N)}) - h(\mathbf{H}^{(N)}|X). \tag{5.11}$$

Using $P(\mathbf{H}^{(N)}|X)$ and $P(\mathbf{H}^{(N)})$, we can compute (5.11) by estimating the entropies using the law of large numbers [36]. The expected value of a random variable $f(Z)$ can be computed by sampling z_i from the distribution $P(z)$ and computing

$$E_Z\left[f(Z)\right] = \frac{1}{n}\sum_{i=1}^{n} f(z_i). \tag{5.12}$$

This can be used to compute the entropies from (5.8).

5.4 A Case Study: Reconstructing With Gaussian Noise

In this section, we consider the special case of Gaussian noise in the motion estimates. We show that for this case we can derive a closed-form expression for the IMI, as opposed to the Monte Carlo simulations necessary for the general case.

Recall equation (2.4). Also recall that we derived an expression for the error covariance in the structure and motion estimates, $\mathbf{R_z}$ in (2.23), which was partitioned to obtain $\mathbf{R_h}$. The IMI can be expressed in terms of $\mathbf{R_h}$ under the Gaussian noise assumption.

Using our previous notation as in (5.1), let X be the unknown true inverse depth. Assume that $X \sim \mathcal{N}(0, \sigma_x^2 = P_X)$[2] and $\{V(i), i = 1, ..., N\}$ is

[2]The mean of X is subtracted out.

a sequence of independent random variables distributed as $\mathcal{N}(0, \sigma^2_{V(i)})$. Let $P_V = \text{diag}\,[P_V(i)]_{i=1,...,N} = \text{diag}\left[\sigma^2_{V(1)}, ..., \sigma^2_{V(N)}\right]$. Now $\text{Cov}[H(i)|X] = \mathbf{R_h}(i)$ from (2.36). Thus $P_V = \text{diag}\,[\mathbf{R_h}(1), ..., \mathbf{R_h}(N)]$, where $\mathbf{R_h}(i)$ is the value of $\mathbf{R_h}$ at a particular point for the inverse depth obtained from the i^{th} and $(i+1)^{\text{st}}$ frames.

From (5.1), $E[H(i)] = 0$ and

$$\begin{aligned} E[H(i)H(j)] &= E[(X + V(i))(X + V(j))] \\ &= P_X + P_V(i)\delta_{ij}, \end{aligned} \tag{5.13}$$

where δ_{ij} is a Kronecker delta function. Thus the covariance of $\mathbf{H}^{(N)}$ is $P_{\mathbf{H}^{(N)}} = P_V^{(N)} + \mathbf{1}_N P_X \mathbf{1}_N^T$, where $\mathbf{1}_N$ is a vector of N 1's. Using the fact that the entropy (differential) of a Gaussian random variable $Z \sim \mathcal{N}(0, \Sigma)$ is $\frac{1}{2}\log(2\pi \exp \Sigma^{-1})$ [94], the mutual information between X and $H(i)$ is

$$\begin{aligned} I(X; H(i)) &= h(H(i)) - h(H(i)|X) \\ &= \frac{1}{2}\log\left(1 + \frac{P_X}{P_V(i)}\right). \end{aligned} \tag{5.14}$$

Next, consider the mutual information between the unknown X and the vector of observations $\mathbf{H}^{(N)}$. We will denote by $|K|$ the determinant of a matrix K.

$$\begin{aligned} I(X; \mathbf{H}^{(N)}) &= h(\mathbf{H}^{(N)}) - h(\mathbf{H}^{(N)}|X) \\ &\overset{(a)}{=} h(\mathbf{H}^{(N)}) - \sum_{i=1}^{N} \frac{1}{2}\log(2\pi e P_V(i)) \\ &\overset{(b)}{=} \frac{1}{2}\log\left(\frac{|P_V + \mathbf{1}_N P_X \mathbf{1}_N^T|}{|P_V|}\right). \end{aligned} \tag{5.15}$$

(a) is the result of applying the chain rule of entropy and substituting the expression for the differential entropy of a Gaussian random variable [94]; (b) is due to the fact that $|P_V| = \prod_{i=1}^{N} P_V(i) = \prod_{i=1}^{N} \sigma^2_{V(i)}$. Using the method of induction and the properties of determinants, it can be shown that $|P_V + \mathbf{1}_N P_X \mathbf{1}_N^T| =$

$\prod_{i=1}^{N} \sigma_{V(i)}^2 + \sigma_x^2 \sum_{i=1}^{N} \prod_{\substack{j=1 \\ j \neq i}}^{N} \sigma_{V(j)}^2$ (see Appendix B). Then from (5.15), the expression for the mutual information becomes

$$I(X; \mathbf{H}^{(N)}) = \frac{1}{2} \log \left(1 + \sum_{i=1}^{N} \frac{\sigma_X^2}{\sigma_{V(i)}^2} \right). \tag{5.16}$$

Thus, the incremental mutual information $\Delta I(N)$ is

$$\begin{aligned}
\Delta I(N) &= I(X; \mathbf{H}^{(N)}) - I(X; \mathbf{H}^{(N-1)}) \\
&= \frac{1}{2} \log \left(\frac{|P_{V^{(N)}} + \mathbf{1}_N P_X \mathbf{1}_N^T|}{|P_{V^{(N-1)}} + \mathbf{1}_{N-1} P_X \mathbf{1}_{N-1}^T|} . \frac{|P_{V^{(N-1)}}|}{|P_{V^{(N)}}|} \right) \\
&= \frac{1}{2} \log \left(\frac{\prod_{i=1}^{N} \sigma_{V(i)}^2 + \sigma_x^2 \sum_{i=1}^{N} \prod_{\substack{j=1 \\ j \neq i}}^{N} \sigma_{V(j)}^2}{\prod_{i=1}^{N} \sigma_{V(i)}^2 + \sigma_x^2 \sum_{i=1}^{N-1} \prod_{\substack{j=1 \\ j \neq i}}^{N} \sigma_{V(j)}^2} \right) \\
&= \frac{1}{2} \log \left(1 + \frac{1/\sigma_{V(N)}^2}{\frac{1}{\sigma_x^2} + \sum_{i=1}^{N-1} \frac{1}{\sigma_{V(i)}^2}} \right) \\
&= \frac{1}{2} \log \left(1 + \frac{1/P_V(N)}{\frac{1}{\sigma_x^2} + \sum_{i=1}^{N-1} \frac{1}{P_V(i)}} \right).
\end{aligned} \tag{5.17}$$

Hence we are able to obtain a closed-form expression for $\Delta I(N)$ in terms of the parameters of the input video sequence by starting from the basic equations of 3D reconstruction from optical flow.

5.4.1 An Estimation-Theoretic Interpretation:

Since we have considered the case of Gaussian noise, it is possible to give an alternative interpretation to the results in (5.17) from an estimation-theoretic perspective. The mean squared distortion for M feature points is defined as

$$D(\mathbf{X}, \hat{\mathbf{X}}) = \frac{1}{M} \sum_{j=1}^{M} E[(X_j - \hat{X}_j)^2]. \tag{5.18}$$

Let $p(X_j, H_j(1), ..., H_j(N))$ denote the joint density function of the parameter and observations. The mean square error estimator $\hat{X}_j$ of X_j, obtained from

$\mathbf{H}^{(N)}$, is $\hat{X}_j(N) = E\left[X_j | H_j^{(N)}\right]$. From the Cramer-Rao lower bound (CRLB) we can write the following set of inequalities:

$$\begin{aligned} D &\geq \frac{1}{M}\sum_{j=1}^{M} \frac{1}{E\left[-\frac{\partial^2}{\partial X^2}\log p(X_j, H_j(1), ..., H_j(N))\right]} \\ &= \frac{1}{M}\sum_{j=1}^{M} \frac{1}{\frac{1}{\sigma_{x_j}^2} + \sum_{i=1}^{N} E\left[-\frac{\partial^2}{\partial X^2}\log p(H_j(i)|X)\right]} \\ &\geq \frac{1}{\frac{1}{M}\sum_{j=1}^{M}\left(\frac{1}{\sigma_{x_j}^2} + \sum_{i=1}^{N}\frac{1}{P_{V_j}(i)}\right)} \\ &\triangleq \frac{1}{\frac{1}{M}\sum_{j=1}^{M}\frac{1}{D_j(N)}}. \end{aligned} \tag{5.19}$$

The last step is a result of the application of Jensen's inequality [35] and that $E\left[-\frac{\partial^2}{\partial X^2}\log p(H_j(i)|X)\right] = \frac{1}{P_{V_j}(i)}$. Recalling that (5.17) is for a particular feature point where the subscript has been suppressed for clarity of notation, let us denote $\Delta I_j \triangleq I(X_j; \mathbf{H}_j^{(N)}) - I(X_j; \mathbf{H}_j^{(N-1)})$. Then from (5.19) and the last expression in (5.17), we get

$$\Delta I_j = \frac{1}{2}\log\left(\frac{D_j(N-1)}{D_j(N)}\right). \tag{5.20}$$

Alternatively, the innovation at the N^{th} stage is $\gamma_N = X_N - \hat{X}_N$. Then following the standard derivation for the Kalman filter [35], it can be shown that the variance of the innovations is

$$P_{\gamma_N} = \sigma_{V(N)}^2\left(1 + \frac{1/\sigma_{V(N)}^2}{\frac{1}{\sigma_x^2} + \sum_{i=1}^{N-1}\frac{1}{\sigma_{V(i)}^2}}\right), \tag{5.21}$$

which shows that for each feature point, the incremental mutual information is related to P_{γ_N} by

$$\Delta I = \frac{1}{2}\log\left(\frac{P_{\gamma_N}}{\sigma_{V(N)}^2}\right). \tag{5.22}$$

5.5 Simulation Results

In this section we present the results of experiments that were carried out in order to analyze our criterion of incremental mutual information using both simulated and real data.

5.5.1 Experiment 1

A set of 3D points was generated so that their true positions were known. Perspective projections of these points were generated and Gaussian noise with zero mean and known variance was added to these 2D locations. The projections were taken for different positions of the camera, so that finally a set of tracked features was obtained. From every pair of such tracked features, the positions of the original 3D points were estimated, which resulted in a set of 3D reconstructions. The first plot of Figure (5.4) shows the true values of the 3D points and their estimated reconstructions from all the frames over which the features could be tracked. [3] The second diagram in Figure (5.4) plots the decrease in incremental mutual information with an increasing number of intermediate reconstructions.

