Online Pose Estimation and Model Matching

Chien-Ping Lu
Research Report YALEU/DCS/RR-1112
June 1996

This work is partially supported by Darpa under ONR contract
N00014-92-J-4048 and AFOSR F49620-92-J-0465.

Abstract

Online Pose Estimation and Model Matching

Chien-Ping Lu

Yale University

1996

Computation of the relative position and orientation (*pose*) between a camera and an object from images is a classical problem in photogrammetry and computer vision. Many solution methods have been proposed. Most of them assume that the problem is to be solved in static environments where object models are exact and the correspondences between object and image features are perfectly known. This dissertation addresses the problem of online pose estimation with noisy 3D model observations and with partial or no knowledge of the feature correspondences.

With uncertainties in both 3D object space and 2D image space, object model (*structure*) and pose must be estimated simultaneously. We present a new error modeling scheme in which error measures in both 3D models and 2D projection are fused in the 3D object space using backprojection. A new pose estimation method is developed based on *alternating subspace minimization* with which the pose estimation problem becomes a series of progressive absolute orientation problems. The theory and the algorithm are validated using statistical hypothesis tests against a typical 0.05 significance level.

Extensive experiments on controlled synthetic data indicate that the new method is much more efficient than previous nonlinear techniques and is much more tolerant to noise and outliers than linear methods under most conditions.

A robust estimation scheme based on *outlier processes* is introduced for rejecting outliers in pose estimation. A continuation method is proposed for minimizing the non-convex objective function resulting from robust estimators. Outlier processes are generalized to *correspondence processes* to solve model matching problems where feature correspondences are unknown.

Online Pose Estimation and Model Matching

A Dissertation
Presented to the Faculty of the Graduate School
of
Yale University
in Candidacy for the Degree of
Doctor of Philosophy

by

Chien-Ping Lu

May 1996

To my parents

who raised me using

love, patience

and . . . sign language

Acknowledgements

I feel grateful to have Eric Mjolsness, a man of wisdom and integrity, as my advisor. This work would not exist without his inspiration and guidence. From him, I learned about the beauty of science as well as the characters of a great researcher. He is not only a teacher, but also a role model.

My collaboration with Greg Hager, a great teacher and colleague, has been exciting for his knowledge and skill in robotics and software engineering. The conversations and discussions with Anand Rangarajan, my peer advisor and friend, has been always enlightening through his insights into both the theoretical and algorithmic aspects of the problems I studied. My colleagues in the neurovision group, including Charles Garrett, Steve Gold, Suguna Pappu, and Roger Smith, have made my stay at Yale enjoyable. Charles Garrett, in particular, provided me with many technical assistances in the early years of my Ph.D. study.

I thank my younger sister Wan-Ping (Doris), who kindly and patiently took over my duty as the eldest son in the family during these years. Finally, and most importantly, I would like to thank my parents, who cannot speak and listen physically, but do speak and listen with their minds better than anyone else. Their persistent love and support, expressed through sign language and silent prayers, gave me the strength to finish this long and challenging adventure in knowledge.

This work would not be possible without the support and the computing facility of the Yale Center for Theoretical and Applied Neuroscience (now Neuroengineering and Neuroscience Center). The statistical validation results were done with the help from Tapas Kanungo using the software he developed at University of Washington. I also gratefully acknowledge the support of the Defense Advanced Research Projects Agency and the Office of Naval Research (under grant N00014-92-J-4048) and Air Force Office of Scientific Research (under grant F49620-92-J-0465).

Contents

List of Figures iii

1 Introduction 1

1.1 The Problem 1

1.2 Estimating Pose: Approaches and Issues 3

1.2.1 Initialization 3

1.2.2 Robustness to noise 4

1.2.3 Robustness to outliers 4

1.2.4 Why least-square methods? 4

1.3 The New Approach 5

1.3.1 Fusion of 3D and 2D error measures 5

1.3.2 Alternating subspace minimization 6

1.3.3 From outliers to correspondences 6

1.4 Outline of the Dissertation 7

2 Pose Estimation: A Review 8

2.1 Problem Formulation 8

2.2 Camera Model 10

2.3 The Absolute Orientation Problem 11

2.4 Classical Least Squares Methods 12

2.4.1 The Gauss-Newton method 13

2.4.2 The Levenberg-Marquardt method 14
2.5 Linear Methods 14
2.5.1 The Projective Transformation Matrix (PTM) method 15
2.5.2 The Radial Alignment Constraint (RAC) method 16
2.6 Two-Step Methods 17

3 Estimating Pose in Object Space **18**
3.1 Minimum Variance Estimation 18
3.2 The Objective Function 20
3.3 Scene Reconstruction and Error Fusion 23
3.4 Choice of Reconstruction Methods 25

4 Alternating Subspace Minimization **27**
4.1 Alternating Subspace Minimization 27
4.2 Solutions to the Absolute Orientation Phase 29
4.3 Ambiguity in the Scale of the Structure 31
4.4 Optimizing Scale 32
4.5 Probabilistic Analysis of Deviation in Scale 34
4.6 Initialization: A Weak-Perspective Approximation 38

5 Performance Evaluation **40**
5.1 Statistical Correctness and Optimality 40
5.2 Statistical Validation 42
5.2.1 Variance test with known mean 43
5.2.2 Mean-and-variance test 44
5.2.3 Results and discussions 46
5.3 Performance Comparison 46
5.3.1 Standard comparison experiments 47
5.3.2 Error measures for 3D rotations 47
5.3.3 Results and Discussions 48

6 Robust Estimation **59**

6.1 Outlier Process and Robust Estimation 59

6.2 A Continuation Method for Robust Estimation 61

6.3 Experiments 64

6.3.1 Absolute orientation 64

6.3.2 Object pose estimation 65

6.3.3 Hand-eye calibration 66

7 Model Matching **72**

7.1 From Robust Estimation to Model Matching 72

7.2 Correspondence Processes 73

7.3 A Continuation Method for Model Matching 74

7.4 2D-2D Point Matching 76

7.5 2D-2D Line-Segment Matching 79

7.5.1 Indexing points on line segments 80

7.5.2 Gaussian sum approximation 80

7.5.3 Results and discussions 82

7.6 3D-3D Point Matching 82

7.6.1 Experiments 84

8 Conclusion and Future Work **88**

8.1 What Has Been Done 88

8.2 Future Work 89

Bibliography **91**

A Solving Absolute Orientation Using Dual Quaternions **97**

B Fitting Orthonormal Matrices **99**

List of Figures

2.1 The reference frames in the pose estimation problem. 9

3.1 Modeling and imaging errors in pose estimation. 21
3.2 Imaging error measured in image space and object space. 24

4.1 Alternating subspace minimization for pose estimation. 28
4.2 Plots of $p(x)$ for $n = 10$ and $\sigma = 0.1, 0.2, 0.3, 0.4$, and 0.5. 37

5.1 Result of hypothesis test **T1** 43
5.2 Result of hypothesis test **T2** 45
5.3 Comparing average numbers of iterations used by ASM with and without scaling for Experiment **C1**. Each point in the plot represents 1,000 trials. 49
5.4 Result of Experiment **C1** for comparing ASM with and without scaling. 49
5.5 Result of Experiment **C1** for comparing ASM with and without scaling. 50
5.6 Result of Experiment **C2** for comparing ASM with and without scaling. Each point in the plot represents 1,000 trials. 50
5.7 Result of Experiment **C3** for comparing ASM with and without scaling. Each point in the plot represents 1,000 trials. 50
5.8 Result of Experiment **C4** for comparing ASM with and without scaling. 51
5.9 Corrected and uncorrected scale factors computed for different SNR. . 52
5.10 The intermediate scene points at iteration 1, 2, 5, and 12 of a typical run of ASM with scaling. 53

5.11 The intermediate scene points at iteration 1, 2, 5, and 12 of a typical run of ASM without scaling. 54
5.12 Result of Experiment **C1** for comparing linear methods and ASM. . . 55
5.13 Result of Experiment **C2** for comparing linear methods and ASM. . . 55
5.14 Result of Experiment **C3** for comparing linear methods and ASM. . . 56
5.15 Result of Experiment **C4** for comparing linear methods and ASM. . . 56
5.16 Running times and average numbers of iterations used by the tested methods. Each point in the plot represents 1,000 trials. 56
5.17 Result of Experiment **C1** for comparing ASM and the Leverberg-Marquardt method. 57
5.18 Result of Experiment **C2** for comparing ASM and the Leverberg-Marquardt method. 57
5.19 Result of Experiment **C3** for comparing ASM and the Leverberg-Marquardt method. 58
5.20 Result of Experiment **C4** for comparing ASM and the Leverberg-Marquardt method. 58

6.1 Plots of the penalty function $2\sigma^2 x(logx - 1)$ for $\sigma = 0.1, 0.3, 0.5, 0.7, 0.9$. 62
6.2 Plots of the soft-delta function $e^{-x^2/2\sigma^2}$ for $\sigma = 0.1, 0.3, 0.5, 0.7, 0.9$. 63
6.3 Comparing robust and non-robust absolute orientation methods against increasing percentages of outliers. 65
6.4 Comparing robust pose estimation methods using ASM and LM. . . . 66
6.5 Average running times of the robust pose estimation methods. 67
6.6 Average numbers of iterations of the robust pose estimation methods for different percentages of outliers. 67
6.7 The experimental setup showing the positions of the two cameras relative to the robot arm. 69

6.8 An image from the cameras showing the tracking used to generate image feature point data. 69
6.9 The projections of 35 model points as seen through cameras. 70
6.10 Results of the hand-eye calibration experiments. 71

7.1 The objective functions for translation only at different scales. 77
7.2 Comparing template matching and continuous optimization 78
7.3 A typical run of the 2D-2D point matching algorithm. 79
7.4 Approximating $\Theta(t)$ by a sum of 3 Gaussian. 81
7.5 Model line segments. 82
7.6 Model line segments overlayed on the scene image. 83
7.7 Matching line segments. 83
7.8 Results of Experiment **C1** and **C2** with different knowledge of correspondence. 86
7.9 Results of 3D-3D point matching. 87

Chapter 1

Introduction

1.1 The Problem

Determining the rigid transformation that relates an object coordinate frame to that of a camera is one of the central problems in computer vision. The available information for solving the problem is usually given in the form of a set of feature correspondences, each composed of a 3D model feature on an object and its corresponding 2D projection. The rigid transformation is called the *object pose* or the *camera pose* depending on which reference frame is referred to.

Solution methods were developed long ago for classical photogrammetry applications [57,60], where the problem is referred to as *exterior orientation.* It is known as *hand-eye calibration* in vision-based robotics where the relation between the reference frames of the camera and the robot arm has to be determined. In model-based recognition, such a problem is called *object pose estimation* or *viewpoint solving* [5,33,46,39] for the purpose of recognition.

Sometimes the problems of estimating the rigid transformation between two sets of 2D features and two sets of 3D features are referred to as the 2D-2D and 3D-3D pose estimation problems, respectively. In this case, the object pose estimation problem can be called the *3D-2D pose estimation problem.* In the following, it is

simply referred to as the *pose estimation problem.*

Recent progress in visual servoing has led to systems which make use of online pose estimation [10,26,32,68]. This is done by tracking visual features of a manipulator as it performs a set of motions. Model points are acquired in the form of 3D positions computed using the robot inverse kinematics and 2D image positions are detected by feature tracking. In telerobotics applications [27,16,6], an operator registers a geometric model with an image by pointing out specific model features in an image. Following this registration operation, features of the model are tracked and the pose of the model is updated in real time. A similar scheme could be used in enhanced reality applications [16,6] to compute a "movie" of a moving object to be rendered graphically.

All the problems described above are typical examples that suffer from the following three sources of errors:

3D modeling error Robot inverse kinematics are notoriously imprecise, particularly on smaller, flexible robots. The real 3D points that are used to generate image features on the image plane are not exactly what are given as inputs to the inverse kinematics system. 3D reconstruction by stereo triangulation or structure from motion give rise to similar errors in the 3D positions of the model features.

2D imaging error The true image coordinate of a 3D model point is perturbed by inherent noise in sensors and precision truncations during digitization. Feature extraction algorithms suffer from localization bias.

Matching error In addition to statistical error, there may be error such as mechanical backlash in robot arm manipulation. It can also be expected that the 3D to 2D correspondences will occasionally be incorrect due to operator error, mistracking, or line-of-sight occlusions. Such correspondences are called *outliers.*

1.2 Estimating Pose: Approaches and Issues

There are three basic approaches to the pose estimation problem. In *nonlinear least squares methods*, the problem is regarded as a nonlinear estimation problem. The classical approach in photogrammetry is to solve the nonlinear estimation problem by iterative methods. *Linear methods* give approximate closed-form solutions by ignoring the orthonormality constraint on the rotation matrix. Modern work using least-squares methods can be found in [46,47,23,24]. *Analytical methods* treat the problem as purely algebraic or geometrical. The problem is solved exactly for some small number of feature correspondences. In this aspect, analytical methods are sometimes referred to as *minimal information methods* [20,19,34,30,12,13]. Some methods use weak-perspective instead of perspective (pinhole) imaging model [39,2] to simplify the problem and also to achieve better efficiency.

Here, we discuss some issues regarding these three approaches.

1.2.1 Initialization

Iterative least squares methods start from an initial approximate solution, and then iteratively improve the solution according to the deviation of the observed data from the predictions based on the previous approximate solution. The methods converge to a final solution which depends on the starting point. In photogrammetry applications, initial approximate solutions are usually available. However, in modern computer vision and robotics applications such as visual-servoing and model-based recognition, initial approximate solutions are either unavailable, or are unreliable.

On the other hand, in analytical methods, a polynomial system is first derived from the algebraic constraints or the geometric configurations for a minimal number of feature correspondences, and the problem is solved by finding roots of the polynomial system. An initial approximate solution is not required.

1.2.2 Robustness to noise

In practice, any observed data are noisy. In the least squares formulation of the problem, redundancy in the data is exploited to smooth out noise. Knowledge of the underlying noise in observation can be utilized to get a better estimation.

On the other hand, there is no implied noise model in the problem formulation of analytical methods, and since only minimal information is used, no redundancy can be utilized for filtering out noise

1.2.3 Robustness to outliers

Robust M-estimate solutions to outlier rejection have led to various modified least squares methods including IRLS (Iteratively Re-weighted Least Squares) (see [38]). For the cases where very high percentage of outliers are presented, or in the extreme case where the correspondences are totally unknown as in model-based recognition, a hypothesis-and-test strategy may be more appropriate. One such procedure for outlier rejection is RANSAC [19], where a solution is computed for every combination of the minimal number of correspondences sufficient for an analytical solution. We call such combinations "evaluation sets". The solutions computed from each evaluation set are then tested against the rest of the feature correspondences. Within the RANSAC framework, it is essential that the number of evaluation sets be minimal (which implies that the size of each evaluation set be minimal), and the computation of the solution for each evaluation set be efficient. These requirements obviously call for analytical methods.

1.2.4 Why least-square methods?

According to the above discussion, neither the least squares approach nor the analytical approach is clearly better than the other. However, even in object recognition by alignment in which analytical methods are used extensively, least squares methods

are still commonly used for verification and refinement of the final solutions [16,24].

To be robust to noise, analytical methods can be applied to more feature correspondences by reducing the problem to a larger polynomial system. However, the total degree of the resulting polynomial system will be too large to handle as pointed out in the review article Huang and Netravali [37]. Therefore, the authors concluded that, after addressing the shortcoming of analytical methods, "the only option appears to be solving the nonlinear least squares problem by iterative methods".

Another support for least-squares methods comes from the fact that the error from 3D reconstructions is usually skewed and elongated. The solutions to such problems can be remarkably improved by more accurate error modeling in the forms of covariance matrices [67], which is not possible with analytical methods. Covariance matrices also contribute important information for outlier rejection and matching when used in the computation of *Mahalanobis distances.*

1.3 The New Approach

1.3.1 Fusion of 3D and 2D error measures

Most reported pose estimation methods assume that the 3D model data are accurate and well-behaved. The only source of error that needs to be considered is 2D image noise. Extremely accurate results can be achieved with precise 3D models under controlled conditions. For this reason, 3D models used in camera calibration systems often use specially designed calibration patterns that are metrically accurate and have high contrast to enhance the performance of feature extraction. Except for such calibration patterns or mechanical parts that are manufactured using predefined CAD models, exact 3D models of objects are usually difficult to get, and we have to deal with the uncertainties in the observed 3D models.

We present a new least-squares framework that takes into account uncertainties resulting from both 3D reconstruction and 2D imaging. Under this framework,

object model (structure) and pose are estimated simultaneously. When the pose and structure estimation is applied recursively to an image sequence, it solves the structure-from-motion problem where the object model is the structure estimated in the previous frame.

1.3.2 Alternating subspace minimization

The problem of estimating pose and structure is nonlinear. We present a subspace minimization method that optimizes pose and structure alternatively. For fixed pose, the structure is solved linearly. Given fixed structure, the pose is obtained by solving an absolute orientation problem which is generally much easier than the pose estimation problem. This approach turns the pose estimation problem into a series of progressive absolute orientation problems, which also lead to a good initialization scheme based on weak-perspective projection.

The new algorithm has been compared to a number of previously existing least squares methods. All methods have been tested extensively on synthetic data with varying noise, percentages of outliers, and numbers of reference points. Experimental data indicate that the new methods require many fewer function evaluations than classical nonlinear techniques and are much more tolerant to noise and outliers than linear methods under most conditions.

1.3.3 From outliers to correspondences

To deal with matching error, it is important that a solution method be able to successfully "reject" these outliers. A robust estimation scheme based on *outlier processes* is introduced for rejecting possible outliers. A continuation method is proposed for minimizing the non-convex objective function resulting from robust estimators and outlier processes.

In the presence of outliers, the feature correspondences can be said to be "partially" known. An important family of problems, in which the correspondences are

totally unknown, is called *model matching* or *model registration*. It occurs when there is no human operator for handling the registration operations described above, or when such operations need to be automated. The correspondences between the model features and the scene features need to be established.

The *alignment method* for model matching [62,39] can be considered as an extension to RANSAC for the cases that the feature correspondences are "totally" unknown. The tentative feature correspondences are generated from all possible pairings of model features and image features, and the algorithm tries to *reject* those correspondences that are incorrect. In this manner, the concept of outlier rejection can be generalized to correspondence establishment. A similar extension is applied to robust IRLS algorithms in which every possible pair of model and scene features has a weight used to represent the "strength" of the associated correspondence. Outlier processes are generalized to *correspondence processes* so that the more difficult model matching problems can be solved in a manner similar to outlier here rejection in robust estimation. A similar continuation method is applied here to 2D-2D point matching, 2D-2D line-segment matching and 3D-3D point matching.