5.5.2 Experiment 2

As in the previous simulation, a set of features were tracked over a number of frames. However, noise was added according to a uniform distribution. The magnitude of the noise was larger than in the previous experiment. This led to mismatches among some of the features. The 3D positions of the points were estimated using the SfM algorithm and some of the results were erroneous, as is

[3] The first point was used to set the scale of the reconstruction, so that the geometric indeterminacies do not affect the result.

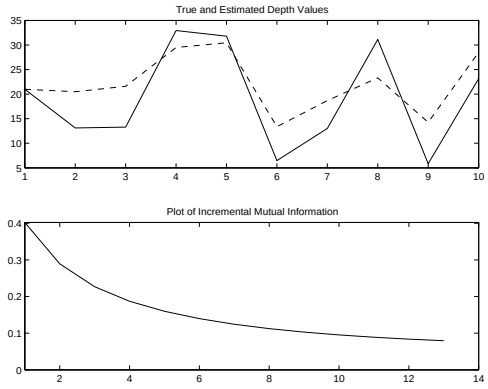


Figure 5.4: The upper plot shows the true depth values of the 3D points (the solid line) and the fused estimate from the intermediate reconstructions from all the frames (the dotted lines). The lower plot shows the decrease in the incremental information with increasing number of frames.

clear from the first plot in Figure (5.5). The second plot in Figure (5.5) shows this case, where the incremental mutual information remains large and does not follow a steadily decreasing trend as in the previous example.

5.5.3 Experiment 3

We now present a result on a real video sequence. The video consists of a person moving his head in front of a static camera. The aim was to reconstruct the model of the head of the person from this video. The focal length of the camera was known. Figure (5.6)(a) represents an image from the video along with some of the feature points which were tracked. Figure (5.6)(b) represents the change in the incremental mutual information between the unknown 3D structure and the intermediate reconstructions from every pair of frames. The covariance of the error in the intermediate reconstructions was estimated using (2.24) and (2.36). This was used to estimate the incremental mutual information using

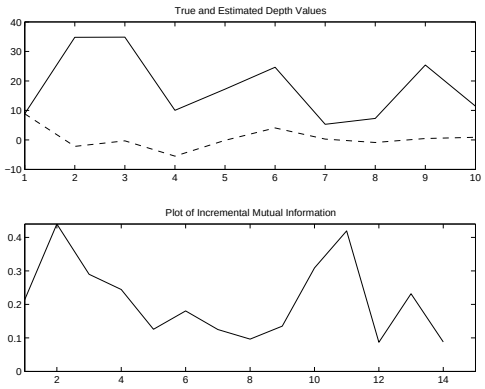
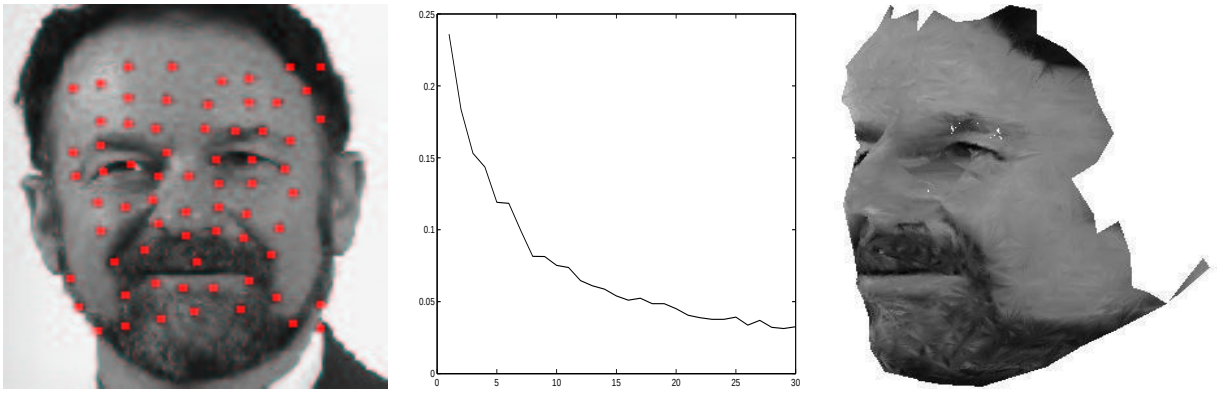


Figure 5.5: The upper plot shows the true depth values of the 3D points (the solid line) and the fused estimate from the intermediate reconstructions from all the frames (the dotted lines). The lower plot is the change in the mutual information with increasing number of frames. This is the case where the estimated reconstruction does not converge to the true value even with an increasing number of observations.



(a) (b) (c)

Figure 5.6: The above figures represent a 3D reconstruction from video using the method of measuring the incremental mutual information to judge the quality of the result. (a) is one of the images from the video along with the set of tracked features used for the reconstruction. (b) represents the change in the incremental mutual information with the number of images; (c) depicts one view from the reconstructed model.

(5.17). Based on this measure, the 3D model was reconstructed from 25 frames using the algorithm described in Chapter 4. Figure (5.6)(c) shows one view of this model.

5.6 Conclusions

In this chapter we introduced a method of evaluating the quality of 3D reconstructions from video sequences in information-theoretic terms. We showed that the 3D reconstruction problem using multi-frame SfM can be represented using a channel model, where the channel characteristic can be estimated from the input parameters of the video. Such a conceptual representation allows us to derive a criterion for evaluating the reconstruction by computing the change in the mutual information between the unknown scene structure and the 3D estimates obtained from increasing numbers of images from the video sequence. Since we can obtain the mutual information using Monte Carlo simulations, our criterion goes beyond the second-order distortion estimates which have been the standard evaluation criterion for 3D reconstruction algorithms. Through an example, we showed how it is possible to obtain analytical expressions for incremental mutual information in terms of the parameters of the feature points tracked across the video sequence. Finally, we carried out experiments on simulated and real data and the results were presented.

Chapter 6

Registration of Partial 3D Models

Thus far, we have concentrated on obtaining 3D models from monocular video sequences. Complete 3D models of a scene are usually created by stitching together separate partial models obtained from different views. This process requires registration of features in different partial representations. Establishing correspondence of features between two or more images obtained from different views of the same object is still a challenging problem. The difficulty of the problem lies in the fact that the images may be obtained under different conditions of lighting and camera settings. In this chapter, we propose a technique for registration of partial models of an object reconstructed from video. We show that prior information extracted from the video sequence used to obtain the partial 3D models, or from a similar sequence, can be used to design a robust correspondence algorithm. The prior information can be collected once for a class of objects and then used for different objects in that class. The method works by matching the 2D shapes of different features. A doubly stochastic matrix, representing the probability of match between the features, is derived using the Sinkhorn normalization procedure. The final correspondence is obtained by minimizing the probability of error of a match between the entire constellation

of features in the two sets, thus taking into account the global spatial structure of the object. The method is applied to create holistic 3D models of a face from partial representations.

Numerous methods have been used to attempt to solve the registration problem, ranging from techniques which take advantage of knowledge of the geometry of the scene to ones which use different information-theoretic measures to compute similarity. One of the best-known methods of registration is the iterative closest point (ICP) algorithm [96] of Besl and McKay. It uses a mean-square distance metric which converges monotonically to the nearest local minimum. It was used for registering 3D shapes by considering the full six degrees of freedom in the motion parameters. It has been extended to include Levenburg-Marquardt non-linear optimization and robust estimation techniques to minimize the registration error [97]. Another well-known method of registering 3D shapes is the work of Vemuri and Aggarwal, where they used range and intensity data to reconstruct complete 3D models from partial ones [98]. Registering range data for the purpose of building surface models of three-dimensional objects was also the focus of the work in [99]. Matching image tokens across triplets, rather than pairs, of images has also been considered. In [100], the authors developed a robust estimator for the trifocal tensor based upon corresponding tokens across an image triplet. This was then used to recover 3D structure. Reconstructing the 3D structure was also considered in [101] using stereo image pairs from an uncalibrated video sequence. However, most of these algorithms work only when given good initial conditions, e.g. for 3D model alignment, the partial models have to be brought into approximately correct positions. The problem of automatic "crude" registration (in order to obtain good initial conditions) was addressed

in [102], where the authors used bitangent curve pairs which could be found and matched efficiently.

In the above methods, geometric properties are used to align 3D shapes. Another important area of interest for registration schemes is 2D image matching, which can be used for applications like image mosaicking, retrieval from a database, medical imaging, etc. 2D matching methods rely on extracting features or *interest points*. In [103], the authors show that interest points are stable under different geometric transformations and define their quality based on repeatability rate and information content. One of the most widely used schemes for tracking feature points is the KLT tracker [104], which combines feature selection and tracking across a sequence of images by minimizing the sum of squared intensity differences over windows in two frames. A probabilistic technique for feature matching in a multi-resolution Bayesian framework was developed in [105] and used in uncalibrated image mosaicking. In [106], the authors introduced the use of Zernike orthogonal polynomials to compute the relative rigid transformations between images. It allows the recovery of rotational and scaling parameters without the need for extensive correlation and search algorithms. Precise registration algorithms are also required for medical imaging applications. A mutual information criterion, optimized using the simulated annealing technique, was used in [107] for aligning images of the retina.

Various probabilistic schemes have also been used for registration problems. One of the best-known techniques is the work of Viola and Wells for aligning 2D and 3D objects by maximizing mutual information [76]. The technique is robust with respect to the surface properties of objects and illumination changes. A stochastic optimization procedure was proposed for maximizing the mutual

information. A probabilistic technique for matching the spatial arrangement of features using shape statistics was proposed in [108]. Most of these techniques in image registration work for rigid objects. Constraints using intensity and shape usually break down for non-rigid objects. The problem of registering a sequence of images of a non-rigid observed scene was addressed in [109]. The images in the sequence were treated as samples from a multi-dimensional stochastic time series (e.g. an auto-regressive model) which is learned. This stochastic model can then be used to extend the video sequence arbitrarily in time.

The above methods for establishing correspondence rely, in essence, on matching image tokens across groups of images. However, extraction of such image tokens (like the intensities or shapes of significant features) is an inherently noisy process, and most methods of extracting them are error-prone. In addition, it is extremely difficult to compute quantities that are invariant under different imaging conditions; both intensity and shape, the two most easily obtainable characteristics in an image, are dependent on the viewing angle. In this paper, we show that the availability of data in the form of a video sequence can help in developing robust correspondence schemes. We also show that the incorporation of *proper* prior information, easily extracted from a spatio-temporal volume of video data, into the registration scheme can produce a robust algorithm for matching.