1.4 Outline of the Dissertation

The remainder of this dissertation is organized as follows. The next chapter describes the least squares formulation of the pose estimation problem and gives some classical solutions. Chapter 3 presents a new statistical and computational framework for pose estimation. Chapter 4 presents a practical *alternating subspace minimization* algorithm based on our new framework. Chapter 5 gives statistical validation of our theory and implementation. Detailed performance analysis using large scale simulations are performed to compare our method to existing methods. Chapter 6 discusses robust methods for outlier rejections and Chapter 7 extends robust methods to deal with the cases that feature correspondences are unknown. Finally, Chapter 8 concludes this dissertation and discusses some possible extensions.

Chapter 2

Pose Estimation: A Review

In this chapter, we give a formulation of the pose estimation problem, introduce the related absolute orientation problem, and review previous work on least-square methods including linear and nonlinear methods.

2.1 Problem Formulation

The mapping from 3D points to 2D image coordinates can be formalized as follows. Given a set of 3D coordinates of model points $\mathbf{x} = (x, y, z)^t$ in an object reference frame, and the corresponding coordinates (the scene points) $\mathbf{y} = (x', y', z')^t$ in a camera reference frame, the two frames can be related by a rigid transformation as

$$\mathbf{y} = R\mathbf{x} + \mathbf{t}, \tag{2.1}$$

where

$$R = \begin{pmatrix} \mathbf{r}_1^t \\ \mathbf{r}_2^t \\ \mathbf{r}_3^t \end{pmatrix} \quad \text{and} \quad \mathbf{t} = \begin{pmatrix} t_1 \\ t_2 \\ t_3 \end{pmatrix} \tag{2.2}$$

are a rotation matrix and a translation vector, respectively.

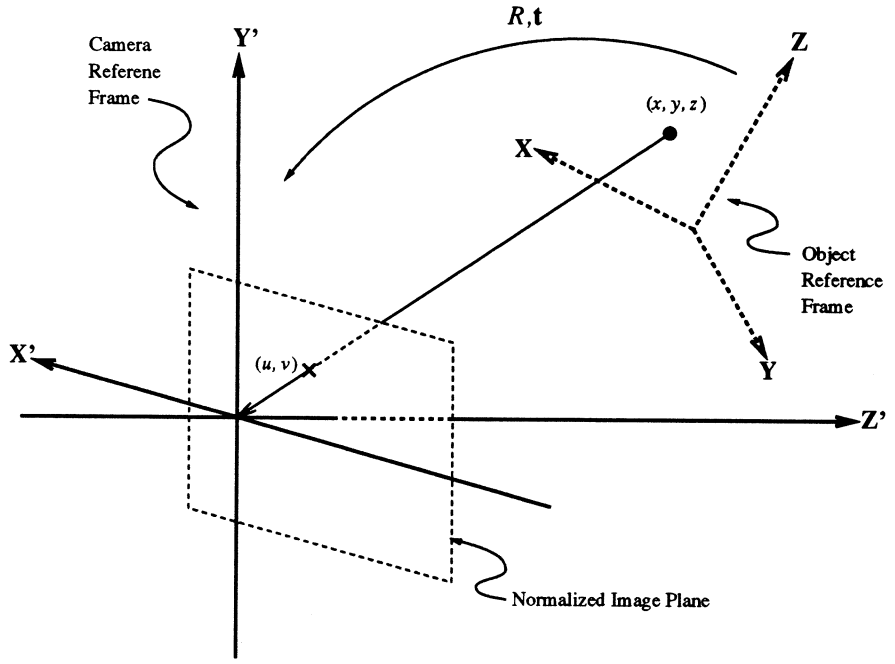

Figure 2.1: The reference frames in the pose estimation problem.

The camera reference frame is chosen so that the projection center of the camera is at the origin, and the optical axis points in the positive z direction. The model points $\mathbf{x}$ are projected to the plane with equation $z' = 1$, referred to as the *normalized image plane*, in the camera reference frame. The resulting projections $\mathbf{u} = (u, v)^t$ are called the *normalized image coordinates.* Under the idealized pinhole imaging model, the image vector (or the backprojection vector) $\mathbf{v} = (u, v, 1)^t$, the scene point $\mathbf{y}$, and the center of projection are collinear. The line extended by a image vector is called a *backprojection line.* The projection equation can be written as

$$\mathbf{v} = \frac{1}{\mathbf{r}_3^t \mathbf{x} + t_3}(R\mathbf{x} + \mathbf{t}), \tag{2.3}$$

which is known as the *collinearity equation* in photogrammetry literature. Due to errors that usually occur in the imaging process, the observed image vector $\tilde{\mathbf{v}}$ is corrupted by some 2D noise vector $\boldsymbol{\epsilon}$ as

$$\tilde{\mathbf{v}} = (\tilde{\mathbf{u}}^t, 1)^t, \quad \tilde{\mathbf{u}} = \mathbf{u} + \boldsymbol{\epsilon}. \tag{2.4}$$

2.2 Camera Model

The imaging geometry of real cameras is somewhat more complex than the pinhole model. This causes various types of optical distortions [21]. Among all studied distortion types, radial distortion, which produces a positive or negative displacement of a given image point along the radial direction from the principle point of the lens, is the most dominant. If only radial distortion is considered, the normalized image coordinates $(\tilde{u}, \tilde{v})^t$ can be corrected from the distorted (uncorrected) ones $(\hat{u}, \hat{v})^t$ approximately by

$$\tilde{u} = \hat{u}(1 + \kappa r^2) \tag{2.5}$$

$$\tilde{v} = \hat{v}(1 + \kappa r^2), \tag{2.6}$$

where $r^2 = \hat{u}^2 + \hat{v}^2$ and κ is the radial distortion coefficient.

In addition, the digitizing hardware imposes its own coordinate system on the digitized image. The mapping from the sensor coordinates $(m, n)^t$ to the distorted image coordinates $(\hat{u}, \hat{v})$ is defined by

$$\hat{u} = (m - m_0)/f_u \tag{2.7}$$

$$\hat{v} = (n - n_0)/f_v, \tag{2.8}$$

where (m_0, n_0) is the image center in sensor coordinates, and (f_u, f_v) are the horizontal and vertical image scales, respectively.

Assuming known distances d_u and d_v between adjacent sensor elements in both horizontal and vertical directions, the image scales can be represented by camera focal length f and correction factor s_u for horizontal scale by

$$f_u = s_u f/d_u, \quad f_v = f/d_v. \tag{2.9}$$

The reason for introducing the correction factor s_u is as follows. The image of 3D scene is first formed on the sensor plane. The digitizing hardware scans and digitizes the sensor image, and stores the result in the frame buffer. For commonly used CCD

cameras, both the sensor plane and the frame buffer are composed of discrete elements, therefore the mapping between them should be one-to-one. It is true for the vertical coordinates since the sensor image is scanned line by line horizontally. However, for horizontal coordinates, the mapping is not one-to-one due to the resolution differences between the camera and the frame buffer.

Both representations give two degrees of freedom. They are equivalent mathematically since their relation as specified by (2.9) is one-to-one. We choose to use the (f_u, f_v) representation instead of the (f, s_u) representation for two reasons: first, the calibration code using the former representation does not need to know d_u and d_v; second, the mapping from sensor to frame buffer is linear in f_u and f_v while it involves a nonlinear term fs_u if the other representation is used.

The parameters f_u, f_v, m_0, n_0 and κ are referred to as the camera intrinsic parameters. In the rest of this dissertation, we assume that the camera intrinsic parameters are known, and normalized image coordinates can be computed from sensor coordinates accordingly. The problem of estimating the camera intrinsic parameters as well as the pose is referred to as *camera calibration* [61,45].

2.3 The Absolute Orientation Problem

If the 3D camera frame coordinates $\mathbf{y}$ have been reconstructed physically (for example, by range sensing) or computationally (for example, by stereo matching or structure-from-motion), we have

$$\tilde{\mathbf{y}} = R\mathbf{x} + \mathbf{t} + \boldsymbol{\eta}, \tag{2.10}$$

where $\tilde{\mathbf{y}}$ are observed 3D camera frame coordinates, and $\boldsymbol{\eta}$ are noise vectors that account for the uncertainties in 3D reconstruction. Assume that $\tilde{\mathbf{y}}$ has a covariance matrix $\boldsymbol{\Sigma}$ due to $\boldsymbol{\eta}$.

The process of determining R and $\mathbf{t}$ from $\tilde{\mathbf{y}}$ and $\mathbf{x}$ is called *absolute orientation* or *3D-3D pose estimation.* It can be generalized to include an unknown scaling factor,

in which case the transformation involved is known as a *similarity transformation*, and Equation (2.1) takes the form

$$\mathbf{y} = sR\mathbf{x} + \mathbf{t}. \tag{2.11}$$

Several closed-form solutions have been proposed for the equally-weighted least-squares minimization corresponding to Equation (2.11) [36,4,35,63] for the cases that $\boldsymbol{\eta}$ are independently and identical distributed, and are also isotropic such that $\boldsymbol{\Sigma} = \sigma^2 I$. These methods can be very easily extended to find scalar-weighted least squares solutions. An algorithm based on linear subspace decomposition is presented in [64] for non-isotropic, but independently and identically distributed noise. Weng *et. al.* presented a two-stage matrix-weighted least-squares solution for more general heterogeneous and non-isotropic noise [67]. The orthonormality constraint on rotation matrix is ignored in the first stage in order to get a closed-form solution, which is in the form of a 3-by-3 matrix and a 3-vector. An improvement is made by finding a rotation matrix that best fits the 3-by-3 matrix.

In practice, it was pointed out in [25,18] that the pose estimation problem can be greatly simplified when 3D depth information is available, since this avoids some of the nonlinearities resulting from projection. This is further confirmed by the fact that good closed-form least squares solutions exist for absolute orientation, although they use linear approximations such as the aforementioned approximate closed-from solution by Weng *et. al.* It appears that the nonlinearity coming from projection contributes much more to the difficulty of problem than that coming from orthonormality of the rotation matrix.

2.4 Classical Least Squares Methods

The rotation matrix R is subject to the orthonormal constraint

$$\mathbf{r}_i^t \mathbf{r}_j = \delta_{ij},\ i, j = 1, 2, 3, \tag{2.12}$$

and has three degrees of freedom. It is related to the Euler angles ϕ, θ, ψ by

(2.13)
$$R = \begin{pmatrix} \cos\theta\cos\psi & \cos\theta\sin\psi & -\sin\theta \\ -\cos\phi\sin\psi + \sin\phi\sin\theta\cos\psi & \cos\phi\cos\psi + \sin\phi\sin\theta\sin\psi & \sin\phi\cos\theta \\ \sin\phi\sin\psi + \cos\phi\sin\theta\cos\psi & -\sin\phi\cos\psi + \cos\phi\sin\theta\sin\psi & \cos\phi\cos\theta \end{pmatrix}.$$

In classical photogrammetry, the pose estimation problem is solved by minimizing image error, which is equivalent to solving the nonlinear least-squares problem

(2.14)
$$F(\boldsymbol{\theta}) = \|\mathbf{f}(\boldsymbol{\theta})\|^2 = \sum_i \left[(\tilde{u}_i - \frac{\mathbf{r}_1^t \mathbf{x}_i + t_1}{\mathbf{r}_3^t \mathbf{x}_i + t_3})^2 + (\tilde{v}_i - \frac{\mathbf{r}_2^t \mathbf{x}_i + t_2}{\mathbf{r}_3^t \mathbf{x}_i + t_3})^2 \right],$$

where $\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3$ are functions of three Euler angles ϕ, θ, ψ, and $\boldsymbol{\theta} = (\phi, \theta, \psi, t_1, t_2, t_3)^t$.

Two commonly used methods for minimizing $F(\boldsymbol{\theta})$ are Gauss-Newton and Levenberg-Marquardt.

2.4.1 The Gauss-Newton method

The Gauss-Newton method is a classical technique for solving nonlinear least-squares problems such as (2.14). It operates by iteratively linearizing the collinearity equation around the current approximate solution $\boldsymbol{\theta}$ using a first-order Taylor's expansion

(2.15)
$$f(\boldsymbol{\theta} + \Delta\boldsymbol{\theta}) = \|\mathbf{f}(\boldsymbol{\theta} + \Delta\boldsymbol{\theta})\|^2$$

(2.16)
$$\approx \|\mathbf{f}(\boldsymbol{\theta}) + J(\boldsymbol{\theta})\Delta\boldsymbol{\theta}\|^2,$$

and then solving the linearized system in $\Delta\boldsymbol{\theta}$

(2.17)
$$-J^t(\boldsymbol{\theta})\mathbf{f}(\boldsymbol{\theta}) = J^t(\boldsymbol{\theta})J(\boldsymbol{\theta})\Delta\boldsymbol{\theta}$$

for the next approximate solution $\boldsymbol{\theta} + \Delta\boldsymbol{\theta}$, where $J(\boldsymbol{\theta}) = \frac{\partial F}{\partial \theta}$ is the Jacobian of F at $\boldsymbol{\theta}$. The Gauss-Newton method relies on a good local linearization. If the initial approximate solution is good enough, it should converge very quickly to the correct solution. However, when the current solution is far from the correct one and/or the

linear system is ill-conditioned, it may even fail to converge. It is reported in [31] that for the Gauss-Newton method to work, the initial approximate solutions have to be within 10% of scale for the translation and within 15° for each of the three rotation angles.

2.4.2 The Levenberg-Marquardt method

The Levenberg-Marquardt method solves the least squares problem by solving the stabilized linear system

$$-J^t(\boldsymbol{\theta})\mathbf{f}(\boldsymbol{\theta}) = (\lambda D + J^t(\boldsymbol{\theta})J(\boldsymbol{\theta}))\Delta\boldsymbol{\theta}, \tag{2.18}$$

where D is a scaling matrix and λ is an adjustable parameter. It can be regarded as an interpolation of steepest descent and the Gauss-Newton method. When the current solution is far from the correct one, the algorithm behaves like a steepest descent method: slow but guaranteed to converge. When the current solution is close to the correct solution, it becomes a Gauss-Newton method. It has become a standard technique for nonlinear least squares problems, and has been widely adopted in computer vision literature (for example, see [47], [65]).

2.5 Linear Methods

With more data points, linear least-squares methods solve for the 9 parameters (or part of them) in the 3-by-3 rotation matrix linearly by ignoring the orthonormality constraint. The solution can then be improved by finding the orthonormal matrix that best fits the 3-by-3 matrix. For our experiments, we have included an orthonormalization step. Representive work on linear squares methods can be found in [1,69,17,61,45].

2.5.1 The Projective Transformation Matrix (PTM) method

By ignoring the orthonormality constraint, the 3-by-3 rotation matrix $R = (r_{ij})$ and the translation vector $\mathbf{t}$ can be solved in closed form up to a scale factor using the collinearity equation (2.3). This method is adapted from the original PTM method [1,69,17] which uses the 4-by-3 projective transformation matrix to encode the image center and image scales in addition to R and $\mathbf{t}$.

Cross-multiplying the collinearity equation (2.3) yields

$$(\mathbf{r}_3^t\mathbf{x} + t_3)\tilde{u} = \mathbf{r}_1^t\mathbf{x} + t_1 \tag{2.19}$$

$$(\mathbf{r}_3^t\mathbf{x} + t_3)\tilde{v} = \mathbf{r}_2^t\mathbf{x} + t_2. \tag{2.20}$$

Dividing by t_3 on both sides, Equations (2.19) and (2.20) become

$$((\mathbf{r}_3')^t\mathbf{x} + 1)\tilde{u} = (\mathbf{r}_1')^t\mathbf{x} + t_1' \tag{2.21}$$

$$((\mathbf{r}_3')^t\mathbf{x} + 1)\tilde{v} = (\mathbf{r}_2')^t\mathbf{x} + t_2', \tag{2.22}$$

where $\mathbf{r}_1' = t_3^{-1}\mathbf{r}_1, \mathbf{r}_2' = t_3^{-1}\mathbf{r}_2, \mathbf{r}_3' = t_3^{-1}\mathbf{r}_3, t_1' = t_3^{-1}t_1$, and $t_2' = t_3^{-1}t_2$. Six pairs of a model point $\mathbf{x}$ and an image vector $\mathbf{v} = (\tilde{u}, \tilde{v}, 1)^t$ are required for solving for the 11 parameters $R' = (\mathbf{r}_1', \mathbf{r}_2', \mathbf{r}_3')^t, t_1'$ and t_2', since each of them contributes two linear equations ((2.21) and (2.22)). A least-squares solution is found by singular value decomposition of the resulting system.

What needs to be done next is to decompose R' into the scale factor t_3^{-1} and the orthonormal matrix R. t_3 can be determined by

$$\det(R') = \det(t_3^{-1}R) = t_3^{-1}. \tag{2.23}$$

R is computed as the rotation matrix that best fits $\text{sign}(\det(R'))R' = |t_3^{-1}|R$. Such decomposition can be done in a straightforward manner by solving a 4-point absolute orientation problem (see Appendix B).

Now with known t_3 and R, the remaining unknowns t_1 and t_2 can be calculated by solving the overdetermined system of Equations (2.19) and (2.20) for each pair

of image vector and model point. To improve the accuracy, t_3 can be recalculated together with t_1 and t_2.

2.5.2 The Radial Alignment Constraint (RAC) method

Tsai introduced a two-step method for camera calibration [61,45]. The first stage makes use of the radial alignment constraint which can be formulated in our notation as

$$\frac{\hat{u}}{\hat{v}} = \frac{\tilde{u}}{\tilde{v}} = \frac{\mathbf{r}_1^t \mathbf{x} + t_1}{\mathbf{r}_2^t \mathbf{x} + t_2}. \tag{2.24}$$

The radial alignment constraint says that the vector defined by the orthographic projection of a 3D model point, the normalized image vector $(\tilde{u}, \tilde{v}, 1)^t$, and the distorted image vector $(\hat{u}, \hat{v}, 1)^t$ under just radial distortion are all parallel. Using this formula, the horizontal scale factor and all of the camera pose parameters except for t_3 can be computed using linear techniques. The second stage computes the remaining parameters by applying nonlinear optimization to the collinearity equation (2.3) using the values computed in the first stage. With known intrinsic parameters or ignoring radial distortion, normalized image vectors are available and t_3 can also be computed linearly using the collinearity equation.

We observe that (2.24) can be converted by division and cross-multiplication to

$$\tilde{v}(\mathbf{r}_1')^t \mathbf{x} + \tilde{v} t_1' = \tilde{u}(\mathbf{r}_2')^t \mathbf{x} + \tilde{u}, \tag{2.25}$$

where $\mathbf{r}_1' = t_2^{-1}\mathbf{r}_1, \mathbf{r}_2' = t_2^{-1}\mathbf{r}_2$, and $t_1' = t_2^{-1} t_1$. Seven pairs of a model point $\mathbf{x}$ and its corresponding image vector $\tilde{\mathbf{v}} = (\tilde{u}, \tilde{v}, 1)^t$ are required to solve for the 7 parameters $\mathbf{r}_1', \mathbf{r}_2'$, and t_1', since each of them contribute one linear equation (2.25). Least-squares solution is computed by singular value decomposition of the resulting system.