The method presented here works with the edge images of local features (which gives approximate notions of the 2D shapes of the features). A doubly stochastic matrix, representing the probability of match between the features, is obtained using Sinkhorn normalization [110] and the prior information. The method works by matching the entire constellation of features in the two sets

by minimizing the probability of error of a match, after taking into account the constraints on the relative configuration of the parts. The motivation for this global strategy (as opposed to the correspondence of individual features which are local to that region) is that it emphasizes the structural description of the object. Our matching technique also supports the identifications of missing features and occlusions between the two views.

Use of prior information about the shape adds robustness to the scheme. The prior information can be collected once for different classes of objects and used across different objects in that class; e.g., in our application to building holistic 3D face models, the prior information can be collected once from a video sequence of a particular person's face and used across a large number of faces with similar characteristics. Computation of the prior requires tracking features across the frames of a video sequence. Tracking algorithms usually work well when the motion between consecutive frames is small, but would perform very poorly in trying to match features from two viewing angles with a wide baseline, which is the case in our application [82]. Thus the extraction of the priors (which also requires correspondence) and the registration of views for 3D model alignment are not the same problem. The tracking algorithm, which works for small baselines, is taken advantage of to solve the wide-baseline correspondence problem. Also, since the prior needs to be extracted only once, a time-consuming method (even a manual one) can be used; however, the model alignment has to be automatic, as it needs to be done for every 3D model we create.

This chapter is organized as follows. In the next section, we present our method of computing the probabilities for matching the individual features. Section 6.2 explains how to incorporate the spatial structure of the object into the

matching scheme. The correspondence algorithm is described in Section 6.3. The results of our algorithm applied to the problem of creating holistic 3D models from partial ones is presented in Section 6.4.

6.1 3D Registration Using Prior Models

6.1.1 Obtaining the Partial Models

The first step toward creating the holistic 3D models is to obtain the partial models. Each of these partial models is obtained from a video sequence or from portions of a longer video sequence using structure from motion (SfM). Almost any method of reconstructing the 3D model from video can be used. For details, the interested reader may refer to [1], [3]. For our application, we used the method outlined in Chapter 4 to obtain each of the partial models.

6.1.2 Formulation of the Registration Problem

Our aim is to obtain correspondences between two sets of features, each extracted from one of the partial models and represented as sets of random variables, $\mathbf{X} = [X_1, ..., X_P]$ and $\mathbf{Y} = [Y_1, ..., Y_M]$. Each of the elements of the sets represents the collection of corners in a local region around the feature of interest, thus giving an idea of the 2D shape of the region (see Figures 6.5 and 6.6); hence we use the term *shape cues*. Though the shapes of different features are usually significantly different, and therefore easier to match, they are dependent on the viewing angle and the extraction process is extremely sensitive to noise. To overcome this, we use priors, which are the mean shape of each feature ("mean feature") collected from the video sequence over a range of viewing angles. Since

the shapes of the features do not vary drastically for different people, the prior information can be collected only once and used across different video sequences.

6.1.3 Computing the Feature Correspondence Probabilities

Let $\mu = \mu_1, ..., \mu_K$ represent the prior information of K features. Let H_i be the hypothesis that Y_i matches X; we wish to compute the *a posteriori* probability $P(H_i|X)$. Defining the event $\mathcal{E}_{X\mu_j} = \{X \text{ matches } \mu_j\}$, we hypothesize that the probability of X matching μ_j is directly proportional to the inner product of X with μ_j (since the inner product gives a measure of similarity). Since X and μ_j are binary images, the inner product will always be non-negative. Then

$$P(\mathcal{E}_{X\mu_j}|X = X_n) = \frac{1}{\sum_{j=1}^{K} < X_n, \mu_j >} < X_n, \mu_j > \tag{6.1}$$

where $< . >$ denotes inner product. For two images of size $P \times Q$, $< X_n, \mu_j >= \frac{1}{PQ} \sum_{p=1}^{P} \sum_{q=1}^{Q} X_n(p,q)\mu_j(p.q)$. Similarly, the probability that Y_i matches X given the event $\mathcal{E}_{X\mu_j}$ is proportional to the inner product of Y_i and μ_j,

$$P(H_i|X, \mathcal{E}_{X\mu_j}) = \frac{1}{\sum_{j=1}^{K} < Y_i, \mu_j >} < Y_i, \mu_j > . \tag{6.2}$$

Then, from the theorem of total probability, the *a posteriori* probability (which is the probability of X_n matching Y_i) is

$$P(H_i|X) = \sum_{j=1}^{K} P(H_i|X, \mathcal{E}_{X\mu_j}) P(\mathcal{E}_{X\mu_j}|X = X_n) \tag{6.3}$$

Maximizing this *a posteriori* probability is equivalent to minimizing the error of a match.

6.1.4 Prior Information

Assume that a feature $X_i(n)$ [1] is corrupted by independent, zero-mean, additive noise ν. Let

$$X_i(n) = S_i(n) + \nu_i(n), \qquad n = 1, ..., L. \tag{6.4}$$

where $S_i(n)$ is the true unknown value of the feature. Then $\mu_i = E[X_i] = E[S_i] = \frac{1}{L(i)} \sum_{n=1}^{L(i)} X_i(n)$, since the noise is zero-mean and independent of the parameter, and the mean is computed over a range of viewing angles $L(i)$ ($L(i)$ can be different for different features). Thus we can compute the probability of a feature X_n in one model matching another feature Y_i in another model from (6.3). The probability is maximum when both X_n and Y_i match a particular prior feature μ_j.

6.1.5 Identifying Unpaired Features

In matching features from two different views, it is important to identify features present in one view but not in the other. If a particular feature X_n does not have a corresponding match in the set $\mathbf{Y}$, then $P(H_i|X = X_n), i = 1, ..., M$ will not have any distinct peak and X_n can be identified. Similarly, $P(H_i|Y = Y_m), i = 1, ..., N$ will have a relatively flat profile if Y_m does not have a corresponding match in $\mathbf{X}$.

6.1.6 Correspondence Matrix

From the posterior probabilities, we would like to obtain a single doubly-stochastic matrix $\mathbf{C}(\mathbf{X}, \mathbf{Y})$, each row of which denotes the probability of matching the el-

[1] The notation $X_i(n)$ represents the i^{th} feature from the n^{th} viewing position.

ements of $\mathbf{Y}$ given a particular $\mathbf{X}$, and each column the probability of matching the elements of $\mathbf{X}$ given a particular $\mathbf{Y}$. This is done by using the Sinkhorn normalization procedure to obtain a doubly-stochastic matrix by alternating row and column normalizations [110]. This allows us to use either $\mathbf{X}$ or $\mathbf{Y}$ as the reference feature set.

6.2 Matching the Spatial Arrangement of Features

Rather than computing a probability of match for individual features, a more reliable correspondence can be obtained if we consider the entire set of features, taking into account their spatial arrangement in the object, i.e. the constraints on the relative configuration of the features. Consider, for the purposes of this analysis, two sets of features $\mathbf{X}$ and $\mathbf{Y}$ having the same cardinality, say N (after identifying the unpaired features). We want to assign a probability of match of $\mathbf{X}$ against all possible permutations of $\mathbf{Y}$. Let the permutations of $\mathbf{Y}$ be represented by $\mathbf{Y}^1, ..., \mathbf{Y}^{N!}$, with $\mathbf{Y}^i = [Y_{(1)}, ..., Y_{(N)}]$, where $[Y_{(1)}, ..., Y_{(N)}]$ represents an ordering of $[Y_1, ..., Y_N]$. Let H^i represent the hypothesis that $\mathbf{Y}^i$ matches $\mathbf{X}$ (note the superscript used to distinguish the hypothesis for individual features). Then

$$P(H^i|\mathbf{X}) = \Pi_{j=1}^{N} P(H_{(j)}|X_j), \tag{6.5}$$

where $H_{(j)}$ is the hypothesis that $Y_{(j)}$ matches X_j for a particular permutation $\mathbf{Y}^i$. This assumes the conditional independence of each hypothesis H_j. This is a valid assumption for the features of a face without much change in expression; however, for other examples like the whole human body in motion, such an assumption is not true, as the different parts usually move together. Computing each of the probabilities in (6.5), we see that $P(H^i|X)$ is maximum when the

permutation $\mathbf{Y}^i$ matches the set $\mathbf{X}$, element to element.

Our method works by maximizing the posterior density. Viewed from a Bayesian perspective, this is equivalent to minimizing the Bayes risk, which is the probability of error under the condition that incorrect decisions incur equal costs [111]. Thus, our algorithm is optimal in the sense that it produces a minimum probability of mismatch.

6.3 The Correspondence Algorithm

We are given two images $\mathcal{I}_1$ and $\mathcal{I}_2$, obtained from projections of the two partial models, and the pre-computed prior information $\mu_1, ..., \mu_K$.

1.*Feature Extraction:* Compute the set of features $\mathbf{X} = [X_1, ..., X_P]$ and $\mathbf{Y} = [Y_1, ..., Y_M]$ using a suitable feature extraction method (in our case, a corner-finder algorithm).

2.*Compute Probability of Match:* Compute the match probabilities from (6.3) using the prior information $\mu_1, ..., \mu_K$.

3.*Identify Unpaired Features:* Identify those features present in one view, but not in the other as explained above. At the end of this process, we are left with two sets with the same cardinality (denoting the paired features) which have to be matched. Denote them by $\mathbf{X} = [X_1, ..., X_N]$ and $\mathbf{Y} = [Y_1, ..., Y_N]$.

4.*Sinkhorn Normalization:* Compute the correspondence matrix $\mathbf{C}(\mathbf{X}, \mathbf{Y})$ by applying the Sinkhorn normalization procedure to the match probabilities after removing the unpaired features.

5.*Compute Probability for the Spatial Arrangement of the Features:* Compute the posterior probability for matching $\mathbf{X}$ with all permutations of $\mathbf{Y}$, i.e. $P(H^i|\mathbf{X}), i = 1, ..., N!$ from (6.5).

6.*Search for Best Match:* Obtain $i = \arg\max_i P(H^i|\mathbf{X})$. Assign $\mathbf{Y}^i = [Y_{(1)}, ..., Y_{(N)}]$ as the match to $\mathbf{X}$.