Given these values, R and t_2 are determined from $\mathbf{r}_1'$ and $\mathbf{r}_2'$ as follows. The absolute value of t_2 is determined by

$$|t_2| = \|\mathbf{r}_1'\|^{-1} \quad \text{or} \quad \|\mathbf{r}_2'\|^{-1}. \tag{2.26}$$

The sign of t_2 also determines the signs of $\mathbf{r}_1, \mathbf{r}_2$ and t_1. It should be chosen such that $\tilde{u}$ and $\tilde{v}$ have the same sign as $\mathbf{r}_1^t\mathbf{x}+t_1$ and $\mathbf{r}_2^t\mathbf{x}+t_2$, respectively. The point $\mathbf{x}$ used to determine the sign of t_2 can be chosen as some model point whose image point is far away from the image center. Then $\mathbf{r}_1$ and $\mathbf{r}_2$ are computed from t_2, $\mathbf{r}'_1$ and $\mathbf{r}'_2$. Using the orthonormality of the rotation matrix, $\mathbf{r}_3$ can be computed from $\mathbf{r}_1$ and $\mathbf{r}_2$.

With known t_1, t_2 and R, the remaining unknown, t_3, can be calculated by solving the overdetermined system of Equations (2.19) and (2.20) for each pair of image vector and model point. Again, to improve accuracy, t_1 and t_2 can be recalculated together with t_3.

2.6 Two-Step Methods

Linear least-squares methods are very fast. However, it should be noted that they provide only an *approximate* closed-form solution. The orthonormality constraint on rotation matrices is not fully considered in the solution process. Consequently, in the presence of noise, the 3-by-3 matrix is not exactly orthonormal, and the accuracy of the final result is relatively poor even when it is further improved by finding the closest orthonormal matrix.

A two-step method solves the problem in two stages. In the first stage, a linear algorithm is employed to get an approximate closed-form solution. In the second stage, a nonlinear method uses the previous closed-form solution as an initial guess to search for a better result. The same idea can be found in structure-from-motion [65] and camera calibration [61,66].

Although two-step methods seem to be the answer to the issues of initialization and robustness, problems remain. Without the orthonormality constraint, linear methods not only overfit noise, but also overfit the outliers if they exist. The latter can lead to a solution that is meaningless. When used as a starting point, such a solution is very likely to cause subsequent nonlinear optimization to fail. These problems motivate the need for a better initialization method than linear methods.

Chapter 3

Estimating Pose in Object Space

In this chapter, we address the problem of estimating pose in the presence of uncertainty in model observation in addition to that in image space. We present an error modeling scheme in which error measures in both object space and image space are fused into a single error measure by backprojection image error into object space.

3.1 Minimum Variance Estimation

Throughout the rest of this dissertation, we follow the conventions described below. An "observed" or "noise-perturbed" quantity is designated as $\tilde{\boldsymbol{y}}$ which is a random variable. Likewise, an "estimate" is written as $\hat{\boldsymbol{y}}$. A quantity with a subscript, $\boldsymbol{y}_i$, represents either a typical member in the set $\{\boldsymbol{y}_i\}$, or the set itself depending on the context.

Different kinds of errors in the observations contribute to different kinds of errors in the pose solution. In order to solve the problem properly when both modeling and imaging errors present, we need to be able to measure their contributions to the estimation errors systematically. The development of our method for the pose estimation problem is based on what is called *minimum variance estimation* framework in which the error measures are represented elegantly as covariance matrices. The minimum

variance estimation framework is outlined as follows.

Suppose that an observation vector $\tilde{\boldsymbol{y}}$ is related linearly to the parameter vector $\boldsymbol{\theta}$ to be estimated by

$$\tilde{\boldsymbol{y}} = A\boldsymbol{\theta} + \boldsymbol{\eta}_{\tilde{\boldsymbol{y}}}, \tag{3.1}$$

where A is called *design matrix*. Assuming that $\boldsymbol{\eta}$ is a Gaussian noise vector with a zero mean ($E(\boldsymbol{\eta}) = 0$) and a covariance matrix $\boldsymbol{\Sigma}_{\tilde{\boldsymbol{y}}} = E(\boldsymbol{\eta}_{\tilde{\boldsymbol{y}}}\boldsymbol{\eta}_{\tilde{\boldsymbol{y}}}^t)$, the "best" (in the sense of minimum variance) linear, unbiased estimator (BLUE) of $\boldsymbol{\theta}$ is the one that minimizes the objective function

$$(\tilde{\boldsymbol{y}} - A\boldsymbol{\theta})^t \boldsymbol{\Sigma}_{\tilde{\boldsymbol{y}}}^{-1} (\tilde{\boldsymbol{y}} - A\boldsymbol{\theta}), \tag{3.2}$$

and the resulting estimator $\hat{\boldsymbol{\theta}}$ is

$$(A^t \boldsymbol{\Sigma}_{\tilde{\boldsymbol{y}}}^{-1} A)^{-1} A^t \boldsymbol{\Sigma}_{\tilde{\boldsymbol{y}}}^{-1} \tilde{\boldsymbol{y}} \tag{3.3}$$

with a covariance matrix

$$\boldsymbol{\Sigma}_{\hat{\boldsymbol{\theta}}} = (A^t \boldsymbol{\Sigma}_{\tilde{\boldsymbol{y}}}^{-1} A)^{-1}. \tag{3.4}$$

The minimum variance estimator in the Gaussian case is also the *linear* minimum variance estimator in the general case, without any Gaussian assumption [41]. This implies that when the best "linear" estimator serves our needs, we can safely assume that the underlying noise distribution $\boldsymbol{\eta}_{\tilde{\boldsymbol{y}}}$ is a Gaussian. Only knowledge of second-order statistics and below is required. Note that such a best estimator may be computed using nonlinear methods.

Similarly, in nonlinear cases where $\tilde{\boldsymbol{y}}$ is related to $\boldsymbol{\theta}$ by a nonlinear equation

$$\tilde{\boldsymbol{y}} = \boldsymbol{f}(\boldsymbol{\theta}) + \boldsymbol{\eta}, \tag{3.5}$$

the best linear estimator of $\boldsymbol{\theta}$ is the one that minimizes the objective function

$$(\tilde{\boldsymbol{y}} - \boldsymbol{f}(\boldsymbol{\theta}))^t \boldsymbol{\Sigma}_{\tilde{\boldsymbol{y}}}^{-1} (\tilde{\boldsymbol{y}} - \boldsymbol{f}(\boldsymbol{\theta})), \tag{3.6}$$

and the covariance matrix of $\hat{\boldsymbol{\theta}}$ can be approximated by

$$\boldsymbol{\Sigma}_{\hat{\boldsymbol{\theta}}} \approx \left(\frac{\partial \boldsymbol{f}(\hat{\boldsymbol{\theta}})}{\partial \boldsymbol{\theta}}^{t} \boldsymbol{\Sigma}_{\tilde{\boldsymbol{y}}}^{-1} \frac{\partial \boldsymbol{f}(\hat{\boldsymbol{\theta}})}{\partial \boldsymbol{\theta}}\right)^{-1}, \tag{3.7}$$

where $\frac{\partial \boldsymbol{f}(\hat{\boldsymbol{\theta}})}{\partial \boldsymbol{\theta}}$ is the Jacobian matrix of f evaluated at $\hat{\boldsymbol{\theta}}$. Since f is nonlinear, iterative methods are required to solve for $\hat{\boldsymbol{\theta}}$.

There are cases in which the observation vector $\tilde{\boldsymbol{y}}$ is obtained indirectly as a function of another observation $\tilde{\boldsymbol{x}}$ which is contaminated by a noise with a zero mean and a covariance matrix $\boldsymbol{\Sigma}_{\tilde{\boldsymbol{x}}}$. The covariance matrix $\boldsymbol{\Sigma}_{\tilde{\boldsymbol{y}}}$ of $\tilde{\boldsymbol{y}}$ to be used in (3.4) or (3.7) can be computed as

$$\boldsymbol{\Sigma}_{\tilde{\boldsymbol{y}}} = \frac{\partial \tilde{\boldsymbol{y}}(\tilde{\boldsymbol{x}})}{\partial \tilde{\boldsymbol{x}}} \boldsymbol{\Sigma}_{\tilde{\boldsymbol{x}}} \frac{\partial \tilde{\boldsymbol{y}}(\tilde{\boldsymbol{x}})}{\partial \tilde{\boldsymbol{x}}}^{t}, \tag{3.8}$$

With the minimum variance estimation framework, we can find the best (in the sense of minimum variance) linear estimator by minimizing a least squares objective function such as (3.6). The measures of the uncertainties, expressed in the form of covariance matrices, can be propagated from observations to estimations according to (3.4) and (3.7) (see [29] for details of theory).

Since the mean and covariance of $\hat{\boldsymbol{\theta}}$ comprise a sufficient statistic for $\boldsymbol{\theta}$ if it is a Gaussian, solving for optimal $\hat{\boldsymbol{\theta}}$ and the associated covariance matrix can be considered as finding the "distribution" of $\hat{\boldsymbol{\theta}}$.

3.2 The Objective Function

When a set of model points is determined as the result of an inverse kinematics computation, stereo triangulation, or structure-from-motion algorithms, the points are noisy. Instead of writing each of them as an exact model point $\mathbf{x}_i$, we have a "perturbed" model point $\tilde{\mathbf{x}}_i$ which is related to $\mathbf{x}_i$ by

$$\tilde{\mathbf{x}}_i = \mathbf{x}_i + \boldsymbol{\zeta}_i, \tag{3.9}$$

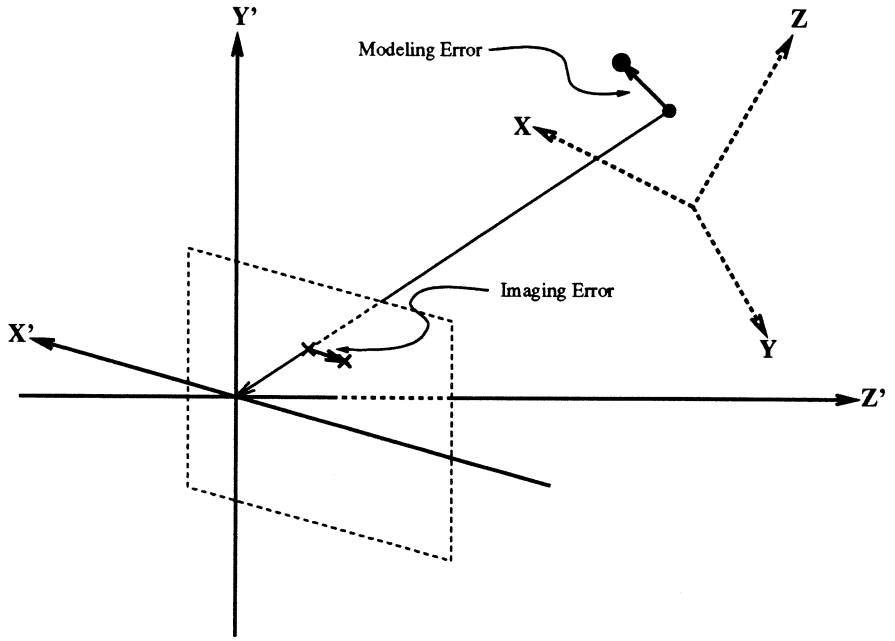

Figure 3.1: Modeling and imaging errors in pose estimation.

where $\boldsymbol{\zeta}_i$ is a noise vector with a covariance matrix $\boldsymbol{\Sigma}_{\tilde{\mathbf{x}}_i}$. A specialization to the case where exact model data is available can be achieved by having the covariance matrix $\boldsymbol{\Sigma}_{\tilde{\mathbf{x}}_i} = \mathbf{0}$.

In the case of noisy model observations, we need to estimate both the pose $(R, \mathbf{t})$ and the model points $\mathbf{x}_i$ simultaneously. Equivalently, we estimate the coordinates of the model points in the object reference frame, the scene points $\mathbf{y}_i$, which are sometimes referred to as the *structure*. We denote the noisy observation $\tilde{\mathbf{y}}_i$ of $\mathbf{y}_i$ as

$$\tilde{\mathbf{y}}_i = \mathbf{y}_i + R\boldsymbol{\zeta}_i, \tag{3.10}$$

which has a covariance matrix $\boldsymbol{\Sigma}_{\tilde{\mathbf{y}}_i} = R\boldsymbol{\Sigma}_{\tilde{\mathbf{x}}_i}R^t$. The modeling error measured in the object space for a scene point $\mathbf{y}_i$ is thus

$$\tilde{\mathbf{y}}_i - \mathbf{y}_i = R\tilde{\mathbf{x}}_i + \mathbf{t} - \mathbf{y}_i. \tag{3.11}$$

The imaging error for $\mathbf{y}_i$ is given by

$$\tilde{\mathbf{u}}_i - \boldsymbol{\pi}(\mathbf{y}_i), \tag{3.12}$$

where

$$\boldsymbol{\pi}(\mathbf{y}_i) = (x_i'/z_i', y_i'/z_i')^t, \tag{3.13}$$

is a predicted image point given $\mathbf{y}_i$. It is the difference between the observed image point $\tilde{\mathbf{u}}_i$ and the projection of the estimated scene point $\mathbf{y}_i$ on the normalized image plane. The objective function for estimating the pose and the structure can be written as

(3.14)

$$\begin{aligned} & f(R, \mathbf{t}, \mathbf{y}_i) \\ & = \sum_i (R\tilde{\mathbf{x}}_i + \mathbf{t} - \mathbf{y}_i)^t \boldsymbol{\Sigma}_{\tilde{\mathbf{y}}_i}^{-1} (R\tilde{\mathbf{x}}_i + \mathbf{t} - \mathbf{y}_i) + \sum_i (\tilde{\mathbf{u}}_i - \boldsymbol{\pi}(\mathbf{y}_i))^t \boldsymbol{\Sigma}_{\tilde{\mathbf{u}}_i}^{-1} (\tilde{\mathbf{u}}_i - \boldsymbol{\pi}(\mathbf{y}_i)) \\ & = f_{mod}(R, \mathbf{t}, \mathbf{y}_i) + f_{img}(\mathbf{y}_i), \end{aligned}$$

where f_{mod} and f_{img} are shorthands for the first and the second terms in the objective function which represent 3D modeling error and 2D imaging error, respectively.

Minimizing the objective function (3.14) directly over $R, \mathbf{t}$ and $\mathbf{y}_i$ involves searching in a parameter space with very high dimension $(3n+6)$. We can reduce the search space by the *subspace decomposition* techniques [67,11]. The basic idea is to decouple the pose and structure by representing the scene point $\mathbf{y}_i$ in terms of $(R, \mathbf{t})$ such that

$$\begin{aligned} \min_{R,\mathbf{t},\mathbf{y}_i} f(R, \mathbf{t}, \mathbf{y}_i) &= \min_{R,\mathbf{t}} \min_{\mathbf{y}_i} f(R, \mathbf{t}, \mathbf{y}_i) \\ &= \min_{R,\mathbf{t}} f(R, \mathbf{t}, \mathbf{y}_i^*(R, \mathbf{t})) \\ &= \min_{R,\mathbf{t}} g(R, \mathbf{t}), \end{aligned} \tag{3.15}$$

where

$$\mathbf{y}_i^*(R, \mathbf{t}) = \arg\min_{\mathbf{y}_i} f(R, \mathbf{t}, \mathbf{y}_i). \tag{3.16}$$

The problem becomes a minimization of $g(R, \mathbf{t})$ over the pose only. Classical optimization techniques like the Gauss-Newton method, or the Levenberg-Marquardt method as introduced in Section 2.4 can be used.

The remaining problem becomes finding the solution to $\mathbf{y}^*(R,\mathbf{t})$. It is difficult since f_{img} is nonlinear. Note f_{mod} is a quadratic function of $\mathbf{y}_i$ for fixed R and $\mathbf{t}$. If f_{img} can be approximated by a quadratic function of $\mathbf{y}_i$ such as

$$\sum_i (\mathbf{z}_i - \mathbf{y}_i)^t \boldsymbol{\Sigma}_{\mathbf{z}_i}^{-1} (\mathbf{z}_i - \mathbf{y}_i) \tag{3.17}$$

then the overall linear minimum variance estimator $\mathbf{y}_i^*(R,t)$ of $\mathbf{y}_i$ for fixed R and $\mathbf{t}$ is given by

$$\begin{aligned} \mathbf{y}_i^*(R,t) &= \boldsymbol{\Sigma}_{\mathbf{z}_i}(\boldsymbol{\Sigma}_{\tilde{\mathbf{y}}_i} + \boldsymbol{\Sigma}_{\mathbf{z}_i})^{-1}\tilde{\mathbf{y}}_i + \boldsymbol{\Sigma}_{\tilde{\mathbf{y}}_i}(\boldsymbol{\Sigma}_{\tilde{\mathbf{y}}_i} + \boldsymbol{\Sigma}_{\mathbf{z}_i})^{-1}\mathbf{z}_i \\ &= \tilde{\mathbf{y}}_i + \boldsymbol{\Sigma}_{\tilde{\mathbf{y}}_i}(\boldsymbol{\Sigma}_{\tilde{\mathbf{y}}_i} + \boldsymbol{\Sigma}_{\mathbf{z}_i})^{-1}(\mathbf{z}_i - \tilde{\mathbf{y}}_i). \end{aligned} \tag{3.18}$$

The covariance matrix of $\mathbf{y}_i^*(R,\mathbf{t})$ is given by

$$\boldsymbol{\Sigma}_{\mathbf{y}^*} = \boldsymbol{\Sigma}_{\tilde{\mathbf{y}}_i}(\boldsymbol{\Sigma}_{\mathbf{z}_i} + \boldsymbol{\Sigma}_{\tilde{\mathbf{y}}_i})^{-1}\boldsymbol{\Sigma}_{\tilde{\mathbf{y}}_i}. \tag{3.19}$$

The next section is devoted to the topic of approximating f_{img} by (3.17). Since this involves finding a 3D scene point $\mathbf{z}_i$ corresponding the image point $\tilde{\mathbf{u}}_i$, we refer to this approximation as *scene reconstruction.*