6.3.1 Reducing the Search Space

The search space in the last step of the above algorithm is of size $N!$. In practice, the search space can be reduced. For each $X = X_n, n = 1, ..., N$ for the paired sets of features, identify the set $\bar{Y}_n = \{Y_i : P(H_i|X = X_n) > p\}$, where p is an appropriately chosen threshold. Alternatively, we can choose the $\{Y_i\}$ that have the largest l values of the posterior densities. This smaller set identifies those features in $\mathbf{Y}$ which are the closest to a particular feature in $\mathbf{X}$. We can then compute the probability of match for the permutations of $\mathbf{Y}$ in this reduced set. The actual number of elements contained in the search space will depend on the exact values of the probabilities of $\bar{Y}_n, n = 1, ..., N$.

6.4 Experimental Analysis and Applications

We present the results of our algorithm applied to the problem of aligning partial 3D models obtained from two different video sequences. Two images obtained from each of the partial models were used to identify the features and obtain the correspondences. The prior information was pre-computed from a video sequence which was different from the one used to obtain the 3D models. The same prior can be used to obtain different holistic 3D face models of different people. We present here detailed results on one such model.

6.4.1 Feature Selection and Prior Extraction

To select the features that need to be registered, we use a corner finder algorithm based on an interest operator [2] [82]. Figure 6.1 shows the outputs of the corner finder algorithm, represented by small dots. Given this set of points defining the corners in the image, a clustering algorithm, like k-means, was used to identify feature points that need to be matched. Fig. 6.2 plots two sets of features identified using this strategy. However, in order to avoid the feature matching problems that can arise due to the symmetry of a face, we only considered features located in the right 70% of the original images. In addition, features lying in the region near the image boundaries were neglected. We will present our results on this smaller set of features. The number of features is sufficient for our application, as the number of distinct features on a face is limited and we are considering not just the specific points represented by these features but the regions around them. Figures 6.3 and 6.4 plot the intensities in the local regions around the features and Figures 6.5 and 6.6 plot the output of the corner-finder algorithm around these features. Fig. 6.7 represents the pre-computed prior information in the form of the mean features. The prior was collected by tracking a set of features across multiple frames of a video sequence and then integrating them.

6.4.2 Estimation of Posterior Probabilities

Figure 6.8 gives a graphical representation of the posterior probability matrix $\mathbf{P}(\mathbf{X}, \mathbf{Y})$ obtained before the Sinkhorn normalization procedure. It can be seen

[2] The interest operator computes the matrix of second moments of the local gradient and determines corners in the image based on the eigenvalues of this matrix.

Figure 6.1: The output of the corner finder algorithm on two images obtained from projections of the partial models, represented by small dots.

Figure 6.2: Features identified in the front and side view images by applying a k-means clustering to the output of the corner-finder.

that there is a distinct peak for each row and column of the matrix, corresponding to matching of a pair of features. The valleys of this surface plot, representing rows or columns with no peaks, correspond to unmatched pairs of features. Figures 6.9 and 6.10 plot the rows and columns of $\mathbf{P}(\mathbf{X}, \mathbf{Y})$ respectively. The true values (as obtained manually) are marked by a * on the horizontal axis, except for those which are unmatched (the unpaired features).

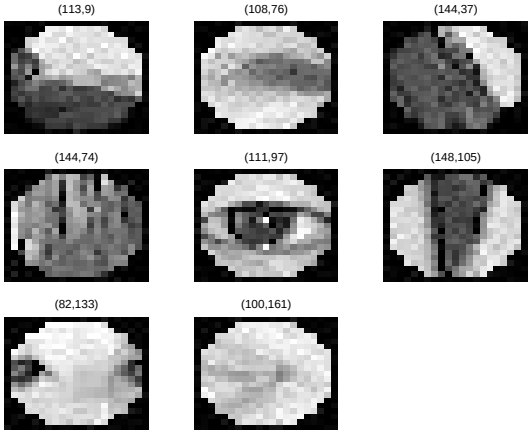


Figure 6.3: Intensity blocks around the features to be matched in the front view. The numbers represent the positions of the corresponding features in the image.

6.4.3 Matching the Spatial Arrangement of Features

Figure 6.11 plots the probabilities for matching $\mathbf{X}$ against all possible permutations of $\mathbf{Y}$. Comparison with Figures 6.9 and 6.10 shows that there is a very distinct peak in this case, justifying our earlier assertion that taking into account the spatial arrangement of the features leads to a more robust algorithm.

6.4.4 Importance of Prior Information

In Figure 6.12, we plot the probabilities of matching each feature in $\mathbf{X}$ to the different features in $\mathbf{Y}$, where we do not have the pre-computed prior information. The probabilities were estimated using the shape similarity of the two features. This was done using the standard technique of computing the ratios of the eigenvalues of the first and second central moments of the coordinates of the set of points representing the features [82]. This was extended to consider the permutations of the features so as to take advantage of the global arrangement.

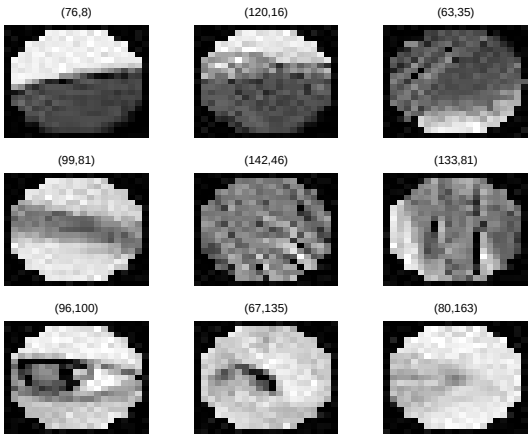


Figure 6.4: Intensity blocks around the features to be matched in the side view. The numbers represent the positions of the corresponding features in the image.

Figure 6.13 plots the probability of matching the spatial arrangement of the features without taking advantage of the prior information. In both cases, we see that the peaks of the probabilities do not correspond to the true match, as indicated in the plots. This emphasizes the importance of the prior information and shows how a simple correlation-based matching technique can be modified to provide a very robust solution by incorporating suitable information gathered from the video data.

6.4.5 Importance of Considering the Relative Configuration of Features

Even though the search over all possible permutations of one feature set leads to increased computational load, the advantage in terms of robustness of the final solution makes it worthwhile. Comparison of Figures 6.8 and 6.11 shows that the peak of the probability curve is much more pronounced when the configuration of

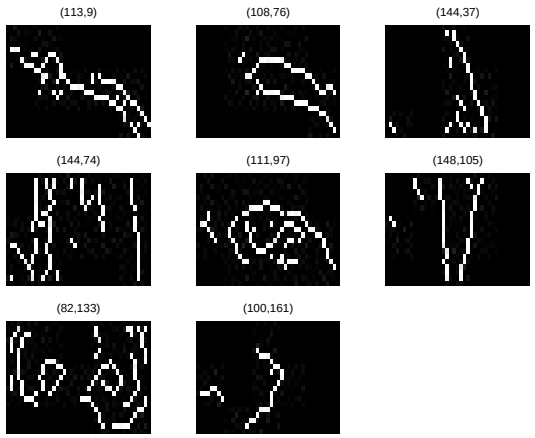


Figure 6.5: The shapes of the significant image attributes in the front view around the feature point whose position in the original image is indicated on top.

the feature set is taken into account than when we try to obtain the registration by considering the probabilities of match of the individual features.

6.4.6 Application to 3D Model Alignment

We now demonstrate the application of our correspondence algorithm to aligning two models of a human face obtained from different views. Each of the models was obtained from a video sequence of a person moving his head in front of a static camera. The video sequence was split into two portions, corresponding to the front and side views of the face. The two partial models were obtained from these two portions of the video sequence. In order to obtain the 3D models from video, a set of features were tracked and the depth and camera motion at these points were computed using a multi-frame structure from motion (SfM) algorithm. The SfM algorithm worked by fusing the depth estimates obtained from two images using optical flow techniques. The fusion was done using robust

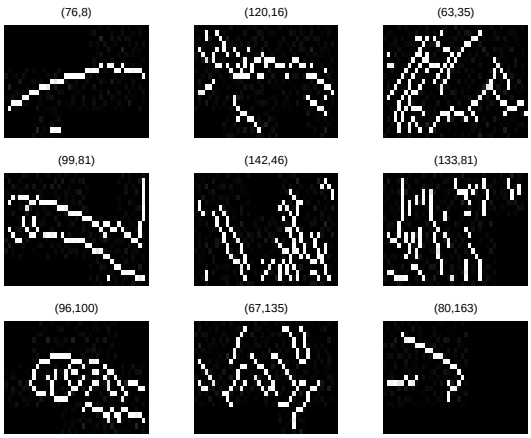


Figure 6.6: The shapes of the significant image attributes in the side view around the feature point whose position in the original image is indicated on top.

statistics and a generic model of a face. The error in the reconstruction was estimated and compensated for. Details of the 3D modeling algorithm were presented in Chapter 4. Fig. 6.14 shows the two models, one from the front, the other from the side, which we aim to integrate into one holistic model.

In order to align these two partial models, one image, obtained from each of the views, was considered, and our algorithm was used to obtain a correspondence between the features automatically selected in these images. Prior information about important features in a human face was pre-computed and used for this application. The prior information was automatically obtained by tracking features across a video sequence and obtaining an average representation for each feature over the entire video sequence. Our algorithm in Section 6.3 was then used to obtain the correspondences between the different features. Having obtained the feature correspondence, we compute the affine transformation between the two models for all of the features, i.e. $\mathbf{y}_i = \mathbf{R}\mathbf{x}_i + \mathbf{T}$ where $\mathbf{x}_i$

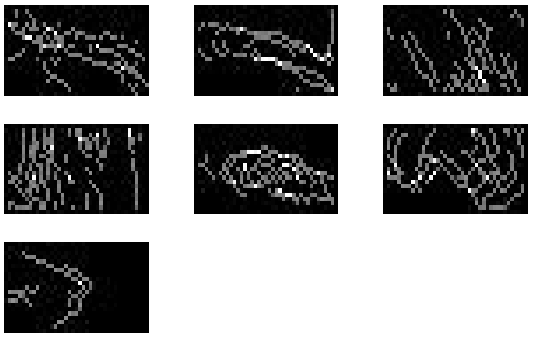


Figure 6.7: The prior information (the shape representation averaged over a large number of viewing angles) which was pre-computed.

and $\mathbf{y}_i$ are the 3D coordinates of a matching pair of points and $\mathbf{R}$ and $\mathbf{T}$ the rotation and translation between the two models. Any other method of obtaining the transformation would work just as well, or even better. Fig. 6.14 also shows two views of the complete model after alignment.

6.5 Conclusions

In this chapter, we have presented an algorithm for creating holistic 3D models by matching two sets of features extracted automatically from the models, taking into consideration the relative configuration of the feature sets. The Sinkhorn normalization procedure is used to obtain a doubly-stochastic matrix denoting the probabilities of match for the two feature sets. The method works by minimizing the probability of a mismatch (using the Bayes error criterion) between the shapes of the features, after taking into account their spatial arrangement. Robustness is achieved by including prior information about the feature sets. We emphasize that the prior can be obtained from the video data collection,

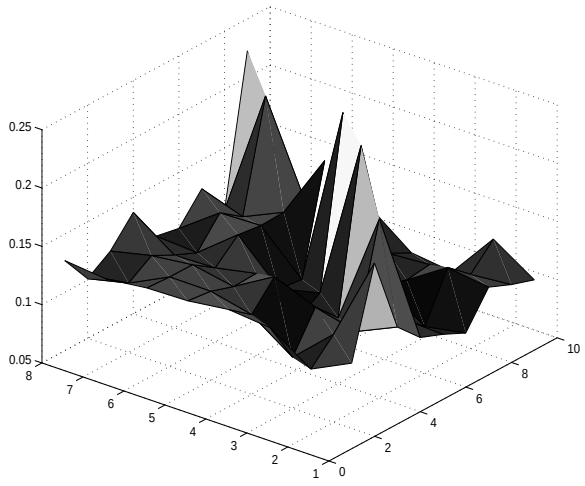


Figure 6.8: The posterior density matrix.

and thus needs to be done only once for each class of examples. The incorporation of the prior information and the spatial structure of the object leads to an extremely robust algorithm.

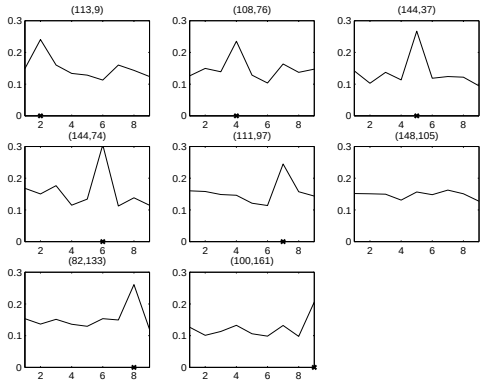


Figure 6.9: The *a posteriori* probabilities for each of the features in the front image, obtained from each of the rows of the correspondence matrix.

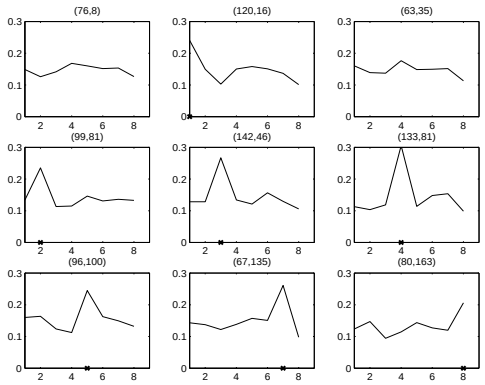


Figure 6.10: The *a posteriori* probabilities for each of the features in the side image, obtained from each of the columns of the correspondence matrix.

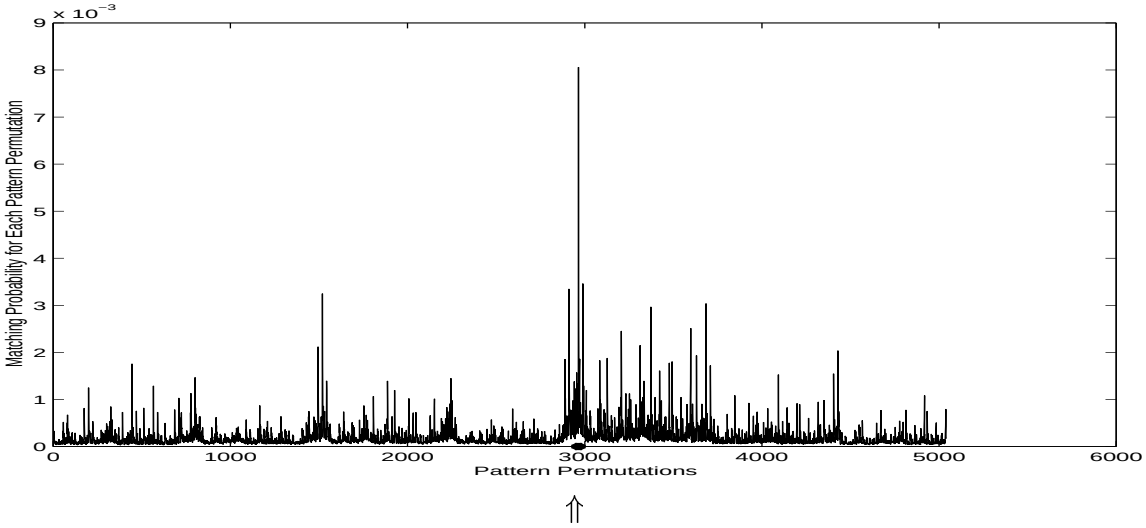


Figure 6.11: The probability of matching $\mathbf{X}$ against all permutations of $\mathbf{Y}$. The true value is marked with a $\Uparrow$ below the horizontal axis.

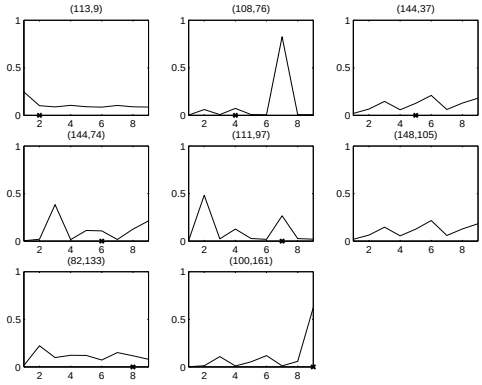


Figure 6.12: The probability of match for each of the features in the front image, for the case where prior information is not available.

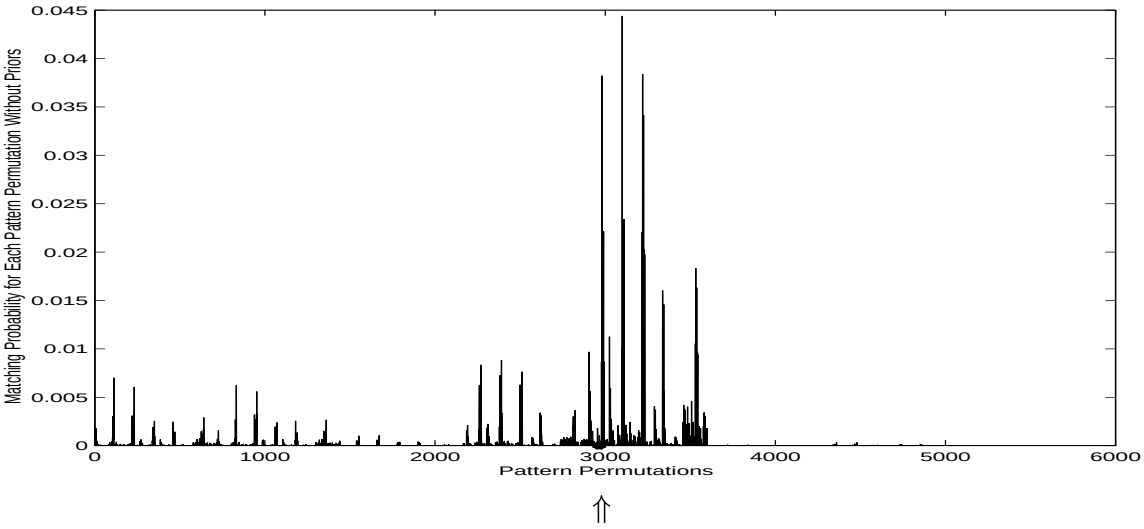


Figure 6.13: The probability of match for the shape of each feature in the front image against all possible combinations of the features in the side view, for the case where prior information is not available. The true value is marked with a ⇑ below the horizontal axis.

Figure 6.14: 3D models from the front and side which are used as input to the algorithm, and two views of the 3D model obtained after the alignment.

Chapter 7

Conclusions and Future Work

In this thesis, we have addressed the problem of reconstructing a 3D scene from a video sequence using optical flow to estimate the motion between pairs of frames. Since optical flow computations assume small inter-frame motion, the errors in the motion estimates are often of comparable magnitude to the actual displacements. This is the typical low signal to noise (SNR) ratio problem of signal processing and any method which relies on using flow needs to take into account this aspect of the problem. In our work, we have tried to solve the problem of estimating 3D structure and camera motion after compensating for the effects of errors in the motion estimates. We have applied our knowledge of the statistics of the errors to obtain robust 3D face reconstructions. The following were the main contributions of the thesis.

Error Covariance of 3D Reconstruction: In Chapter 2, we derived a precise expression relating the error covariances of the camera motion and structure estimates as a function of the error covariances in the measurements of the feature positions. We showed that we could derive a second order statistic to measure the error in a multi-frame reconstruction as a function of the number of frames.

Bias in 3D Reconstruction: We showed in Chapter 3 that the depth estimate is statistically biased and the magnitude of the bias is significant. We analyzed how the camera motion affects the bias and found that different motion trajectories have drastically different effects on the bias. We also noted that our analytical results are in accordance with the phenomenon of bias observed in psychophysics experiments on human observers.

3D Estimation Algorithm: The main application of our work was in building accurate 3D models of human faces from a video sequence. We showed that a combination of robust statistics and our error analysis could be used to achieve this purpose. A generic face model was used to correct for the residual errors from such a scheme in an MCMC optimization framework.

Information-Theoretic Quality Evaluation: To evaluate the quality of a 3D reconstruction algorithm from multiple images, we proposed a criterion termed incremental mutual information (IMI). IMI allows us to estimate the number of frames necessary to obtain a reconstruction of the desired fidelity. We showed how the IMI can be estimated using Monte Carlo techniques.

Combining Partial 3D Models: Since holistic 3D models are usually obtained by stitching together partial ones, we proposed a probabilistic algorithm for matching the shapes of significant features in two partial models and applied it to the face reconstruction problem.