3.3 Scene Reconstruction and Error Fusion

In order to find a quadratic approximation (3.17) to f_{img}, we linearize $\boldsymbol{\pi}(\mathbf{y}_i)$ around $\tilde{\mathbf{y}}_i$ to give

$$\boldsymbol{\pi}(\mathbf{y}_i) \approx \boldsymbol{\pi}(\tilde{\mathbf{y}}_i) + \frac{\partial \boldsymbol{\pi}(\tilde{\mathbf{y}}_i)}{\partial \mathbf{y}_i}(\mathbf{y}_i - \tilde{\mathbf{y}}_i), \tag{3.20}$$

where the Jacobian of $\boldsymbol{\pi}$ at $\mathbf{y}_i = (x_i', y_i', z_i')^t$ is

$$\frac{\partial \boldsymbol{\pi}(\mathbf{y}_i)}{\partial \mathbf{y}_i} = \begin{pmatrix} \frac{1}{z_i'} & 0 & -\frac{x_i'}{(z_i')^2} \\ 0 & \frac{1}{z_i'} & -\frac{y_i'}{(z_i')^2}. \end{pmatrix} \tag{3.21}$$

The optimal $\mathbf{y}_i$ based on the above approximation should satisfy

$$\tilde{\mathbf{u}}_i - \boldsymbol{\pi}(\tilde{\mathbf{y}}_i) + \frac{\partial \boldsymbol{\pi}(\tilde{\mathbf{y}}_i)}{\partial \mathbf{y}_i}(\tilde{\mathbf{y}}_i - \mathbf{y}_i) = 0. \tag{3.22}$$

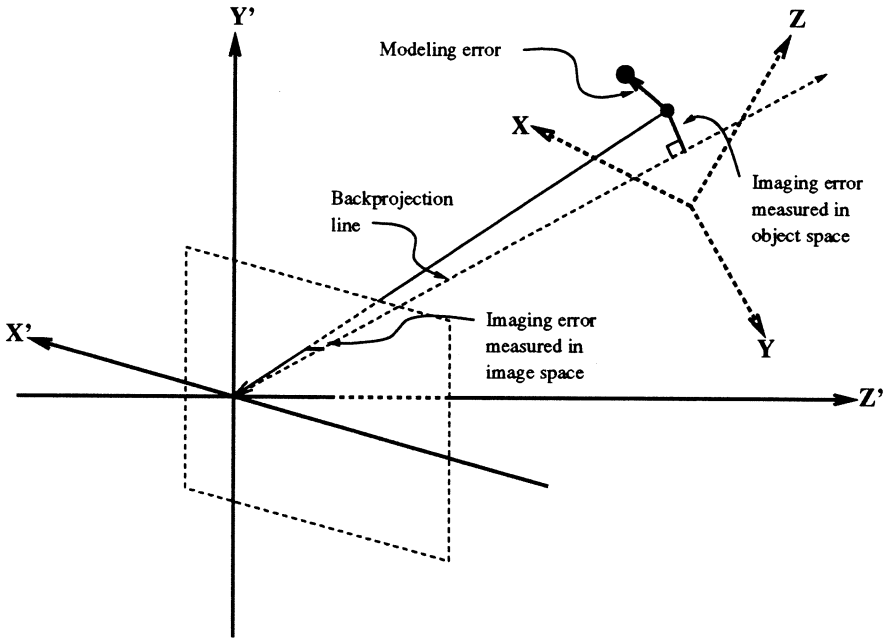

Figure 3.2: Imaging error measured in image space and object space.

Solving for $\mathbf{y}_i$ using (3.2), (3.3), and (3.14), and designating the solution as $\mathbf{z}_i = \mathbf{z}_i(R, \mathbf{t})$, we have

(3.23)
$$\mathbf{z}_i = \tilde{\mathbf{y}}_i + (\Sigma_{\tilde{\mathbf{y}}_i}^{-1} + \frac{\partial \boldsymbol{\pi}(\tilde{\mathbf{y}}_i)}{\partial \mathbf{y}_i}^t \Sigma_{\tilde{\mathbf{u}}_i}^{-1} \frac{\partial \boldsymbol{\pi}(\tilde{\mathbf{y}}_i)}{\partial \mathbf{y}_i})^{-1} \frac{\partial \boldsymbol{\pi}(\tilde{\mathbf{y}}_i)}{\partial \mathbf{y}_i}^t \Sigma_{\tilde{\mathbf{u}}_i}^{-1} (\tilde{\mathbf{u}}_i - \boldsymbol{\pi}(\tilde{\mathbf{y}}_i))$$

where $\mathbf{z}_i$ has the covariance matrix

(3.24)
$$\Sigma_{\mathbf{z}_i} = (\Sigma_{\tilde{\mathbf{y}}_i}^{-1} + \frac{\partial \boldsymbol{\pi}(\tilde{\mathbf{y}}_i)}{\partial \mathbf{y}_i}^t \Sigma_{\tilde{\mathbf{u}}_i}^{-1} \frac{\partial \boldsymbol{\pi}(\tilde{\mathbf{y}}_i)}{\partial \mathbf{y}_i})^{-1}$$

For the case that $\Sigma_{\tilde{\mathbf{y}}_i}$ and $\Sigma_{\tilde{\mathbf{u}}_i}$ are isometric, (3.23) can be simplified by orthogonally projecting $\tilde{\mathbf{y}}$ to the backprojection line of $\tilde{\mathbf{u}}_i$ as in [15]:

(3.25)
$$\mathbf{z}_i = A(\tilde{\mathbf{u}}_i)\tilde{\mathbf{y}}_i$$

where

$$A(\mathbf{u}_i) = \frac{\mathbf{v}_i\mathbf{v}_i^t}{\mathbf{v}_i^t\mathbf{v}_i} = \frac{1}{u_i^2+v_i^2+1}\begin{pmatrix} u_i^2 & u_iv_i & u_i \\ u_iv_i & v_i^2 & v_i \\ u_i & v_i & 1 \end{pmatrix}, \tag{3.26}$$

$$\mathbf{v}_i = (\mathbf{u}_i^t, 1)^t = (u_i, v_i, 1)^t. \tag{3.27}$$

Note that A is a projection operator. Here, $\mathbf{z}_i$ represents the closest point to $\tilde{\mathbf{y}}_i$ on the backprojection line of $\tilde{\mathbf{u}}_i$. The covariance matrix of $\mathbf{z}_i$ is given by

(3.28)

$$\begin{aligned}\mathbf{\Sigma}_{\mathbf{z}_i} &= \frac{\partial A(\tilde{\mathbf{u}}_i)\tilde{\mathbf{y}}_i}{\partial \mathbf{y}_i}\mathbf{\Sigma}_{\tilde{\mathbf{y}}_i}\frac{\partial A(\tilde{\mathbf{u}}_i)\tilde{\mathbf{y}}_i}{\partial \mathbf{y}_i}^t + \frac{\partial A(\tilde{\mathbf{u}}_i)\tilde{\mathbf{y}}_i}{\partial \mathbf{u}_i}\mathbf{\Sigma}_{\tilde{\mathbf{u}}_i}\frac{\partial A(\tilde{\mathbf{u}}_i)\tilde{\mathbf{y}}_i}{\partial \mathbf{u}_i}^t \\ &= A(\tilde{\mathbf{u}}_i)\mathbf{\Sigma}_{\tilde{\mathbf{y}}_i}A(\tilde{\mathbf{u}}_i)^t + \left(\frac{\partial A(\tilde{\mathbf{u}}_i)}{\partial u_i}\tilde{\mathbf{y}}_i \quad \frac{\partial A(\tilde{\mathbf{u}}_i)}{\partial v_i}\tilde{\mathbf{y}}_i\right)\mathbf{\Sigma}_{\tilde{\mathbf{u}}_i}\left(\frac{\partial A(\tilde{\mathbf{u}}_i)}{\partial u_i}\tilde{\mathbf{y}}_i \quad \frac{\partial A(\tilde{\mathbf{u}}_i)}{\partial v_i}\tilde{\mathbf{y}}_i\right)^t\end{aligned}$$

$A(\tilde{\mathbf{u}}_i)$, $\frac{\partial A(\tilde{\mathbf{u}}_i)}{\partial u_i}$ and $\frac{\partial A(\tilde{\mathbf{u}}_i)}{\partial v_i}$ can all be precomputed to improve the efficiency.

The difference $\mathbf{z}_i - \mathbf{y}_i$ can be considered as an error in object space *backprojected* from the imaging error $\tilde{\mathbf{u}}_i - \boldsymbol{\pi}(\mathbf{y}_i)$. If we use (3.25) to compute $\mathbf{z}_i$, it is the difference between a scene point $\mathbf{y}_i$ and its orthogonal projection on the backprojection line of the image point $\mathbf{u}_i$. By backprojecting imaging error to object space, the modeling error and the imaging error can be *fused* into covariance matrices (3.24) and (3.28) both measured in object space.

Note that all points that lie along the backprojection line from $\tilde{\mathbf{u}}_i$ would give small imaging error relative to $\tilde{\mathbf{u}}_i$. In this sense, trying to find points to fit the hypothesized scene points $\tilde{\mathbf{y}}_i$ can be considered as disambiguating the backprojection process.

3.4 Choice of Reconstruction Methods

Having f_{img} be approximated by a quadratic function in the form of (3.17), where $\mathbf{z}_i$ is obtained by (3.20) or (3.25), $\mathbf{y}^*(R, \mathbf{t})$ can be computed using (3.18). The objective function of pose and structure can now be converted to the objective function of

pose only. It is no more difficult than the original pose estimation problem where models are exact. Furthermore, the error measures of modeling and imaging can be propagated to that of the structure.

We should keep in mind that the pose and the structure are not completely decoupled as expected by using subspace decomposition, since $\mathbf{z}_i$ is approximated by linearization or orthogonal projection. The resulting nonlinear objective function $g(R, \mathbf{t}) = f(R, \mathbf{t}, \mathbf{y}^*(R, t))$ can only be solved by iterative methods which require another linearization or other approximation.

One major disadvantage of (3.20) is that it uses Jacobian of $\boldsymbol{\pi}$ which depends on $(R, \mathbf{t})$. If a first-order or higher minimization algorithm is applied, it will in effect compute the second-order derivatives with respect to $(R, \mathbf{t})$, which tend to be destabilizing when the initial guess for $(R, \mathbf{t})$ is bad.

On the other hand, when (3.25) is used, $\mathbf{z}_i$ depends linearly on $\tilde{\mathbf{y}}_i$ and hence $(R, \mathbf{t})$. When a first-order algorithm is used to minimize $g(R, \mathbf{t})$, only the first-order derivatives with with respect to $(R, \mathbf{t})$ are taken. Therefore, we will mainly use (3.25) in the rest of the dissertation.

Chapter 4

Alternating Subspace Minimization

In this chapter, we introduce a special subspace decomposition technique, called *alternating subspace minimization* to solve the pose and structure estimation problem introduced in Chapter 3. We also cover the issues on scaling the structure and initializing the iterative algorithm.

4.1 Alternating Subspace Minimization

The subspace decomposition method (3.15)(3.16) depends on the accuracy of the linearization around $\tilde{\mathbf{y}}_i = R\tilde{\mathbf{x}}_i + \mathbf{t}$ (3.23). To make the dependency of f_{img} on $(R, \mathbf{t})$ explicit, the corresponding linearized objective function in (3.17) can be written as

$$f_{img}(\mathbf{y}_i; R, \mathbf{t}) = \sum_i (\mathbf{y}_i - \mathbf{z}_i(R, \mathbf{t}))^t \mathbf{\Sigma}_{\mathbf{z}_i}^{-1} (\mathbf{y}_i - \mathbf{z}_i(R, \mathbf{t})), \tag{4.1}$$

and the best scene point estimate $\mathbf{y}^*$ is computed using (3.18).

By fixing $\mathbf{y}_i = \mathbf{y}_i^*$, $\tilde{f}_{mod}(R, \mathbf{t}, \mathbf{y}_i)$ is an objective function of an absolute orientation problem for $(R, \mathbf{t})$:

$$\tilde{f}_{mod}(R, \mathbf{t}; \mathbf{y}_i^*) = \sum_i (R\tilde{\mathbf{x}}_i + \mathbf{t} - \mathbf{y}_i^*)^t (\mathbf{\Sigma}_{\tilde{\mathbf{y}}_i} + \mathbf{\Sigma}_{\mathbf{y}_i^*})^{-1} (R\tilde{\mathbf{x}}_i + \mathbf{t} - \mathbf{y}_i^*). \tag{4.2}$$

The error that such an absolute orientation problem has to deal with is a 3 - D error

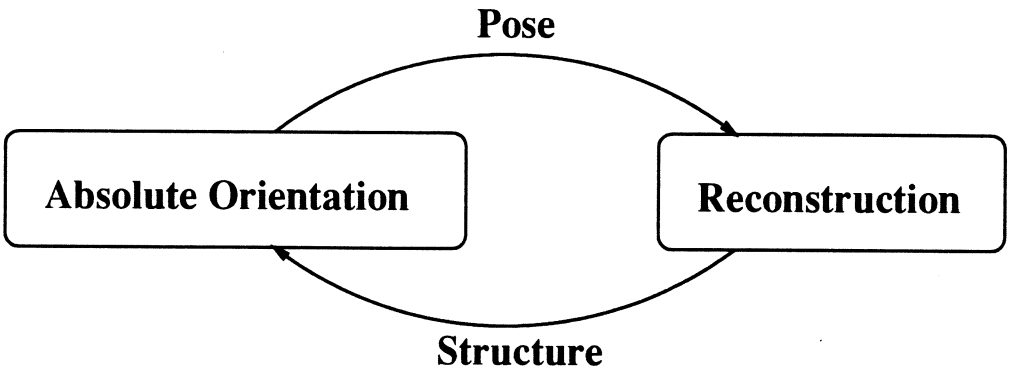

Figure 4.1: Alternating subspace minimization for pose estimation.

that is a vector sum of modeling error and structure error caused by imaging error (or imaging error backprojected into 3 - D object space, see Figure 3.2).

We can see that $\tilde{f}_{mod}(R, \mathbf{t}; \mathbf{y}_i)$ and $f_{img}(\mathbf{y}_i; R, \mathbf{t})$ are complementary. Both solve for the parameters that are to be fixed in their counterparts. A simple interaction between the pose and the structure is by alternatively minimizing $\tilde{f}_{mod}$ for the pose and f_{img} for the structure. This can be considered as simultaneously optimizing the pose and the structure by coordinate relaxation. Note that the computation of the structure is linear with complexity no more than a function evaluation, and the pose can be solved in closed-form.

Let $(R^{(k)}, \mathbf{t}^{(k)})$ and $\mathbf{y}_i^{(k)}$ be the kth estimate of the pose and the structure, respectively. The Alternating Subspace Minimization (ASM) method proposed above can be described as

$$(R^{(k)}, \mathbf{t}^{(k)}) = \arg\min_{R,\mathbf{t}} \tilde{f}_{mod}(R, \mathbf{t}; \mathbf{y}_i^{(k)}) \tag{4.3}$$

(absolute orientation phase)

$$\mathbf{y}_i^{(k+1)} = \arg\min_{\mathbf{y}_i} (f_{img}(\mathbf{y}_i; R^{(k)}, \mathbf{t}^{(k)}) + \tilde{f}_{mod}(R^{(k)}, \mathbf{t}^{(k)}; \mathbf{y}_i)) \tag{4.4}$$

(reconstruction phase)

or more elegantly as the clocked objective function [1]

(4.5)
$$\min_{R,\mathbf{t},\mathbf{y}_i} f(R,\mathbf{t},\mathbf{y}_i) = \min_{R,\mathbf{t}} \tilde{f}_{mod}(R,\mathbf{t};\mathbf{y}_i) \oplus \min_{\mathbf{y}_i}(f_{img}(\mathbf{y}_i;R,\mathbf{t}) + \tilde{f}_{mod}(R,\mathbf{t};\mathbf{y}_i))$$

where the covariance matrix $\mathbf{\Sigma}_{\mathbf{y}_i^{(k)}}$ of $\mathbf{y}_i^{(k)}$ is computed using (3.24) or (3.28) with $\tilde{\mathbf{y}}_i = R^{(k-1)}\tilde{\mathbf{x}}_i + \mathbf{t}^{(k-1)}$. Note that $\mathbf{\Sigma}_{\mathbf{y}_i^{(k)}}$ includes contribution from previous pose estimate $(R^{(k-1)},\mathbf{t}^{(k-1)})$ and the model point $\tilde{\mathbf{x}}_i$ through $\tilde{\mathbf{y}}_i = R^{(k-1)}\tilde{\mathbf{x}}_i + \mathbf{t}^{(k-1)}$, as well as from the corresponding image coordinate $\tilde{\mathbf{u}}_i$.

4.2 Solutions to the Absolute Orientation Phase

The ASM method solves the pose and structure estimation problem in two phases: the absolute orientation phase and the reconstruction phase. The reconstruction phase is simply a linear operation on each point. In the absolute orientation phase, an absolute orientation problem is solved based an estimate of the scene points to get a better pose estimate. Since such an absolute orientation step is one iteration among many others, it is not necessary to solve it perfectly. A important guideline for choosing an appropriate absolute orientation solution is that it must be noniterative.

The absolute orientation phase in the kth step of the ASM iterations involves minimizing the following objective function:

(4.6)
$$\tilde{f}_{mod}(R,\mathbf{t};\mathbf{y}_i^{(k)}) = \sum_i (R\tilde{\mathbf{x}}_i + \mathbf{t} - \mathbf{y}_i^{(k)})^t \mathbf{\Sigma}_i^{-1} (R\tilde{\mathbf{x}}_i + \mathbf{t} - \mathbf{y}_i^{(k)}),$$

where $\mathbf{\Sigma}_i = \mathbf{\Sigma}_{\tilde{\mathbf{y}}_i} + \mathbf{\Sigma}_{\mathbf{y}_i^{(k)}}$. Recall that $\mathbf{\Sigma}_{\tilde{\mathbf{y}}_i} = R\mathbf{\Sigma}_{\tilde{\mathbf{x}}_i}R^t$. The dependence of $\mathbf{\Sigma}_{\tilde{\mathbf{y}}_i}$ on the pose to be estimated make closed-form solutions impossible. Fortunately, we can find a good approximation by

(4.7)
$$\mathbf{\Sigma}_{\tilde{\mathbf{y}}_i} \approx R^{(k-1)}\mathbf{\Sigma}_{\tilde{\mathbf{x}}_i}(R^{(k-1)})^t,$$

[1] An objective function $f(x,y)$, to be optimized by coordinate descent on x and y, can be represented as a two-phase clocked objective function [51] $f(\bar{x},y) \oplus f(x,\bar{y})$, where x is *clamped* or fixed (denoted as $\bar{x}$) in the phase for coorinate descent on y, and vice versa.

where the unknown rotation R is replaced by previous estimate of the rotation $R^{(k-1)}$.