7.0.1 Future Work

Being able to build accurate 3D models is the holy grail of computer vision since it can lead to the solution of numerous other problems like recognition, tracking, automatic navigation, etc. Our work thus opens up possibilities in numerous other areas. We will outline two directions which can lead to future work, one dealing with the understanding of the fundamentals of the 3D reconstruction problem, and the other with its applications.

Error Analysis for Surfaces: Our study of the accuracy of 3D reconstruction from video has concentrated on the errors in point features sampled from the optical flow. An important extension would be to obtain similar results for surfaces rather than individual points. The optical flow framework is particularly suited to this as it gives a dense flow field. Also, partial differential equation methods used to derive the flow are suitable for analyzing continuous surface patches. It would be interesting and useful to obtain expressions for the error in the reconstruction along the lines of the bias, covariance, and mutual information for surfaces.

Multi-resolution surface reconstruction: Our algorithm concentrates on reconstructing the 3D surface at one resolution. However, analysis of most real objects reveal that there are portions which are rich in detail and others which are not. This fact has been taken advantage of in other areas of image processing like image compression. There, those areas which are rich in texture are usually represented with more bits than those which are not. A similar idea can be applied to depth values. Those portions of a surface which have detailed depth variations can be reconstructed at

higher levels of resolution than parts which are smooth. The use of different multi-resolution techniques like the Laplacian pyramid scheme [112] or wavelets [113] can be investigated for this purpose.

Recognition and Retrieval Applications: Most face recognition algorithms perform very well when the training and test images are from similar viewpoints. However, if the training image is a frontal face view and the test image is a profile view of the same person, existing algorithms usually do not perform satisfactorily. The ability to create accurate 3D models from a few images obtained from a particular view can be used to solve this problem. In our work, we have shown how to build 3D models of a face from a few frames of a video sequence which concentrate on the front view. However, once the 3D model has been obtained, its projections at different viewing angles can be used to synthesize novel views (e.g. profile views) and match them against the test images. Hence 3D models can be used for recognition, human ID and surveillance applications. Similar ideas can be used for identifying objects from a video sequence in image retrieval and content analysis problems.

Multimedia Communications: Video communications, like video conferencing, usually use MPEG-2 standards where individual frames of the sequence are coded either individually (intra-coding) using image transforms (usually discrete cosine transforms) or by predicting from previous and future frames (predictive coding). A completely different approach, which is an area of active research today, is to use 3D models in communications. Once the 3D model has been built and transmitted to the decoder side, the encoder needs to send the camera positions and the differences

between the images and the projections of the 3D model at these positions. The decoder can then reconstruct the original video sequence, up to limits imposed by coding and transmission errors. The use of 3D models for communications is one of the most important ideas in MPEG-4 standards, and present research in this area is investigating the advantages of such methods over MPEG-2 transmission in terms of bit-rate efficiency and reconstruction accuracy.

Flexible Appearance Modeling and 3D Tracking: Our work on building 3D models of faces assumes that the object is rigid. A very interesting application of the 3D models would be to make them flexible so as to model different human expressions and emotions. The models can then be used for face tracking. The motions of certain control points on the face can be learnt for different expressions and then combined with the 3D face model to obtain flexible appearance models.

Appendix A

Stochastic Approximation

The concept of stochastic approximation has been developed in statistics for certain sequential parameter estimation problems [12, 53, 58, 114, 115]. It "may be considered as a recursive estimation method, updated by an appropriately weighted, arbitrarily chosen error corrective term, with the only requirement that in the limit it converges to the true parameter value sought" [116]. The multiplicity of the sources of error that combine in a complicated manner and the danger of assuming a statistical model that is incorrect and will produce erroneous reconstructions inspired us to use SA in SfM, as we do not need to know the distribution function of the error. Besides, it provides a recursive algorithm and guarantees local optimality of the estimate, which can be non-linear. On the other hand, in non-Gaussian cases, the Kalman filter is optimal only among the class of linear filters (in the mean square sense). For the Gaussian distribution, it is an optimal filter in the mean square sense. Since LMedS is a non-linear estimator and the distribution of the depth sub-estimates is unknown, SA is used to obtain an optimal solution based on the method of calculating the quantile of any distribution recursively, proposed originally by Robbins and Monro in their seminal paper [70].

Let $\{e(k)\}$ be a sequence of random variables with the same distribution indexed by a discrete time variable k. A function $Q(x, e(k))$ is given such that

$$E[Q(x, e(k))] = g(x) = 0 \tag{A.1}$$

where E denotes expectation over e. The distribution of $e(k)$ is not known; the exact form of the function $Q(x, e)$ may also be unknown, though its values are observed and it can be constructed for any chosen x. The problem is to determine the solution of $g(x) = 0$. Robbins and Monro (RM) suggested the following scheme for solving (A.1) recursively as time evolves [70]:

$$\hat{x}(k) = \hat{x}(k-1) + a_k Q(\hat{x}(k-1), e(k)) \tag{A.2}$$

where the gain sequence $\{a_k\}$ must satisfy the following conditions [114, 12, 58]:

$$a_k \geq 0, \quad a_k \to 0, \quad \sum_{k=1}^{\infty} a_k = \infty, \quad \sum_{k=1}^{\infty} a_k^2 < \infty. \tag{A.3}$$

A popular choice of the gain sequence, which was used in our experiments also, is $a_k = a/(k+1)^{0.501}$.

The stochastic gradient is only one of the ways of applying stochastic approximation ideas. It can be viewed as the stochastic analog of the method of steepest descent [117]. However, as is well known in the optimization literature, steepest descent is fairly inefficient, especially when the iterates get close to the minimum. Since Newton's method gives an improvement over the steepest descent method for deterministic problems, it is reasonable to assume that it will perform better in the stochastic approximation case also. Suppose that for each x, we can construct an approximation of the Hessian, denoted by $\overline{L}''(x, e^k)$, where $e^k = [e(k), e(k-1), ..., e(1)]$. Then the natural stochastic equivalent of Newton's method is

$$\hat{x}(k) = \hat{x}(k-1) - a_k[\overline{L}''(x, e^k)]^{-1} Q(\hat{x}(k-1), e_k). \tag{A.4}$$

We shall call this the "stochastic Newton's method".

It can be shown that the estimate obtained from SA is unbiased, consistent and asymptotically normal , and in many cases, also efficient [12, 35].

Appendix B

Computation of Mutual Information

We will show how to compute the determinant of the matrix $P_{V^{(N)}} + \mathbf{1}_N P_X \mathbf{1}_N^T$ in (5.15), which is required for computing the incremental mutual information in Chapter 5.

Let us denote by $\mathbf{A}_N$ the following matrix:

$$\mathbf{A}_N = \begin{bmatrix} \sigma_x^2 + \sigma_{v_1}^2 & \sigma_x^2 & \cdots & \sigma_x^2 \\ \sigma_x^2 & \sigma_x^2 + \sigma_{v_2}^2 & \cdots & \sigma_x^2 \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_x^2 & \cdots & \sigma_x^2 & \sigma_x^2 + \sigma_{v_N}^2 \end{bmatrix}. \tag{B.1}$$

The aim is to compute the determinant of $\mathbf{A}_N$. We will do so by using the method of induction. Consider $N = 2$. Then the determinant of $\mathbf{A}_2$ is denoted by $|\mathbf{A}_2| = \sigma_x^2(\sigma_{v_1}^2 + \sigma_{v_2}^2) + \sigma_{v_1}^2 \sigma_{v_2}^2$. For $N = 3$, $|\mathbf{A}_3| = \sigma_x^2(\sigma_{v_1}^2\sigma_{v_2}^2 + \sigma_{v_2}^2\sigma_{v_3}^2 + \sigma_{v_3}^2\sigma_{v_1}^2) + \sigma_{v_1}^2\sigma_{v_2}^2\sigma_{v_3}^2$. Assume that

$$|\mathbf{A}_N| = \prod_{i=1}^{N} \sigma_{v_i}^2 + \sum_{i=1}^{N} \sigma_x^2 \prod_{\substack{j=1 \\ j \neq i}}^{N} \sigma_{v_j}^2. \tag{B.2}$$

Now consider the matrix

$$\mathbf{A}_{N+1} = \begin{bmatrix} \sigma_x^2 + \sigma_{v_1}^2 & \sigma_x^2 & \cdots & \sigma_x^2 & \sigma_x^2 \\ \sigma_x^2 & \sigma_x^2 + \sigma_{v_2}^2 & \cdots & \sigma_x^2 & \sigma_x^2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \sigma_x^2 & \cdots & \sigma_x^2 & \sigma_x^2 + \sigma_{v_N}^2 & \sigma_x^2 \\ \sigma_x^2 & \sigma_x^2 & \cdots & \sigma_x^2 & \sigma_x^2 + \sigma_{v_{N+1}}^2 \end{bmatrix}. \tag{B.3}$$

Subtracting the second to last row from the last one, we get the following matrix:

$$\begin{bmatrix} \sigma_x^2 + \sigma_{v_1}^2 & \sigma_x^2 & \cdots & \sigma_x^2 & \sigma_x^2 \\ \sigma_x^2 & \sigma_x^2 + \sigma_{v_2}^2 & \cdots & \sigma_x^2 & \sigma_x^2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \sigma_x^2 & \cdots & \sigma_x^2 & \sigma_x^2 + \sigma_{v_N}^2 & \sigma_x^2 \\ 0 & 0 & \cdots & -\sigma_{v_N}^2 & \sigma_{v_{N+1}}^2 \end{bmatrix}. \tag{B.4}$$

Then

$$|\mathbf{A}_{N+1}| = \sigma_{v_{N+1}}^2 |\mathbf{A}_N| + \sigma_{v_N}^2 |\mathbf{B}|, \tag{B.5}$$

where

$$\mathbf{B} = \begin{bmatrix} \sigma_x^2 + \sigma_{v_1}^2 & \sigma_x^2 & \cdots & \sigma_x^2 & \sigma_x^2 \\ \sigma_x^2 & \sigma_x^2 + \sigma_{v_2}^2 & \cdots & \sigma_x^2 & \sigma_x^2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \sigma_x^2 & \sigma_x^2 & \cdots & \sigma_x^2 + \sigma_{v_{N-1}}^2 & \sigma_x^2 \\ \sigma_x^2 & \cdots & \sigma_x^2 & \sigma_x^2 & \sigma_x^2 \end{bmatrix}. \tag{B.6}$$

Now using the fact that the determinant of a matrix remains unchanged by elementary row and column operations, we get

$$|\mathbf{B}| = \begin{vmatrix} \sigma_x^2 + \sigma_{v_1}^2 & -\sigma_{v_1}^2 & \sigma_{v_1}^2 & \cdots & -\sigma_{v_1}^2 & -\sigma_{v_1}^2 \\ \sigma_x^2 & \sigma_{v_2}^2 & 0 & \cdots & 0 & 0 \\ \vdots & 0 & \sigma_{v_3}^2 & \ddots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \sigma_x^2 & 0 & 0 & \cdots & \sigma_{v_{N-1}}^2 & 0 \\ \sigma_x^2 & 0 & 0 & \cdots & 0 & 0 \end{vmatrix},$$

$$
\begin{aligned}
&= (-1)^{N+1}\sigma_x^2 \begin{vmatrix} -\sigma_{v_1}^2 & -\sigma_{v_1}^2 & \cdots & -\sigma_{v_1}^2 & -\sigma_{v_1}^2 \\ \sigma_{v_2}^2 & 0 & \cdots & 0 & 0 \\ 0 & \sigma_{v_3}^2 & \cdots & \vdots & \vdots \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & \sigma_{v_{N-1}}^2 & 0 \end{vmatrix}, \\
&= \sigma_x^2\sigma_{v_1}^2\sigma_{v_2}^2\cdots\sigma_{v_{N-1}}^2. \qquad \text{(B.7)}
\end{aligned}
$$

Then, substituting (B.2) and (B.7) into (B.5), we get

$$
\begin{aligned}
\mathbf{A}_{N+1} &= \prod_{i=1}^{N+1}\sigma_{v_i}^2 + \sigma_{v_{N+1}}^2\left(\sum_{i=1}^{N}\sigma_x^2\prod_{\substack{j=1\\ j\neq i}}^{N}\sigma_{v_j}^2\right) + \sigma_x^2\prod_{i=1}^{N}\sigma_{v_i}^2, \\
&= \prod_{i=1}^{N+1}\sigma_{v_i}^2 + \sum_{i=1}^{N}\sigma_x^2\prod_{\substack{j=1\\ j\neq i}}^{N+1}\sigma_{v_j}^2 + \sigma_x^2\prod_{i=1}^{N}\sigma_{v_i}^2, \\
&= \prod_{i=1}^{N+1}\sigma_{v_i}^2 + \sum_{i=1}^{N+1}\sigma_x^2\prod_{\substack{j=1\\ j\neq i}}^{N+1}\sigma_{v_j}^2, \qquad \text{(B.8)}
\end{aligned}
$$

which proves the hypothesis about $|\mathbf{A}_N|$ in (B.2).

BIBLIOGRAPHY

[1] Z. Zhang and O. Faugeras, *3D Dynamic Scene Analysis*, Springer-Verlag, 1992.

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

[3] R. I. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, Cambridge University Press, 2000.

[4] C. Jerian and R. Jain, "Structure from motion: A critical analysis of methods," *IEEE Trans. on Systems, Man and Cybernetics* **21**, pp. 572–588, 1991.

[5] T. Huang and A. Netravali, "Motion and structure from feature correspondences: A review," *Proceedings of the IEEE* **82**, pp. 252–268, 1994.

[6] J. Oliensis, "A critique of structure from motion algorithms," Tech. Rep. http://www.neci.nj.nec.com/homepages/oliensis/, NECI, 2000.

[7] B. Horn and B. Schunck, "Determining optical flow," *AI* **17**, pp. 185–203, 1981.

[8] S. Srinivasan, "Extracting structure from optical flow using fast error search technique," *International Journal of Computer Vision* **37**, pp. 203–230, 2000.

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

[10] R. Tsai and T. Huang, "Estimating 3-D motion parameters of a rigid planar patch: I," *IEEE Trans. on Acoustics, Speech and Signal Processing* **29**, pp. 1147–1152, 1981.

[11] D. Gennery, "Tracking known three-dimensional objects," in *AAAI*, pp. 13–17, 1982.

[12] L. Ljung and T. Soderstrom, *Theory and Practice of Recursive Identification*, MIT Press, 1987.

[13] T. Broida and R. Chellappa, "Estimating the kinematics and structure of a rigid object from a sequence of monocular images," *IEEE Trans. on Pattern Analysis and Machine Intelligence* **13**, pp. 497–513, 1991.

[14] A. Azarbayejani and A. Pentland, "Recursive estimation of motion, structure, and focal length," *IEEE Trans. on Pattern Analysis and Machine Intelligence* **17**, pp. 562–575, 1995.

[15] C. Tomasi and T. Kanade, "Shape and motion from image streams under orthography: A factorization method," *International Journal of Computer Vision* **9**, pp. 137–154, 1992.

[16] R. Szeliski and S. Kang, "Recovering 3D shape and motion from image streams using non-linear least squares," *Journal of Visual Computation and Image Representation* **5**, pp. 10–28, 1994.

[17] J. Oliensis, "A multi-frame structure-from-motion algorithm under perspective projection," *International Journal of Computer Vision* **34**, pp. 1–30, 1999.

[18] J. Oliensis and Y. Genc, "Fast and accurate algorithms for projective multi-image structure from motion," *IEEE Trans. on Pattern Analysis and Machine Intelligence* **23**, pp. 546–559, 2001.

[19] J. Inigo Thomas and J. Oliensis, "Dealing with noise in multiframe structure from motion," *Computer Vision and Image Understanding* **76**, pp. 109–124, 1999.

[20] J. Weng, N. Ahuja, and T. Huang, "Optimal motion and structure estimation," *IEEE Trans. on Pattern Analysis and Machine Intelligence* **15**, pp. 864–884, 1993.

[21] J. Weng, T. Huang, and N. Ahuja, "3-D motion estimation, understanding, and prediction from noisy image sequences," *IEEE Trans. on Pattern Analysis and Machine Intelligence* **9**, pp. 370–389, 1987.

[22] T. Broida and R. Chellappa, "Performance bounds for estimating three-dimensional motion parameters from a sequence of noisy images," *Journal of the Optical Society of America A* **6**, pp. 879–889, 1989.

[23] G. Young and R. Chellappa, "Statistical analysis of inherent ambiguities in recovering 3-D motion from a noisy flow field," *IEEE Trans. on Pattern Analysis and Machine Intelligence* **14**, pp. 995–1013, 1992.

[24] G. Young and R. Chellappa, "3-D motion estimation using a sequence of noisy stereo images: Models, estimation, and uniqueness results," *IEEE*

Trans. on Pattern Analysis and Machine Intelligence **12**, pp. 735–759, 1990.

[25] K. Daniilidis and H. Nagel, "The coupling of rotation and translation in motion estimation of planar surfaces," in *Conference on Computer Vision and Pattern Recognition*, pp. 188–193, 1993.

[26] K. Daniilidis and H. Nagel, "Analytic results on error sensitivity of motion estimation from two views," *Image and Vision Computing* **8**, pp. 297–303, 1990.

[27] Z. Zhang, "Determining the epipolar geometry and its uncertainty: A review," *International Journal of Computer Vision* **27**, pp. 161–195, 1998.

[28] R. Haralick, "Covariance propagation in computer vision," in *ECCV Workshop on Performance Characteristics of Vision Algorithms*, 1996.

[29] S. Soatto and R. Brockett, "Optimal structure from motion: Local ambiguities and global estimates," in *Conference on Computer Vision and Pattern Recognition*, pp. 282–288, 1998.

[30] Y. Ma, J. Kosecka, and S. Sastry, "Linear differential algorithm for motion recovery: A geometric approach," *International Journal of Computer Vision* **36**, pp. 71–89, 2000.

[31] Z. Sun, V. Ramesh, and A. Tekalp, "Error characterization of the factorization method," *Computer Vision and Image Understanding* **82**, pp. 110–137, 2001.

[32] D. Morris, K. Kanatani, and T. Kanade, "3D model accuracy and gauge fixing," tech. rep., Carnegie-Mellon University, 2000.

[33] R. Walter, *Principles of Mathematical Analysis, 3rd Edition*, McGraw-Hill, 1976.

[34] V. Nalwa, *A Guided Tour of Computer Vision*, Addison-Wesley, 1993.

[35] H. Poor, *An Introduction to Signal Detection and Estimation*, Springer-Verlag, 1988.

[36] A. Papoulis, *Probability, Random Variables and Stochastic Processes*, McGraw-Hill, 1991.

[37] J. Shao, *Mathematical Statistics*, Springer-Verlag, 1998.

[38] P. Rousseeuw and A. Leroy, *Robust Regression and Outlier Detection*, Wiley, 1987.

[39] P. Rousseeuw, "Least median of square regression," *Journal of the American Statistical Association* **79**, pp. 871–880, 1984.

[40] P. Meer, D. Mintz, and A. Rosenfeld, "Analysis of the least median of squares estimator for computer vision applications," in *Conference on Computer Vision and Pattern Recognition*, pp. 621–623, 1992.

[41] M. Black and A. Rangarajan, "On the unification of line processes, outlier rejection, and robust statistics with applications in early vision," *International Journal of Computer Vision* **19**, pp. 57–91, 1996.

[42] G. Golub and C. Van Loan, *Matrix Computations*, Johns Hopkins University Press, 1989.

[43] J. Todd, "Theoretical and biological limitations on the visual perception of three-dimensional structure from motion," in *High Level Motion Processing: Computational, Neurobiological and Psychophysical Perspectives*, 1998.

[44] K. Daniilidis and M. Spetsakis, "Understanding noise sensitivity in structure from motion," in *VisNav*, 1993.

[45] K. Kanatani, "Unbiased estimation and statistical analysis of 3-D rigid motion from two views," *IEEE Trans. on Pattern Analysis and Machine Intelligence* **15**, pp. 37–50, 1993.

[46] J. Besag, "Errors-in-variable estimation for Gaussian lattice schemes," *Journal of Royal Statistical Society* **B 39**, pp. 73–78, 1977.

[47] W. Fuller, *Measurement Error Models*, Wiley, 1987.

[48] T. Kailath, *Linear Systems*, Prentice-Hall, 1980.

[49] S. Van Huffel and J. Vandewalle, *The Total Least Squares Problem*, SIAM Frontiers in Applied Mathematics, 1991.

[50] R. Carpenter, *Movements of the Eye*, Pion, London, 1988.

[51] T. Broida and R. Chellappa, "Estimation of object motion parameters from noisy images," *IEEE Trans. on Pattern Analysis and Machine Intelligence* **8**, pp. 90–99, 1986.

[52] K. Kanatani, *Statistical Optimization for Geometric Computation: Theory and Practice*, North-Holland, 1996.