Even with the approximation in (4.7), minimizing (4.6) is still a difficult problem since it is a matrix-weighted least squares. The matrix weights prohibit utilization of the orthonormality of the rotation matrix to simplify the dependency of the objective function on R. An exact closed-form solution is not possible unless the orthonormality constraint on rotation is dropped, in which case the problem becomes a linear least squares problem. The solution comprises a 3-by-3 matrix for rotation and a 3 vector for translation. The extra degrees of freedom due to the lack of the orthonormality constraint may result in non-orthonormal matrix which can be improved by finding a rotation matrix that best fits the 3-by-3 matrix. The linear solution faces the same problem encountered by linear methods for pose estimation described in Section 2.5. The difference is that in our method, the linear solutions are computed progressively from previous results, and the final solutions are much more accurate and stable than one-shot linear solutions. For the matrix-weighted solution, the linear method of Weng *et. al.* [67] is used.

If the absolute orientation problem is presented as an equally-weighted or a scalar-weighted least squares, we can find closed-form solutions with the orthonormality constraint fully considered. This requires simplification of the matrix weights, or the covariance matrices. Assume that image error for each image coordinate is identical. Notice that the scene point $\tilde{\mathbf{y}}_i$ is estimated from the image coordinates $\tilde{\mathbf{u}}_i$. Since error in a scene point due to error in the corresponding image coordinate is roughly proportional to the depth of the scene point, the covariance matrix of $\mathbf{y}_i^{(k)}$ can be approximated as

$$\Sigma_{\mathbf{y}_i^{(k)}} \approx (d_i^{(k-1)})^2 a I, \tag{4.8}$$

where a is some constant, and $d_i^{(k-1)}$ is the depth of $\mathbf{y}_i^{(k-1)}$. If imaging error is dominant over modeling error, or the model is exact, the absolute orientation problem

can be formulated as a scalar-weighted least squares

$$\tilde{f}_{mod}(R, \mathbf{t}; \mathbf{y}_i^{(k)}) = \sum_i \frac{1}{(d_i^{(k)})^2} \|R\tilde{\mathbf{x}}_i + \mathbf{t} - \mathbf{y}_i^{(k)}\|^2. \tag{4.9}$$

Similar weighting schemes were used in [54,43]. In case that no prior knowledge of noise is available, we may want to use a plain equally-weighted least squares

$$\tilde{f}_{mod}(R, \mathbf{t}; \mathbf{y}_i^{(k)}) = \sum_i \|R\tilde{\mathbf{x}}_i + \mathbf{t} - \mathbf{y}_i^{(k)}\|^2. \tag{4.10}$$

For scalar-weighted cases, we follow the absolute orientation solution in [63] to solve for the rotation and the translation (see Appendix B).

4.3 Ambiguity in the Scale of the Structure

The centroid-coincidence theorem [67] states that if $(R^*, \mathbf{t}^*)$ minimizes (4.6), then the centroids of $\mathbf{y}_i = \mathbf{y}_i^{(k)}$ and $R^*\tilde{\mathbf{x}}_i + \mathbf{t}^*$ should coincide

$$\bar{\mathbf{y}} = R^*\bar{\mathbf{x}} + \mathbf{t}^*, \tag{4.11}$$

where $\bar{\mathbf{y}}$ and $\bar{\mathbf{x}}$ can be matrix-weighted centroids

$$\bar{\mathbf{y}} \stackrel{\text{def}}{=} (\sum_i \mathbf{\Sigma}_i^{-1})^{-1} \sum_i \mathbf{\Sigma}_i^{-1}\mathbf{y}_i, \quad \bar{\mathbf{x}} \stackrel{\text{def}}{=} (\sum_i \mathbf{\Sigma}_i^{-1})^{-1} \sum_i \mathbf{\Sigma}_i^{-1}\tilde{\mathbf{x}}_i. \tag{4.12}$$

Since the optimal translation $\mathbf{t}^*$ can be determined by the centroid of the structure and the rotation as:

$$\mathbf{t}^* = \bar{\mathbf{y}} - R^*\bar{\mathbf{x}}, \tag{4.13}$$

it is clear that the ASM method effectively optimizes over only R and $\mathbf{y}_i$ as:

$$R^{(k)} = \arg\min_R \tilde{f}_{mod}(R, \bar{\mathbf{y}}^{(k)} - R\bar{\mathbf{x}}; \mathbf{y}_i^{(k)}) \tag{4.14}$$

$$\mathbf{y}_i^{(k+1)} = \arg\min_{\mathbf{y}_i} f_{img}(\mathbf{y}_i; R^{(k)}, \bar{\mathbf{y}} - R^{(k)}\bar{\mathbf{x}}) + \tilde{f}_{mod}(R^{(k)}, \bar{\mathbf{y}} - R^{(k)}\bar{\mathbf{x}}; \mathbf{y}_i) \tag{4.15}$$

where

$$\mathbf{t}^{(k)} = \bar{\mathbf{y}}^{(k)} - R^{(k)}\bar{\mathbf{x}}. \tag{4.16}$$

Note that R^* does not change if $\mathbf{y}_i$ is replaced by $\mathbf{z}_i = s\mathbf{y}_i$, where s is some positive number, in the approximate matrix-weighted solution for R^* presented in [67]. On the other hand, f_{img} has no preference to any one in the family of $\{s\mathbf{y}_i | s > 0\}$, since $s\mathbf{y}_i$ and $\mathbf{y}_i$ project to the same image coordinate. In practice, f_{img} is linearized using previous pose estimate. Assume that $\mathbf{y}_i^{(k)}$ and $\mathbf{z}_i^{(k)}$ are two sequences of intermediate scene points computed using (3.25), and $\mathbf{z}_i^{(k)} = s\mathbf{y}_i^{(k)}$ at the kth iteration, then $\mathbf{y}_i^{(k+1)}$ and $\mathbf{z}_i^{(k+1)}$ are computed by

$$\mathbf{y}_i^{(k+1)} = A(\tilde{\mathbf{u}}_i)(R(\tilde{\mathbf{x}}_i - \bar{\mathbf{x}}) + \bar{\mathbf{y}}^{(k)}) \tag{4.17}$$

and

$$\begin{aligned} \mathbf{z}_i^{(k+1)} &= A(\tilde{\mathbf{u}}_i)(R(\tilde{\mathbf{x}}_i - \bar{\mathbf{x}}) + \bar{\mathbf{z}}^{(k)}) \\ &= \mathbf{y}_i^{(k+1)} + (s-1)A(\tilde{\mathbf{u}}_i)\bar{\mathbf{y}}^{(k)} \end{aligned} \tag{4.18}$$

This suggests that the impact of an overall scaling of $\mathbf{y}_i^{(k)}$ at a certain time step k propagates persistently to the next iteration. The effect can visualized as a shallow and narrow valley along the dimension of $\mathbf{y}_i$ at each R. Consequently, the algorithm can be slowly convergent as it descends along the shallow valley, and will very probably stop at some incorrectly scaled $\mathbf{y}_i$. This, in turn, results in incorrect translation $\mathbf{t}^*$ computed using (4.13).

This problem can be solved by optimizing an additional scale factor in the absolute orientation phase.

4.4 Optimizing Scale

By introducing a scale factor s for correcting the scale of $\mathbf{y}_i$, (4.6) becomes

$$\sum_i (R\tilde{\mathbf{x}}_i + \mathbf{t} - s\mathbf{y}_i)^t \boldsymbol{\Sigma}_i^{-1} (R\tilde{\mathbf{x}}_i + \mathbf{t} - s\mathbf{y}_i). \tag{4.19}$$

The optimal scale is given by [36]

$$\frac{\sum_i (\mathbf{y}_i - \bar{\mathbf{y}})^t \mathbf{\Sigma}_i^{-1} R(\tilde{\mathbf{x}}_i - \bar{\mathbf{x}})}{\sum_i (\mathbf{y}_i - \bar{\mathbf{y}})^t \mathbf{\Sigma}_i^{-1} (\mathbf{y}_i - \bar{\mathbf{y}})}. \tag{4.20}$$

Horn presented two least solutions to the scale in the cases for equally-weighted least squares [35,36]: The first one is

$$s_1 = \frac{\sum_i (\mathbf{y}_i - \bar{\mathbf{y}})^t R(\tilde{\mathbf{x}}_i - \bar{\mathbf{x}})}{\sum_i \|\mathbf{y}_i - \bar{\mathbf{y}}\|^2} \tag{4.21}$$

obtained by minimizing the following objective function

$$\sum_i \|R(\tilde{\mathbf{x}}_i - \bar{\mathbf{x}}) - s(\mathbf{y}_i - \bar{\mathbf{y}})\|^2, \tag{4.22}$$

and the second one is

$$s_2 = \sqrt{\frac{\sum_i \|\tilde{\mathbf{x}}_i - \bar{\mathbf{x}}\|^2}{\sum_i \|\mathbf{y}_i - \bar{\mathbf{y}}\|^2}}. \tag{4.23}$$

obtained by minimizing the following objective function

$$\sum_i \|\frac{1}{\sqrt{s}} R(\tilde{\mathbf{x}}_i - \bar{\mathbf{x}}) - \sqrt{s}(\mathbf{y}_i - \bar{\mathbf{y}})\|^2. \tag{4.24}$$

Both objective functions are equally-weighted least squares which imply that $\boldsymbol{\eta}_i$ are independently and identically distributed and are isotropic with $\mathbf{\Sigma}_i = \sigma^2 I, i = 1, \ldots, n$. The results can be easily extend to scalar-weighted cases.

The second solution is favored by Horn for two reasons. First, it is determined without the knowledge of the rotation R, and therefore the overall pose and scale solutions can be computed in a non-iterative manner. Second, it is *symmetrical* in the sense that if we switch the roles of $\tilde{\mathbf{x}}_i$ and $\mathbf{y}_i$, the resulting scale factor s_2' is equal to $1/s_2$. On the other hand, when the first approach is used instead, s_1 is usually not equal to $1/s_1'$. For this reason, we call (4.21) the *asymmetrical solution*, and (4.23) the *symmetrical solution*. In our pose and structure estimation framework where both the model points and the scene points are noisy, the symmetry property is especially desirable.

In the presence of noise, the scale of the perturbed scene points will usually larger than the unperturbed ones. The objective function with an additional scale variable (4.19) favors smaller scale factor, that is, the resulting scale factor tries to *shrink* the scene points. Under some simplified conditions, we will show that the deviation from the optimal solution is very sharply distributed around some small value.

The scale of a 3D point set can be defined as "Mean Squares of the Deviations from the Centroid" (MSDC). For example, the MSDCs of the model and the estimated structure are

$$\text{MSDC}(\tilde{\mathbf{x}}_i) \stackrel{\text{def}}{=} \frac{1}{n}\sum_i \|\tilde{\mathbf{x}}_i - \bar{\mathbf{x}}\|^2, \tag{4.25}$$

$$\text{MSDC}(\mathbf{y}_i) \stackrel{\text{def}}{=} \frac{1}{n}\sum_i \|\mathbf{y}_i - \bar{\mathbf{y}}\|^2, \tag{4.26}$$

(4.27)

respectively, where $\bar{\mathbf{x}}$ and $\bar{\mathbf{y}}$ are equally-weighted centroids. The symmetrical solution of scale factor can be rewritten as

$$s_2 = \sqrt{\frac{\text{MSDC}(\tilde{\mathbf{x}}_i)}{\text{MSDC}(\mathbf{y}_i)}}, \tag{4.28}$$

that is, it is a square root ratio of the scale of the model to the scale of the estimated structure.

4.5 Probabilistic Analysis of Deviation in Scale

In our framework of pose and structure estimation, $\boldsymbol{\eta}_i$ is not identically distributed and non-isotropic. In order to facilitate mathematical analysis, we consider a simpler case that $\boldsymbol{\eta}_i$ is non-isotropic but identical Gaussian, that is

$$s\tilde{\mathbf{y}}_i = R\tilde{\mathbf{x}}_i + \mathbf{t} + \boldsymbol{\eta}_i, \quad i = 1, \ldots, n, \ \boldsymbol{\eta}_i \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_i), \tag{4.29}$$

where $\boldsymbol{\Sigma}_i = \boldsymbol{\Sigma}, i = 1, \ldots, n$. Let the variances of the three components of noise, or the three diagonal elements of $\boldsymbol{\Sigma}$ be σ_x^2, σ_y^2, and σ_z^2, respectively.

Based on our assumption on noise vectors $\boldsymbol{\eta}_i$, we try to find how the symmetrical scale solution s_2 deviates from the true scale factor s, which can be optimally estimated using (4.20).

The true scale of the scene points can be represented by $s^2\,\mathrm{MSDC}(\tilde{\mathbf{y}}_i)$, which can be further decomposed as

$$
\begin{aligned}
(4.30)\qquad & s^2\,\mathrm{MSDC}(\tilde{\mathbf{y}}_i) \\
=&\frac{1}{n}\sum_i \|R\tilde{\mathbf{x}}_i + \mathbf{t} + \boldsymbol{\eta}_i - (R\bar{\mathbf{x}} + \mathbf{t} + \bar{\boldsymbol{\eta}})\|^2 \\
=&\,\mathrm{MSDC}(\tilde{\mathbf{x}}_i) + 2\frac{1}{n}\sum_i (\boldsymbol{\eta}_i - \bar{\boldsymbol{\eta}})^t R(\tilde{\mathbf{x}}_i - \bar{\mathbf{x}}) + \frac{1}{n}\sum_i \|\boldsymbol{\eta}_i - \bar{\boldsymbol{\eta}}\|^2,
\end{aligned}
$$

where $\frac{1}{n}\sum_i(\boldsymbol{\eta}_i - \bar{\boldsymbol{\eta}})^t R(\tilde{\mathbf{x}}_i - \bar{\mathbf{x}})$ is an estimate of the trace of the cross-covariance matrix of $\boldsymbol{\eta}_i$ and $R\tilde{\mathbf{x}}_i$. It tends to cancel out since $\boldsymbol{\eta}_i$ and $R\tilde{\mathbf{x}}_i$ are uncorrelated. We can safely assume it to be zero. The scale of the structure after corrections deviates from the scale of the model by an amount of

$$
(4.31)\qquad S^2 = \frac{1}{n}\sum_i \|\boldsymbol{\eta}_i - \bar{\boldsymbol{\eta}}\|^2.
$$

The properties of the distribution of S^2 can be found as follows.

The reference frame for the model can be rotated such that each of the three components of noise vectors is an independently distributed Gaussian, that is, a Gaussian with a diagonal covariance matrix. Assume that the Gaussian distribution is not singular. Since $\boldsymbol{\Sigma}$ is symmetric and positive definite, it can be decomposed as

$$
(4.32)\qquad \boldsymbol{\Sigma} = A\boldsymbol{\Sigma}'A^t,
$$

where A is orthonormal, and $\boldsymbol{\Sigma}' = \mathrm{diag}(\sigma_{x'}^2, \sigma_{y'}^2, \sigma_{z'}^2)$. Note that $\mathrm{tr}\,\boldsymbol{\Sigma}' = \sigma_{x'}^2 + \sigma_{y'}^2 + \sigma_{y'}^2 = \sigma_x^2 + \sigma_y^2 + \sigma_y^2 = \mathrm{tr}\,\boldsymbol{\Sigma}$.

The noise vector $\boldsymbol{\zeta}_i = (\zeta_{i1}, \zeta_{i2}, \zeta_{i3})^t$ defined in the new reference frame is related to $\boldsymbol{\eta}_i$ in the old reference frame by

$$
(4.33)\qquad \boldsymbol{\zeta}_i = A^{-1}\boldsymbol{\eta}_i, \quad \boldsymbol{\zeta}_i \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}'),
$$

or

$$\zeta_{i1} \sim \mathcal{N}(0, \sigma_{x'}^2), \quad \zeta_{i2} \sim \mathcal{N}(0, \sigma_{y'}^2), \quad \zeta_{i3} \sim \mathcal{N}(0, \sigma_{z'}^2). \tag{4.34}$$

Let $S_{x'}^2$, $S_{y'}^2$ and $S_{z'}^2$ be the sample variances of the three components of noise in the new reference frame, then we have

$$S^2 = \frac{1}{n}\sum_i \|\boldsymbol{\eta}_i - \bar{\boldsymbol{\eta}}\|^2 = \frac{1}{n}\sum_i \|\boldsymbol{\zeta}_i - \bar{\boldsymbol{\zeta}}\|^2 = S_{x'}^2 + S_{y'}^2 + S_{z'}^2 \tag{4.35}$$

It is well known that $\frac{nS_{x'}^2}{\sigma_{x'}^2}$, $\frac{nS_{y'}^2}{\sigma_{y'}^2}$, and $\frac{nS_{z'}^2}{\sigma_{z'}^2}$ are all $\chi^2(n-1)$, from which we have

$$E(S_{x'}^2) = \frac{n-1}{n}\sigma_{x'}^2, \boldsymbol{\sigma}^2(S_{x'}^2) = 2\frac{n-1}{n^2}\sigma_{x'}^4 \tag{4.36}$$

$$E(S_{y'}^2) = \frac{n-1}{n}\sigma_{y'}^2, \boldsymbol{\sigma}^2(S_{y'}^2) = 2\frac{n-1}{n^2}\sigma_{y'}^4 \tag{4.37}$$

$$E(S_{z'}^2) = \frac{n-1}{n}\sigma_{z'}^2, \boldsymbol{\sigma}^2(S_{z'}^2) = 2\frac{n-1}{n^2}\sigma_{z'}^4. \tag{4.38}$$

The p.d.f. of each one of them is

$$p(x) = \frac{1}{\Gamma(\frac{n-1}{2})}\frac{n}{2\sigma^2}^{\frac{n-1}{2}} x^{\frac{n-3}{2}} e^{-\frac{n}{2\sigma^2}x}, \tag{4.39}$$

which has the mode at $x = \frac{n-3}{n}\sigma^2$, where x represents one of $S_{x'}^2$, $S_{y'}^2$ and $S_{z'}^2$, and σ^2 is one of $\sigma_{x'}^2$, $\sigma_{y'}^2$ and $\sigma_{z'}^2$.

Figure 4.2 shows plots of $p(x)$ for $n = 10$ and 5 different σ from 0.1 to 0.5 in 0.1 step. We can see that $p(x)$ becomes much sharper and much closer to zero as σ decreases.

The expected value of S^2 is

$$E(S^2) = \frac{n-1}{n}\operatorname{tr}\boldsymbol{\Sigma}, \tag{4.40}$$

and the variance is

$$\boldsymbol{\sigma}^2(S^2) = 2\frac{n-1}{n^2}(\sigma_{x'}^4 + \sigma_{y'}^4 + \sigma_{z'}^4). \tag{4.41}$$

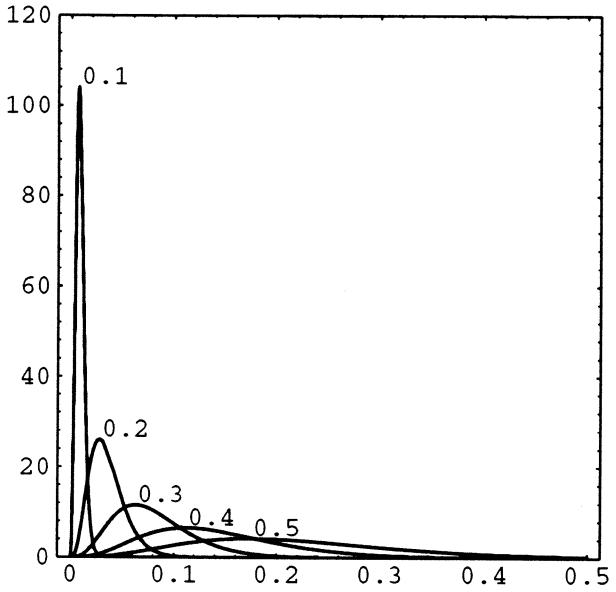

Figure 4.2: Plots of $p(x)$ for $n = 10$ and $\sigma = 0.1, 0.2, 0.3, 0.4$, and 0.5.