[53] V. Solo and X. Kong, *Adaptive Signal Processing Algorithms*, Prentice-Hall, 1995.

[54] A. Roy Chowdhury and R. Chellappa, "Stochastic approximation and rate-distortion analysis for robust structure and motion estimation," Tech. Rep. CAR-TR-971, University of Maryland, 2001.

[55] B. Triggs, P. McLauchlan, R. Hartley, and A. Fitzgibbon, "Bundle adjustment - A modern synthesis," in *Vision Algorithms: Theory and Practice*, B. Triggs, A. Zisserman, and R. Szeliski, eds., pp. 298–373, Springer-Verlag, 1999.

[56] P. Fua, "Regularized bundle-adjustment to model heads from image sequences without calibration data," *International Journal of Computer Vision* **38**(2), pp. 153–171, 2000.

[57] Y. Shan, Z. Liu, and Z. Zhang, "Model-based bundle adjustment with application to face modeling," in *International Conference on Computer Vision*, pp. 644–651, 2001.

[58] J. Spall, *Introduction to Stochastic Search and Optimization*, Wiley, 2000.

[59] J. Clark and A. Yuille, *Data Fusion for Sensory Information Processing Systems*, Kluwer, 1990.

[60] S. Kirpatrick, C. Gelatt, Jr., and M. Vecchi, "Optimization by simulated annealing," *Science* **220**, pp. 671–680, 1983.

[61] J. Besag, "Markov chain Monte Carlo for statistical inference," Tech. Rep. 9, University of Washington, 2000.

[62] J. S. Liu, "Markov chain Monte Carlo and related topics," tech. rep., Stanford University, 1999.

[63] A. Doucet, "On sequential simulation based methods for bayesian filtering," *Statistics and Computing* **10**, pp. 197–208, 1998.

[64] R. Neal, "Probabilistic inference using Markov chain Monte Carlo methods," Tech. Rep. CRG-TR-93-1, University of Toronto, 1993.

[65] T. Poggio, V. Torre, and C. Koch, "Computational vision and regularization theory," *Nature* **317**, pp. 314–319, 1985.

[66] A. Rosenfeld, R. Hummel, and S. Zucker, "Scene labeling by relaxation operations," *IEEE Trans. on Systems, Man and Cybernetics* **6**, pp. 420–433, 1976.

[67] D. Terzopoulos, "The role of constraints and discontinuities in visible surface reconstruction," in *IJCAI*, pp. 1019–1022, 1983.

[68] S. Geman and D. Geman, "Stochastic relaxation, gibbs distributions, and the bayesian restoration of images," *IEEE Trans. on Pattern Analysis and Machine Intelligence* **6**, pp. 721–741, 1984.

[69] R. Chellappa and A. Jain, *Markov Random Fields: Theory and Applications*, Academic Press, 1993.

[70] H. Robbins and S. Monro, "A stochastic approximation method," *Annals of Mathematical Statistics* **22**, pp. 400–407, 1951.

[71] J. Liu and R. Chen, "Sequential Monte Carlo methods for dynamic systems," *Journal of the American Statistical Association* **93**, 1998.

[72] N. Metropolis, A. Rosenbluth, M. Rosenbluth, A. Teller, and E. Teller, "Equations of state calculations by fast computing machines," *Journal of Chemical Physics* **21**, pp. 1087–1091, 1953.

[73] W. Hastings, "Monte Carlo sampling methods using Markov chains and their applications," *Biometrika* **57**, pp. 97–109, 1970.

[74] C. Tomasi and J. Shi, "Good features to track," in *Conference on Computer Vision and Pattern Recognition*, pp. 593–600, 1994.

[75] B. Efron and R. Tibshirani, *An Introduction to the Bootstrap*, Chapman and Hall, 1993.

[76] P. Viola and W. Wells, III, "Alignment by maximization of mutual information," *International Journal of Computer Vision* **24**, pp. 137–154, 1997.

[77] B. Schiele and J. Crowley, "Transinformation for active object recognition," in *International Conference on Computer Vision*, pp. 249–254, 1998.

[78] J. Denzler and C. Brown, "Information theoretic sensor data selection for active object recognition and state estimation," *IEEE Trans. on Pattern Analysis and Machine Intelligence* **24**, pp. 145–157, 2002.

[79] J. Fisher, III and J. Principe, "A nonparametric methodology for information theoretic feature extraction," in *DARPA Image Understanding Workshop*, pp. 1077–1084, 1997.

[80] E. Gokcay and J. Principe, "Information theoretic clustering," *IEEE Trans. on Pattern Analysis and Machine Intelligence* **24**, pp. 158–171, 2002.

[81] A. Rosenfeld and A. Kak, *Digital Picture Processing*, Academic Press, 1976.

[82] R. Duda and P. Hart, *Pattern Classification and Scene Analysis*, Wiley, 1973.

[83] H. Borotschnig, L. Paletta, M. Prantl, and A. Pinz, "Appearance-based active object recognition," *Image and Vision Computing* **18**, pp. 715–727, 2000.

[84] L. Paletta, M. Prantl, and A. Pinz, "Learning temporal context in active object recognition using bayesian analysis," in *International Conference on Pattern Recognition*, pp. I: 695–699, 2000.

[85] S. Watanabe, *Pattern Recognition: Human and Mechanical*, Wiley, 1985.

[86] S. Watanabe, "Pattern recognition as a quest for minimum entropy," *Pattern Recognition* **14**, pp. 381–387, 1981.

[87] T. Hofmann and J. Buhmann, "Pairwise data clustering by deterministic annealing," *IEEE Trans. on Pattern Analysis and Machine Intelligence* **19**, pp. 1–14, 1997.

[88] A. Renyi, "On measures of entropy and information," in *Fourth Berkeley Symp. Math., Statistics and Probability*, pp. 547–561, 1960.

[89] K. Kanatani, "Geometric information criterion for model selection," *International Journal of Computer Vision* **26**, pp. 171–189, 1998.

[90] M. Hansen and B. Yu, "Model selection and the principle of minimum description length," *Journal of the American Statistical Association* , to appear.

[91] R. Kashyap, "Bayesian comparison of different classes of dynamic models using empirical data," *IEEE Trans. on Automatic Control* **22**, pp. 715–727, 1977.

[92] H. Akaike, "A new look at the statistical model identification," *IEEE Trans. on Automatic Control* **19**, pp. 716–723, 1974.

[93] J. Rissanen, "Modeling by shortest data description," *Automatica* **14**, pp. 465–471, 1978.

[94] T. Cover and J. Thomas, *Elements of Information Theory*, Wiley, 1991.

[95] G. Darbellay and I. Vajda, "Estimation of the information by an adaptive partitioning of the observation space," *IEEE Trans. on Information Theory* **45**, pp. 1315–1321, 1999.

[96] P. Besl and N. McKay, "A method for registration of 3-d shapes," *IEEE Trans. on Pattern Analysis and Machine Intelligence* **14**, pp. 239–256, 1992.

[97] A. W. Fitzgibbon, "Robust registration of 2D and 3D point sets," in *British Machine Vision Conference*, pp. 662–670, 2001.

[98] B. Vemuri and J. Aggarwal, "3D model construction from multiple views using range and intensity data," in *Conference on Computer Vision and Pattern Recognition*, pp. 435–437, 1986.

[99] G. Blais and M. Levine, "Registering multiview range data to create 3D computer objects," *IEEE Trans. on Pattern Analysis and Machine Intelligence* **17**, pp. 820–824, 1995.

[100] P. Beardsley, P. Torr, and A. Zisserman, "3D model acquisition from extended image sequences," in *European Conference on Computer Vision*, pp. 683–695, 1996.

[101] R. Koch, M. Pollefeys, and L. Van Gool, "Multi viewpoint stereo from uncalibrated sequences," in *European Conference on Computer Vision*, pp. 55–71, 1998.

[102] J. Van den Wyngaerd, L. VanGool, R. Koch, and M. Proesmans, "Invariant-based registration of surface patches," in *International Conference on Computer Vision*, pp. 301–306, 1999.

[103] C. Schmid, R. Mohr, and C. Bauckhage, "Comparing and evaluating interest points," in *International Conference on Computer Vision*, pp. 230–235, 1998.

[104] C. Tomasi and J. Shi, "Good features to track," in *Computer Vision and Pattern Recognition*, pp. 593–600, 1994.

[105] T. Cham and R. Cipolla, "A statistical framework for long-range feature matching in uncalibrated image mosaicing," in *Computer Vision and Pattern Recognition*, pp. 442–447, 1998.

[106] F. Badra, A. Qumsieh, and G. Dudek, "Robust mosaicing using zernike moments," *International Journal of Pattern Recognition and Artificial Intelligence* **13**, pp. 685–704, 1999.

[107] N. Ritter, R. Owens, J. Cooper, R. Eikelboom, and P. Van Saarloos, "Registration of stereo and temporal images of the retina," *IEEE Trans. on Medical Imaging* **18**(5), pp. 404–418, 1999.

[108] M. Burl, M. Weber, and P. Perona, "A probabilistic approach to object recognition using local photometry and global geometry," in *European Conference on Computer Vision*, 1998.

[109] A. W. Fitzgibbon, "Stochastic rigidity: Image registration for nowhere-static scenes," in *International Conference on Computer Vision*, pp. I:662–670, 2001.

[110] R. Sinkhorn, "A relationship between arbitrary positive matrices and doubly stochastic matrices," *Annals Math. Statist.* **35**, pp. 876–879, 1964.

[111] M. Srinath, P. Rajasekaran, and R. Viswanathan, *Introduction to Statistical Signal Processing with Applications*, Prentice-Hall, 1996.

[112] P. Burt and E. Adelson, "The Laplacian pyramid as a compact image code," *Communications of the ACM* **31**(4), pp. 532–540, 1983.

[113] S. Mallet, *A Wavelet Tour of Signal Processing*, Academic Press, 1999.

[114] A. Benveniste, M. Metivier, and P. Priouret, *Adaptive Algorithms and Stochastic Approximations*, Springer-Verlag, 1987.

[115] H. J. Kushner and D. S. Clark, *Stochastic Approximation Methods for Constrained and Unconstrained Systems*, Springer-Verlag, 1978.

[116] G. Saridis, "Stochastic approximation methods for identification and control – A survey," *IEEE Trans. on Automatic Control* **19**, 1974.

[117] D. Luenburger, *Optimization by Vector Space Methods*, Wiley, 1969.