The maximum likelihood estimate of S^2 is

$$\frac{n-3}{n}(\sigma_{x'}^2 + \sigma_{y'}^2 + \sigma_{z'}^2) = \frac{n-3}{n}\operatorname{tr}\boldsymbol{\Sigma}, \tag{4.42}$$

and consequently the maximum likelihood estimate of s is

$$\sqrt{(s_2)^2 + \frac{(n-3)\operatorname{tr}\boldsymbol{\Sigma}}{\sum_i \|\tilde{\mathbf{y}}_i - \bar{\mathbf{y}}\|^2}}, \tag{4.43}$$

where s_2 is the symmetrical solution to the scale factor computed using (4.28).

As expected, s_2 is generally smaller than the true scale factor s with the presence of noise. The mean and the mode of the deviation S^2 is about the same size as the variance of the error in structure while the variance is about the sum of the squares of the structure variance divided by the number of points. S^2 becomes much more sharply distributed around a value much closer to zero as the variance of structure decreases. For this reason, we can use s_2 as the correcting scale factor in the absolute orientation phase.

4.6 Initialization: A Weak-Perspective Approximation

Since the ASM method is iterative, it requires an initial guess. Unlike other methods, the initial guess for the ASM method is presented in the form of an initial set of scene points, not an initial pose. While it takes some specific prior knowledge to find a good initial guess for the pose, a good initial guess for scene points can just be the image vectors themselves, which form a plane parallel to the image plane. The scaling step described in Section 4.4 can automatically normalize the scale. This is equivalent to assuming the scene points to have the same depth initially. We show that the initial pose found by the initialization scheme described above is a pose solution under the weak-perspective projection model.

Previous work using weak-perspective has mostly focused on analytical methods using a minimal number of feature correspondences (e.g. 3 points). In this context, the problem is usually interpreted as purely algebraic or geometrical. Here, we formulate the problem as a least squares problem. Under the weak-perspective projection imaging model, we have the following relation for each model point $\tilde{\mathbf{x}}_i$

$$s\tilde{u}_i = \mathbf{r}_1^t \tilde{\mathbf{x}}_i + t_1 \tag{4.44}$$

$$s\tilde{v}_i = \mathbf{r}_2^t \tilde{\mathbf{x}}_i + t_2, \tag{4.45}$$

where s is a positive scale. Weak-perspective projection is valid when the depths of all 3D scene points are roughly the same. Let's call this depth the *principle depth.* If $(\tilde{u}_i, \tilde{v}_i)$ is on the normalized image plane ($z' = 1$) as defined in Chapter 2, then it is clear that s is the principle depth. Determining the scale s under weak-perspective projection can be interpreted as choosing an appropriate principle depth. In analytical methods, the principle depth is chosen as the depth of one of a minimal number of model points. Using the least squares approach, the principle depth can be chosen as

the one that minimizes its deviation from the depths of the scene points

$$\sum_i (\mathbf{r}_3^t \tilde{\mathbf{x}}_i + t_3 - s)^2. \tag{4.46}$$

The whole imaging process can be visualized in two stages: in the first stage, all scene points are orthogonally projected to a plane $z = s$ (called the *principle plane*) parallel to the image plane, and in the second stage, all the planar points on the principle plane are perspectively projected to the normalized image plane.

We also need to minimize the square of the image error

$$\sum_i \left[(\mathbf{r}_1^t \tilde{\mathbf{x}}_i + t_1 - s\tilde{u}_i)^2 + (\mathbf{r}_2^t \tilde{\mathbf{x}}_i + t_2 - s\tilde{v}_i)^2 \right]. \tag{4.47}$$

Combining (4.46) and (4.47), and weighting them equally, we have the following least squares objective function

$$\sum_i \| R\tilde{\mathbf{x}}_i + \mathbf{t} - s\tilde{\mathbf{v}}_i \|^2. \tag{4.48}$$

This is the objective function (4.10) with the additional constraint that all scene points have the same depth. The scale factor s can be found using the symmetrical solution (4.23) which does not require knowledge of the pose.

Chapter 5

Performance Evaluation

In this chapter, the theory and the algorithm, as well as the software implementation are evaluated using different test strategies.

5.1 Statistical Correctness and Optimality

After the least-squares objective function is formulated using the minimum variance principle described in Section 3.1, the original estimation problem becomes an optimization problem. Since the problem is nonlinear, we need approximations such as linearization or coordinate relaxation to solve the problem iteratively.

It may seem that we only need to be concerned about how to reach the optimal or at least a locally optimal solution. However, in addition to the approximations employed to develop iterative algorithms, there are at least two other approximations:

- using Jacobians to calculate covariance matrices (3.7).
- using either linearization (3.20) or orthogonal projection (3.25) for backprojection.

The former is a statistical simplification while the latter is a geometrical approximation. Both are employed to derive the objective function itself. We not only want the

compuitational method to be able find good solutions, but also require the objective function to be statistically correct.

We validate and evaluate both our objective function and the associated optimization method on artificially generated input data with known pose solution as well as controlled statistical characteristics. Two kinds of evaluation methods are employed:

- Testing our algorithm on a random population in a specific problem setting to see whether the output of the algorithm is distributed as predicted by the theory. The focus is on the statistical correctness of our 3D-2D error fusion scheme.

- Comparing our algorithm to others in a large population of different problem settings to see the relative optimality of the pose solutions. The focus is on the performance of the ASM method.

Data Generation Protocol The protocol for generating input data used throughout this chapter is introduced as follows:

The data is generated according to the following control parameters: number of points N, signal-to-noise ratios in 3D modeling (SNR_{mod}) and 2D imaging (SNR_{img}), and percentage of outliers (PO).

A set of N 3D model points $\mathbf{x}_i = (x_i, y_i, z_i)^t$ are generated uniformly within a box defined by $x_i,\ y_i,\ z_i \in [-5, 5]$. Gaussian noise $\mathcal{N}(0, \sigma)$ is added to the three components of $\mathbf{x}_i$ to generate the perturbed model points $\tilde{\mathbf{x}}_i$. The variance σ is related to SNR_{mod} by $\text{SNR}_{mod} = -20\log(\sigma/10)$ dB. Accordingly, the covariance matrix of $\tilde{\mathbf{x}}_i$ is $\sigma I_{3\text{x}3}$.

In order to generate a 3D rotation R, a unit quaternion is uniformly selected from a unit 4-sphere. The resulting distribution of 3D rotations is also uniform [14]. For translation $\mathbf{t}$, t_1 and t_2 are uniformly selected from $[5, 15]$, and t_3 from $[20, 50]$. The set of 3D scene points $\tilde{\mathbf{y}}_i = R\tilde{\mathbf{x}}_i + \mathbf{t}$ are generated using the selected R and $\mathbf{t}$.

A fraction (= PO) of the 3D points are selected as outliers. Each of these points $\tilde{\mathbf{y}}_i = (\tilde{x}_i', \tilde{y}_i', \tilde{z}_i')^t$ is replaced by another 3D point $(x_i^*, y_i^*, z_i^*)^t, z_i^* = z_i'$, where x_i^* and y_i^* are uniformly distributed within $[t_1 - 5, t_1 + 5]$ and $[t_2 - 5, t_2 + 5]$, respectively.

The 3D model points $\mathbf{x}_i$ are projected onto the normalized image plane ($z = 1$) to produce image points $\mathbf{u}_i$. Gaussian noise $\mathcal{N}(0, \sigma')$ is added to both coordinates of the image points to generate the perturbed image points $\tilde{\mathbf{u}}_i$, where the variance, σ', is related to SNR_{img} by $\text{SNR}_{img} = -20\log(\sigma'/0.3)$ dB (the image size is roughly $10/35 \approx 0.3$). Accordingly, the covariance matrix of $\tilde{\mathbf{u}}_i$ is $\sigma' I_{2\text{x}2}$.

5.2 Statistical Validation

Two hypothesis tests are used:

T1 Variance test with known mean. The purpose is to measure only the correctness of the covariance propergation scheme introduced in Chapter 3.

T2 Mean-and-variance test. The purpose is to measure the correctness of both the output, appearing as the mean of the output distribution, and its uncertainty measure (the covariance)

Since pose and structure are coupled, we can simply test on the estimated structure $\boldsymbol{y} = \mathbf{z}_i$ computed using (3.23) or (3.25) (only the latter is tested in our experiments). The purpose of our statistical validation tests is to verify whether $\boldsymbol{y}$ is distributed as predicted by (3.25) and (3.28).

In the following discussion we will use the following definitions of the sample mean $\bar{\boldsymbol{y}}$ and the sample covariance S:

$$\bar{\boldsymbol{y}} = \frac{1}{n}\sum_{k=1}^{n} \boldsymbol{y}_k$$

and

$$S = \frac{1}{n-1}\sum_{k=1}^{n} (\boldsymbol{y}_k - \bar{\boldsymbol{y}})(\boldsymbol{y}_k - \bar{\boldsymbol{y}})^t,$$

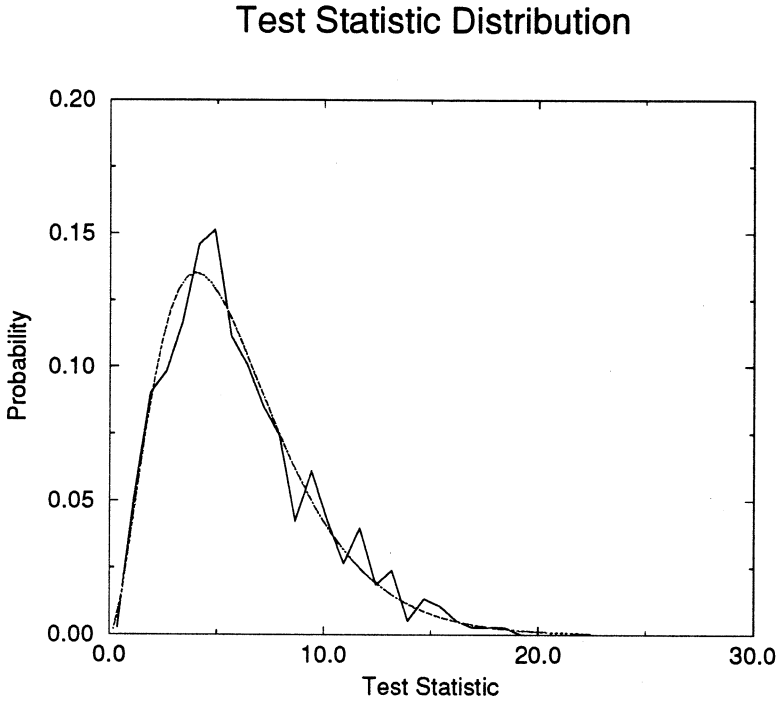

Figure 5.1: Empirical and theoretical distributions of **T1** test statistic. Here, the theoretical null distribution, plotted in dashed line, is χ_6. The histogram of the empirical null distribution, plotted in solid line, is obtained by computing the T statistic using 500 trials.

where the sample size is n and the kth sample of $\boldsymbol{y}$ is written as $\boldsymbol{y}_k$, which is computed from $\tilde{\mathbf{x}}_i$ using (3.25) (see Section 3.3).

5.2.1 Variance test with known mean

The variance test with known mean **T1** can be formally described as:

$$H_0 : \quad \boldsymbol{\Sigma} = \boldsymbol{\Sigma}_0, \quad \boldsymbol{\mu} = \boldsymbol{\mu}_1, \tag{5.1}$$

$$H_A : \quad \boldsymbol{\Sigma} \neq \boldsymbol{\Sigma}_0. \tag{5.2}$$

The known mean $\boldsymbol{\mu}_1$ is computed by

$$\boldsymbol{\mu}_1 = R\mathbf{x}_i + \mathbf{t}, \tag{5.3}$$

where $(R, \mathbf{t})$ is the true pose and $\mathbf{x}_i$ is the unperturbed model point.

Let

$$C = \sum_{k=1}^{n} (\boldsymbol{y}_k - \boldsymbol{\mu}_1)(\boldsymbol{y}_k - \boldsymbol{\mu}_1)^t = (n-1)S + (\bar{\boldsymbol{y}} - \boldsymbol{\mu}_1)(\bar{\boldsymbol{y}} - \boldsymbol{\mu}_1)^t \ . \tag{5.4}$$

The likelihood ratio criterion for testing H_0 is

$$\lambda = (e/n)^{3n/2} |C\boldsymbol{\Sigma}_0^{-1}|^{n/2} \exp(-\operatorname{tr}(C\boldsymbol{\Sigma}_0^{-1})/2) \ , \tag{5.5}$$

and the test statistic is

$$T = -2 \log \lambda, \tag{5.6}$$

which is χ^2-distributed with 6 degree of freedom:

$$T \sim \chi^2_6. \tag{5.7}$$

The result is a specialization to more general result in [3] page 249, 434, 436 with the dimension of the random variable set to 3.

5.2.2 Mean-and-variance test

The mean-and-variance hypothesis test **T2** can be formally described as:

$$H_0 : \ \boldsymbol{\Sigma} = \boldsymbol{\Sigma}_0 \ \text{and} \ \boldsymbol{\mu} = \boldsymbol{\mu}_0, \tag{5.8}$$

$$H_A : \ \boldsymbol{\Sigma} \neq \boldsymbol{\Sigma}_0 \ \text{and} \ \boldsymbol{\mu} \neq \boldsymbol{\mu}_0. \tag{5.9}$$

The hypothesized mean and covariance, $\boldsymbol{\mu}_0$ and $\boldsymbol{\Sigma}_0$, are computed using (3.25) and (3.28) using the pose solution computed for the first data sample.

Let

$$B = (n-1)S. \tag{5.10}$$

The likelihood ratio criterion for testing H_0 is

$$\lambda = (e/n)^{3n/2} |B\boldsymbol{\Sigma}_0^{-1}|^{n/2} \exp\left(-[\operatorname{tr}(B\boldsymbol{\Sigma}_0^{-1}) + n(\bar{\boldsymbol{y}} - \boldsymbol{\mu}_0)^t \boldsymbol{\Sigma}_0^{-1} (\bar{\boldsymbol{y}} - \boldsymbol{\mu}_0)]/2\right), \tag{5.11}$$

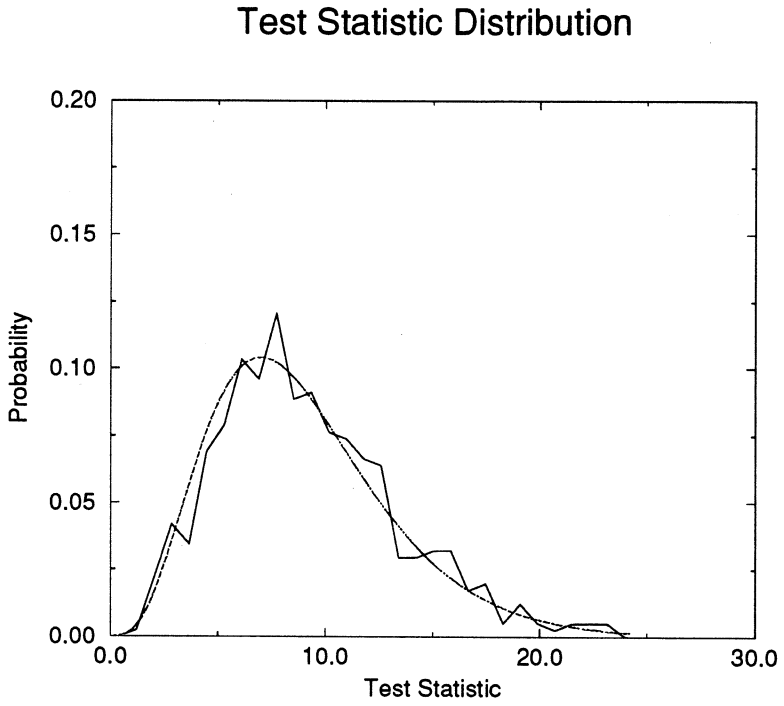

Figure 5.2: Empirical and theoretical distributions of **T2** test statistic. Here, the theoretical null distribution, plotted in dashed line, is χ_9. The histogram of the empirical null distribution, plotted in solid line, is obtained by computing the T statistic using 500 trials.

and the test statistic is

$$T = -2 \log \lambda \tag{5.12}$$

which is χ^2-distributed with 9 degree-of-freedom

$$T \sim \chi^2_9. \tag{5.13}$$

The result is a specialization to more general result in [3] page 442 with the dimension of the random variable set to 3.

5.2.3 Results and discussions

More general treatment of mean and variance tests is detailed in [3] and summarized in [42].

Each hypothesis test is repeated in five hundred trials in order to compute an empirical distribution for the test statistic. In each trial, one hundred samples are generated to compute the test statistic. The control parameters for generating samples are: $N = 20$, $\text{SNR}_{mod} = 60$, $\text{SNR}_{img} = 80$, and $\text{PO} = 0$. 95 percent of the trials are passed with 0.05 level of significance.

The empirical distributions are binned into thirty intervals. The resulting histograms for the 7th scene points are shown in Figure 5.1 and Figure 5.2. As we can see, the empirical distributions fit the corresponding χ^2 distributions very well. Similar results can be seen for other scene points.

5.3 Performance Comparison

In the following section, we will investigate the properties of the ASM method in comparison to other techniques based on experimental results. For this purpose, we design a set of standard comparison tests on synthetic data with varying noise, percentages of outliers and numbers of model points.

5.3.1 Standard comparison experiments

The following four standard experiments were conducted on the generated input data:

C1 Set $N = 20$, PO $= 0$, $\text{SNR}_{mod} = 70$ dB. Record the log errors of rotation and translation against SNR_{img} (30 dB-70 dB in 10 dB step). The purpose is to measure how well the tested methods resist imaging error.

C2 Set $N = 20$, $\text{SNR}_{img} = 60$ dB, $\text{SNR}_{mod} = 70$ dB. Record the log errors of rotation and translation against PO (5 %-25 % in 5 % step). The purpose is to see how well the tested methods tolerate outliers.

C3 Set PO $= 0$, $\text{SNR}_{img} = 50$ dB, $\text{SNR}_{mod} = 70$ dB. Record the log errors of rotation and translation against N (10 to 50 by step of 10). The purpose is to investigate how the performance can be improved by increasing the number of model points.

C4 Set $N = 20$, PO $= 0$, $\text{SNR}_{img} = 70$ dB. Record the log errors of rotation and translation against SNR_{mod} (30 dB-70 dB in 10 dB step). The purpose is to measure how well the tested methods resist modeling error.

To assess the performance of the methods, we measure the mean errors in rotation and translation of 1,000 trials for each setting of the control parameters. All the comparisons were conducted on a Silicon Graphics IRIS Indigo with a MIPS R4400 processor.

5.3.2 Error measures for 3D rotations

The error measure for translation is straightforward since a 3-vector has a natural Euclidean norm. The error measure for rotation depends on its representation. When represented by Euler angles, there is no natural norm for 3D rotation. A commonly used error measure is the average of the absolute errors for each Euler angle. When

the rotation is represented by a unit quaternion, the rotation error can be represented by quaternion error. The difference between any two unit quaternions $\boldsymbol{q}, \boldsymbol{q}'$ is

$$\|\boldsymbol{q} - \boldsymbol{q}'\|^2 = 2(1 - \boldsymbol{q}^t\boldsymbol{q}') \tag{5.14}$$

using the law of cosines. Note that every unit quaternion $\boldsymbol{q}$, and its negation $-\boldsymbol{q}$ represent the same 3D rotation. Therefore, the error between $\boldsymbol{q}$ and $\boldsymbol{q}'$ can be uniquely defined by

$$1 - |\boldsymbol{q}^t\boldsymbol{q}'| \in [0, 1]. \tag{5.15}$$

An important advantage of this error measure is that it is independent of coordinate system. We will use average Euler angle error most of time because it is most intuitive, and will use quaternion error when appropriate.

5.3.3 Results and Discussions

Importance of scale optimization

Although the scaling step in the absolute orientation phase is only a reasonable approximation, it significantly improves the convergence rate and the accuracy of the solution. A similar approach, referred to as the *initial approximation* algorithm, was proposed by Haralick et al. [15]. It uses an equally-weighted least squares solution in the absolute orientation phase. This algorithm converges very slowly as reported by the authors. The major difference of their work from ours is that the scale optimization is not utilized to help pull out the correct pose solution. We compare the ASM method with and without the scaling step. Both methods are initialized using the weak-perspective approximation.

Figure 5.3 shows the average numbers of iterations of both methods with error bars of the population of individual estimation errors. It is clear that ASM without scaling coverges slowly as discussed in Section 4.3. Figures 5.5, 5.7, and 5.8 show that ASM without scaling does sometimes produce better rotation results while the results

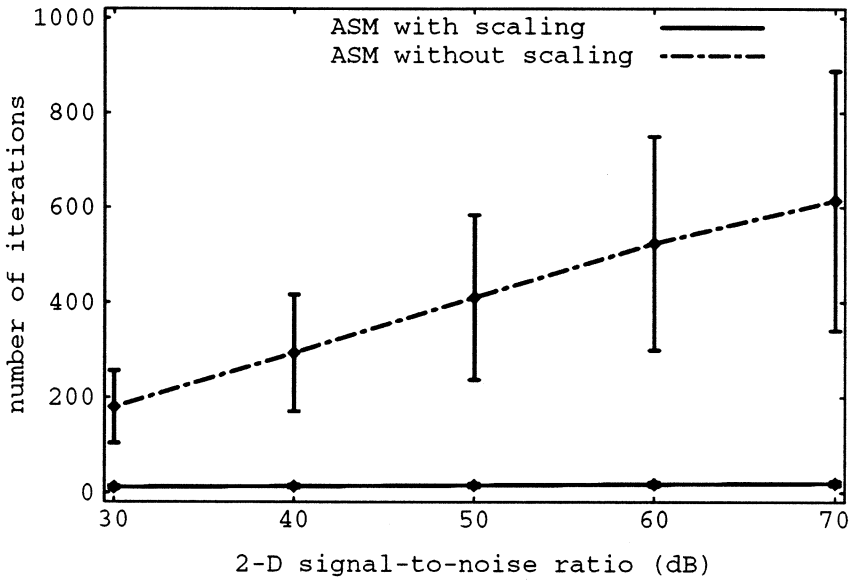

Figure 5.3: Comparing average numbers of iterations used by ASM with and without scaling for Experiment **C1**. Each point in the plot represents 1,000 trials.

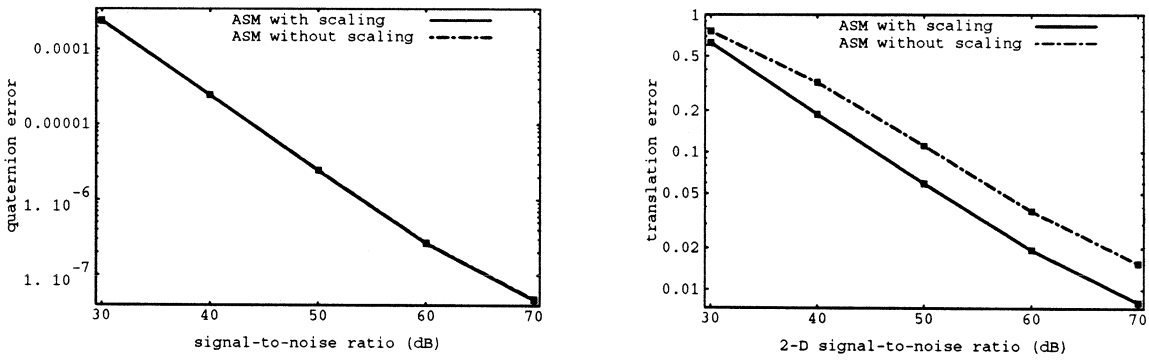

Figure 5.4: Result of Experiment **C1** for comparing ASM with and without scaling. The rotation error is represented by log quaternion error to emphasize the log-linearity. Each point in the plot represents 1,000 trials.

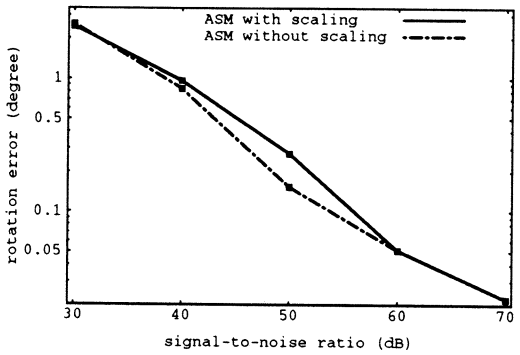

Figure 5.5: Result of Experiment **C1** for comparing ASM with and without scaling. Only log rotation error (average Euler angle error) is shown. Each point in the plot represents 1,000 trials.

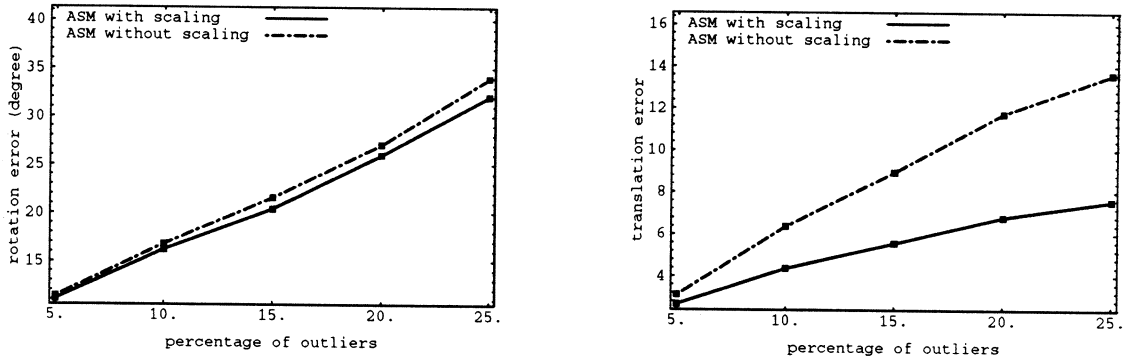

Figure 5.6: Result of Experiment **C2** for comparing ASM with and without scaling. Each point in the plot represents 1,000 trials.

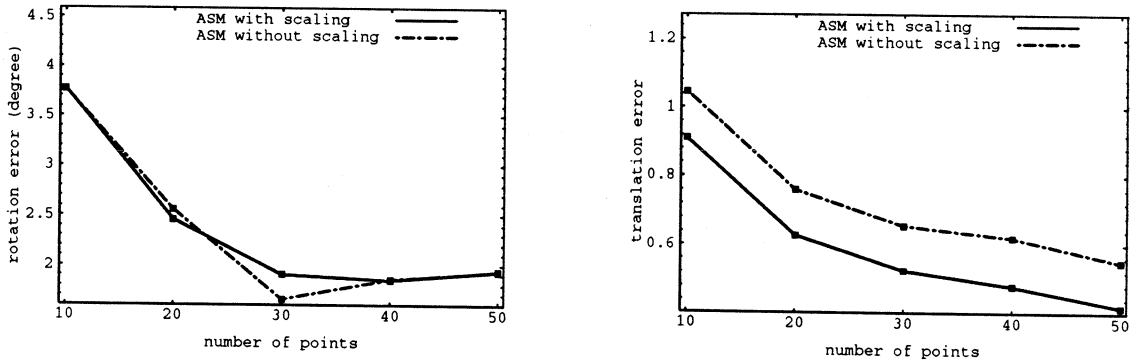

Figure 5.7: Result of Experiment **C3** for comparing ASM with and without scaling. Each point in the plot represents 1,000 trials.

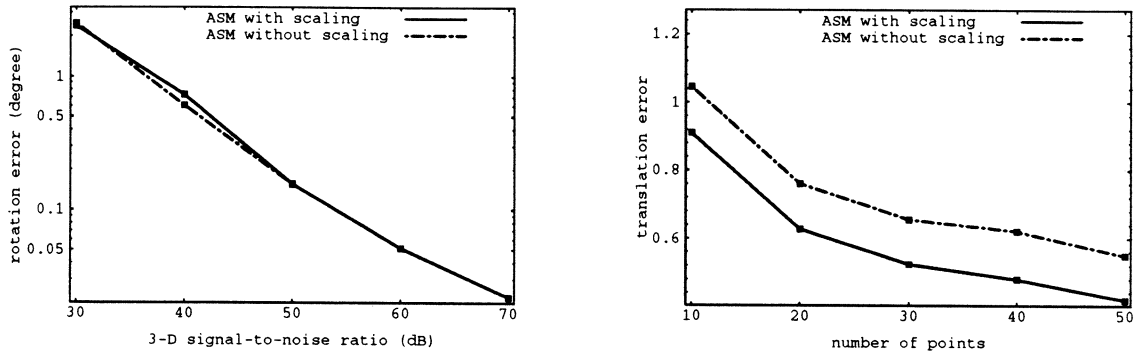

Figure 5.8: Result of Experiment **C4** for comparing ASM with and without scaling. Error is in log scale. Each point in the plot represents 1,000 trials.

for the translation are consistently and significantly worse than ASM with scaling. These results together confirm the fact that scale optimization effectively removes the ambiguity in the scale of the scene points, and consequently improves dramatically the rate of convergence as well as the accuracy of the translation.

Figure 5.9 compares experimentally the uncorrected scale factor computed using (4.28) and the corrected one computed using (4.43) against SNR (5 dB-70 dB in 5 dB step). The resulting points are disturbed by isotropic Gaussian noise with σ related to SNR by $\sigma = 10 * 10^{\frac{-SNR}{20}}$. Ideally, the computed scale factor should be equal to one. In practice, the scale of the scene points expands due to noise, and make the scale factor smaller than one. The asymmetrical solution causes significantly more shrinkage than the symmetrical solution. After the correction by (4.43), the scale is consistently close to one.

Figure 5.11 and Figure 5.10 demonstrate a typical run of ASM with and without scaling. The scene points are initialized with the same depth. We can see that the scene points computed by both methods have very similar orientation as early as in the second iteration. The scaling step pushes the scene points to the correct scale at the 5th iteration. On the other hand, the scene points without scaling move very slowly towards the correct locations although their orientation is almost correct.

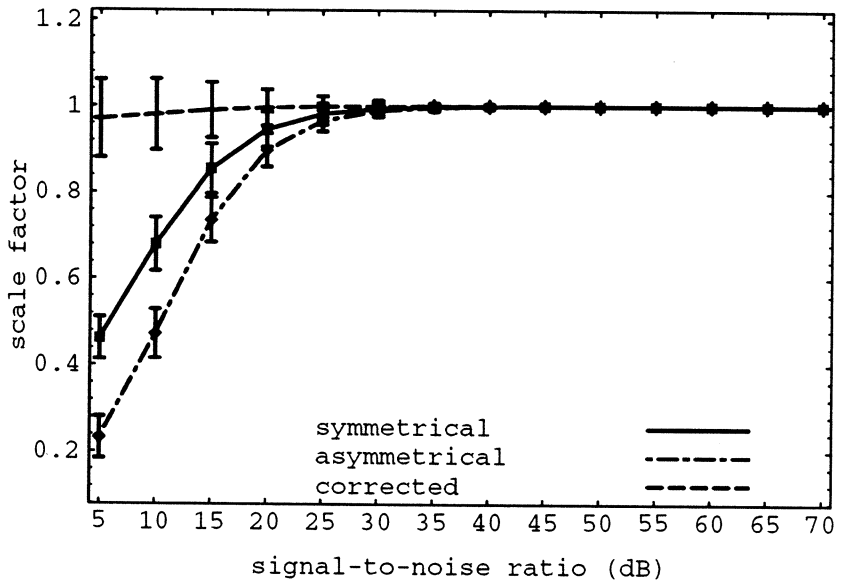

Figure 5.9: Corrected and uncorrected scale factors computed for different SNR.

Linear methods and ASM

As mentioned in Section 4.2, when a matrix-wighted least squares solution is required, the absolute orientation phase can only be solved approximately by ignoring the orthonormality constraint on the rotation matrix. This raises a question: since it uses a linear solution in the inner loop, why not directly uses the linear methods for pose estimation?

We will now compare ASM and two linear methods (PTM and RAC) described in Section 2.5. According to Section 2.5, the two linear methods PTM and RAC can only consider 2D imaging error. By solving only 3 to 6 iterations linearly along with orthonormalization, we will see that ASM clearly outerperforms linear pose estimation methods in the comparison tests, as shown in Figures 5.12, 5.14 and 5.15, including the cases when 2D imaging error is dominant (Experiment **C1**, Figure 5.12). In the presence of outliers, linear methods broke down, while the ASM method still produces reasonable results, as indicated by Figure 5.13.

This result will show that even with imperfectly reconstructed structure, the ab-

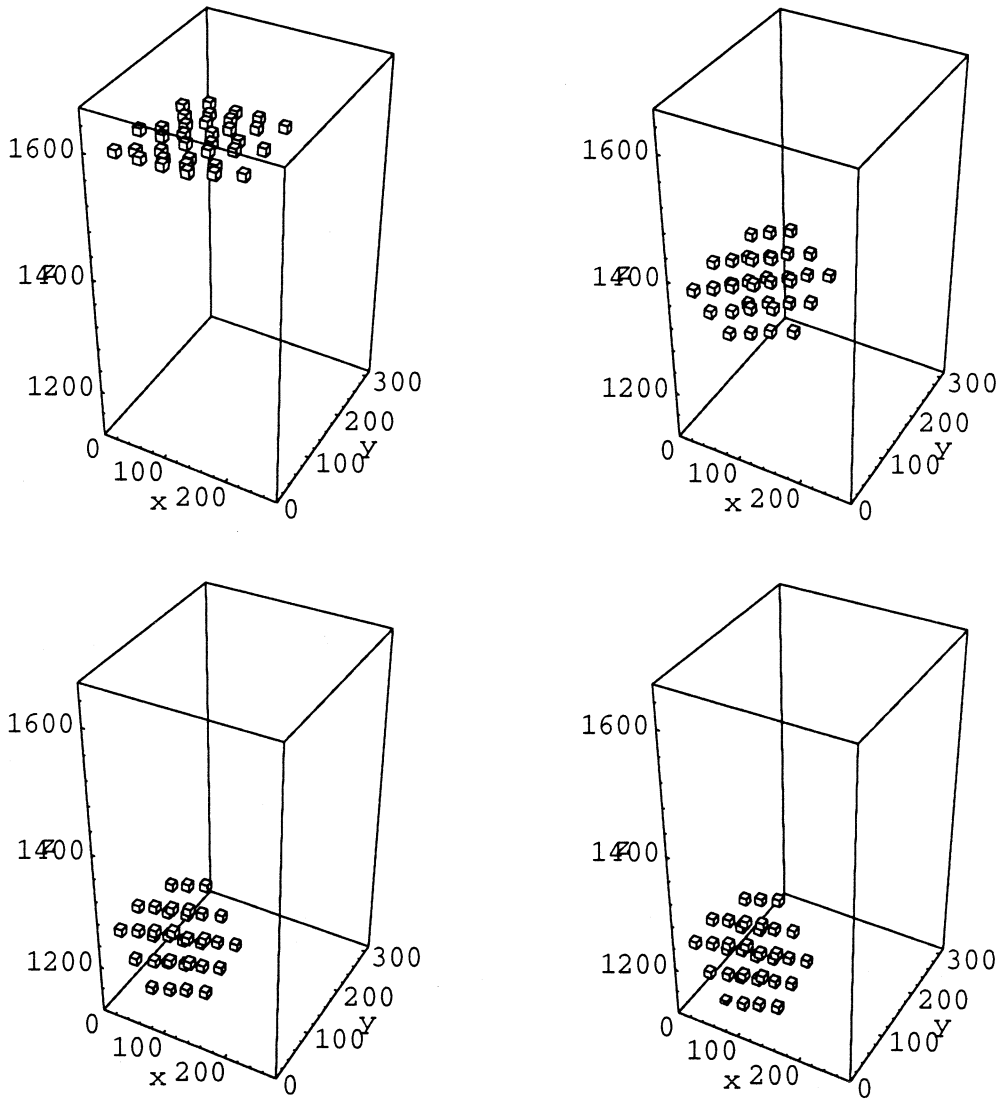

Figure 5.10: The intermediate scene points at iteration 1, 2, 5, and 12 of a typical run of ASM with scaling.

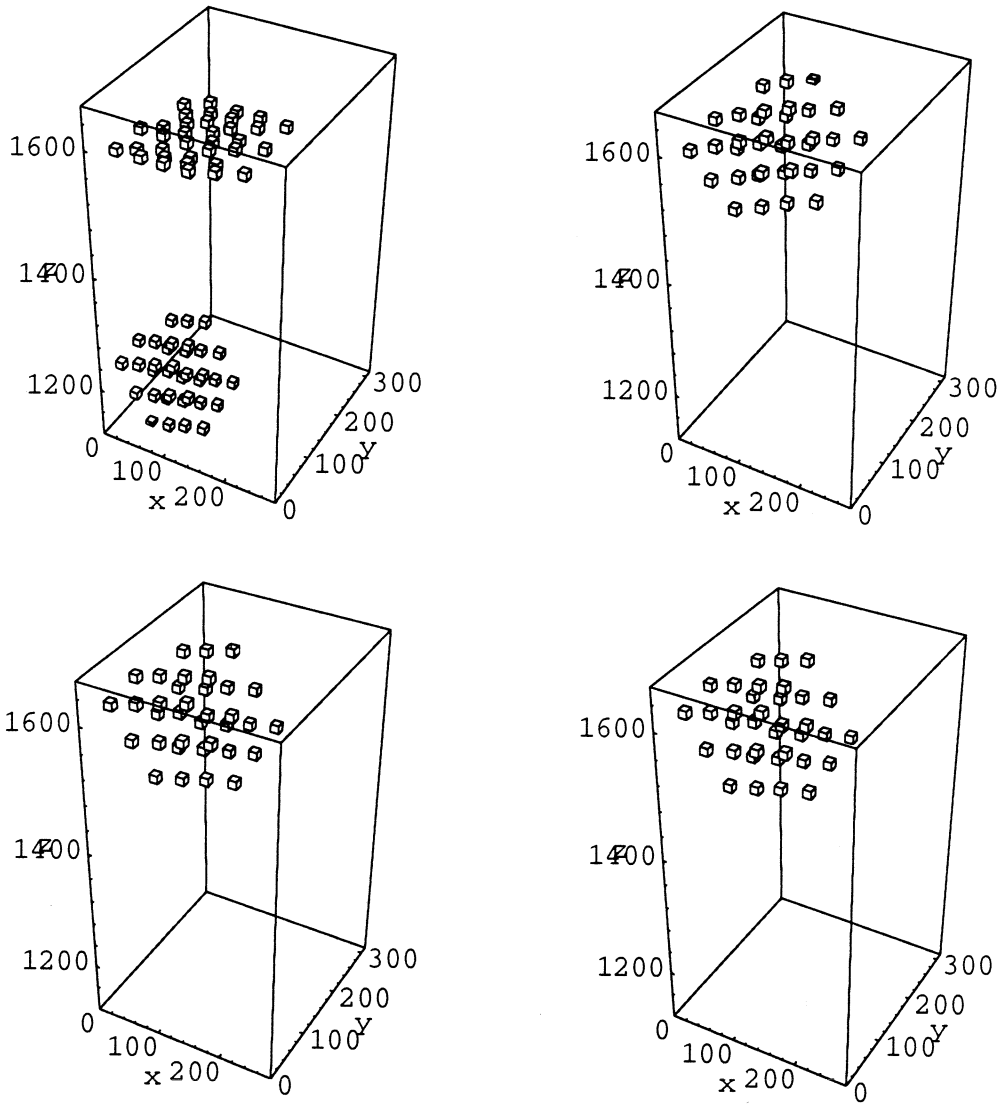

Figure 5.11: The intermediate scene points at iteration 1, 2, 5, and 12 of a typical run of ASM without scaling.

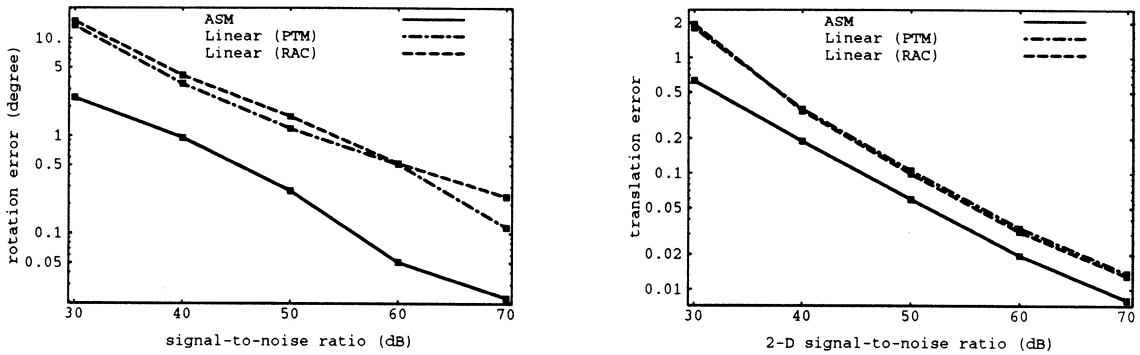

Figure 5.12: Result of Experiment **C1** for comparing linear methods and ASM. Error is in log scale. Each point in the plot represents 1,000 trials.

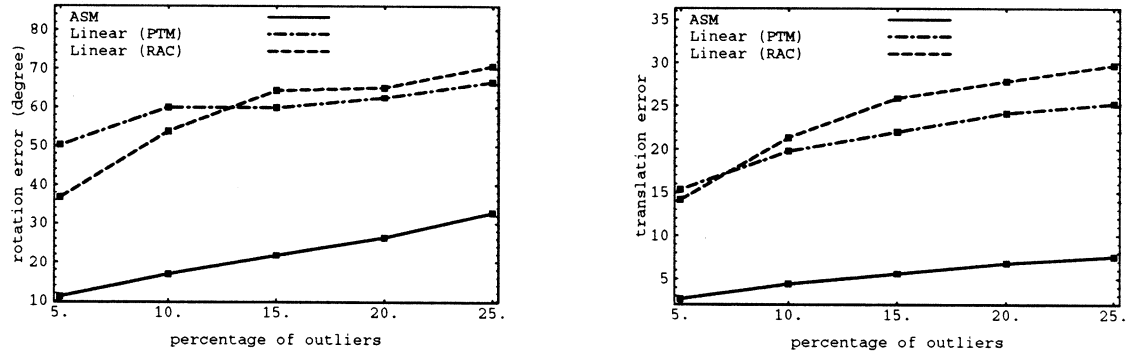

Figure 5.13: Result of Experiment **C2** for comparing linear methods and ASM. Error is in log scale. Each point in the plot represents 1,000 trials.

solute orientation phase of ASM is still able to find good intermediate linear solutions that lead to the correct one, and demonstrates the advantage of solving the pose estimation problem by simultaneous absolute orientation and scene reconstruction even if the absolute orientation phases are solved linearly.

Classical methods and ASM

The methods tested here are ASM, a linear method using the Projective Transform Matrix (PTM) formulation, and a classical method using Levenberg-Marquardt (LM) minimization. An implementation of LM (called `LMDIF`) in MINPACK (a public domain package from Argonne National Laboratory) is used in our experiments. LM

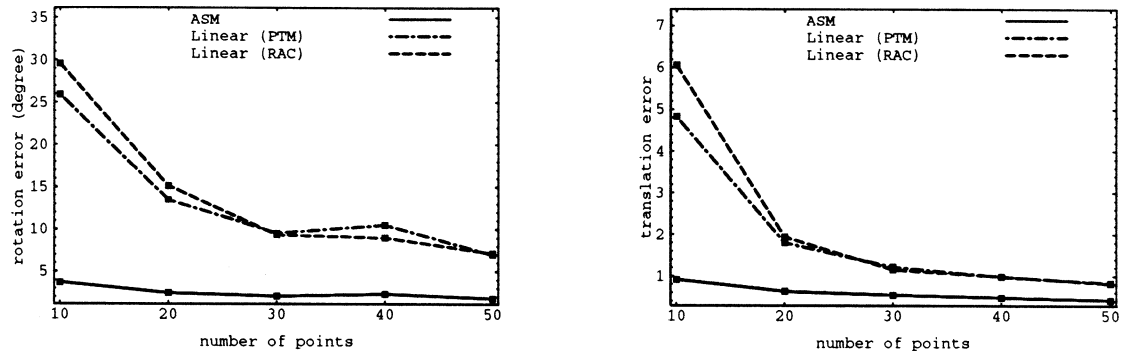

Figure 5.14: Result of Experiment **C3** for comparing linear methods and ASM. Error is in log scale. Each point in the plot represents 1,000 trials.

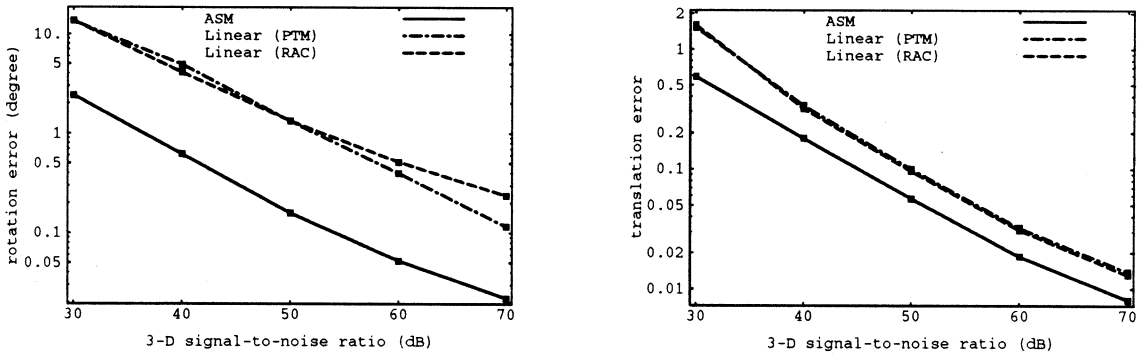

Figure 5.15: Result of Experiment **C4** for comparing linear methods and ASM. Error is in log scale. Each point in the plot represents 1,000 trials.

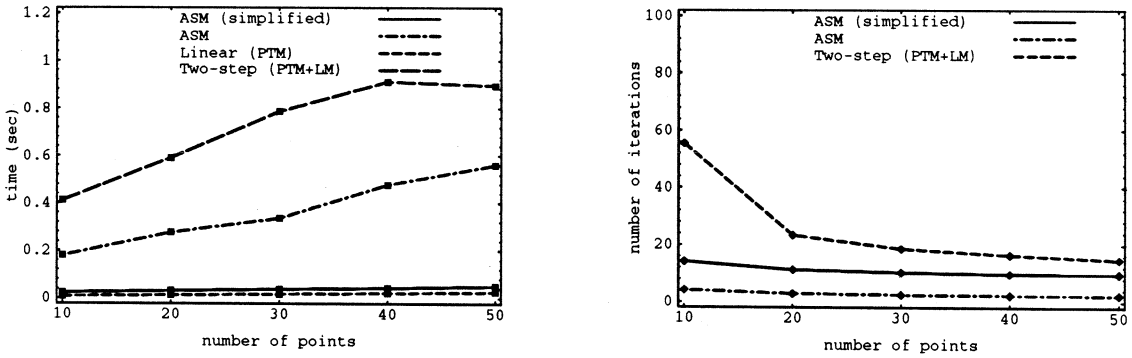

Figure 5.16: Running times and average numbers of iterations used by the tested methods. Each point in the plot represents 1,000 trials.

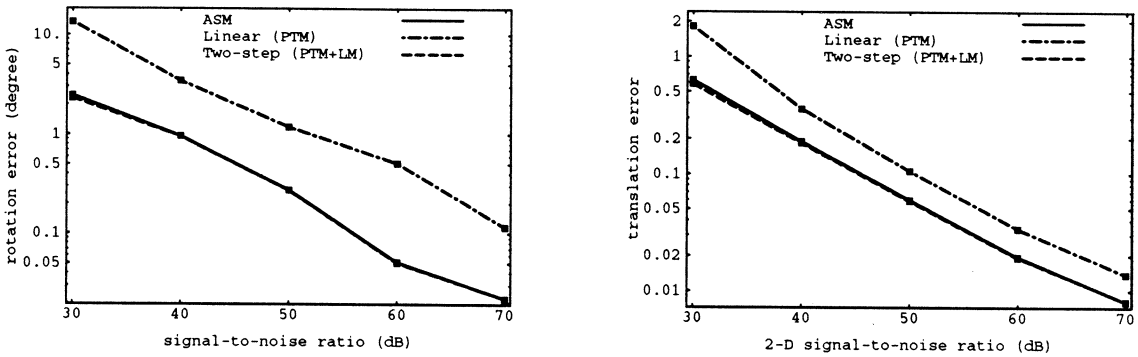

Figure 5.17: Result of Experiment **C1** for comparing ASM and the Leverberg-Marquardt method. Error is in log scale. Each point in the plot represents 1,000 trials.

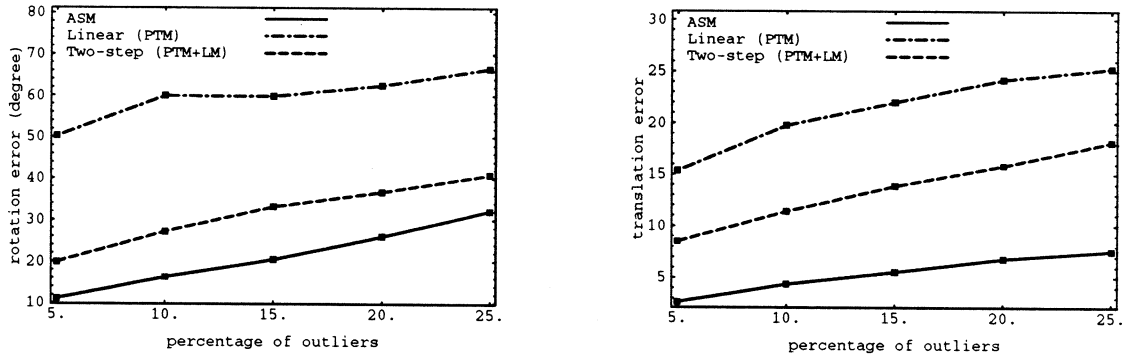

Figure 5.18: Result of Experiment **C2** for comparing ASM and the Leverberg-Marquardt method. Error is in log scale. Each point in the plot represents 1,000 trials.

starts from the initial solutions provided by PTM. This combination is an example of two-step methods that combine linear methods and classical nonlinear optimizations.

Figure 5.16 shows the average running times and number of iterations of the methods we tested against the number of model points. These times are measured for $\mathrm{SNR}_{img} = 60$ dB and $\mathrm{PO} = 0$. ASM is clearly much more efficient than LM, having about the same performance as LM without outliers (see Figures 5.17, 5.19 and 5.20). It significantly outperforms LM in the presence of outliers as shown by Figure 5.18.

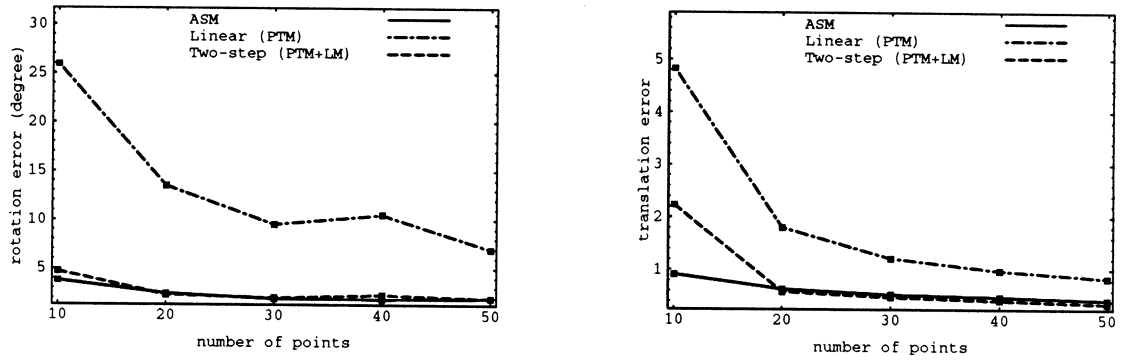

Figure 5.19: Result of Experiment **C3** for comparing ASM and the Leverberg-Marquardt method. Error is in log scale. Each point in the plot represents 1,000 trials.

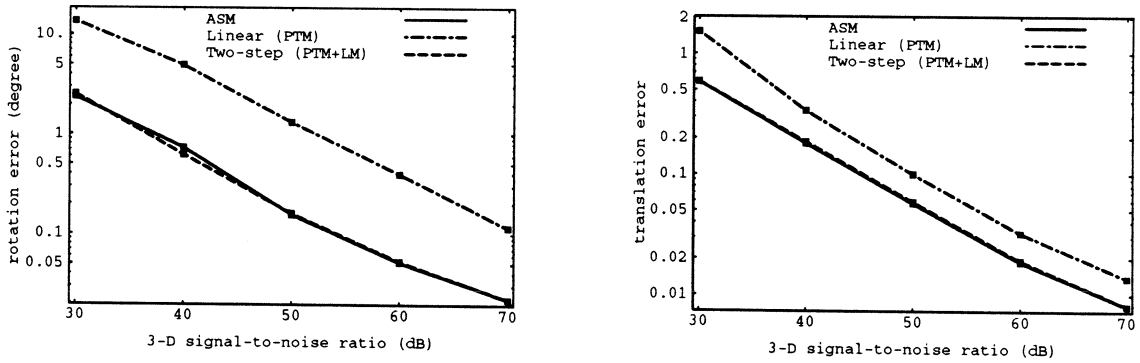

Figure 5.20: Result of Experiment **C4** for comparing ASM and the Leverberg-Marquardt method. Error is in log scale. Each point in the plot represents 1,000 trials.

Chapter 6

Robust Estimation

6.1 Outlier Process and Robust Estimation

The algorithms we developed in this dissertation are all least squares methods, which are well-known to be sensitive to outliers. The goal of robust methods is to remove or lessen the effect of outliers [38]. In Iteratively Reweighted Least Squares (IRLS), methods, this is done by assigning zero or small weight to potential outliers, which can be considered as *deciding* whether an observation is an outlier or not, although such decision may be a fuzzy one. *Outlier processes*, introduced and related to robust estimators in [7], can be used to represent such a *decision*. They are a generalization of *line processes* used in surface interpolation [8] for introducing discontinuity wherever continuity assumption is not valid. Outlier processes serve to break the Gaussian assumption implicit in least square methods when such an assumption is insufficient to model the observed data.

An objective function for robust estimation can be formulated as

$$f(r_i) = \sum_i \rho(r_i; \sigma) = \sum_i \phi(r_i^2; \sigma), \tag{6.1}$$

where $\rho(\cdot)$ (or $\phi(\cdot)$) is a robust estimator (also called the *object function*), and r_i is the residual for the ith observation. The scale parameter or shape parameter σ controls

the scale or the shape of the object function. Equation (6.1) can be converted to an objective function with outlier processes as

$$f(r_i, A_i) = \sum_i A_i r_i^2 + \sum_i \Psi(A_i; \sigma) \tag{6.2}$$

and vice versa, where $0 \leq A_i \leq 1, i = 1, \cdots, n$ is an outlier process, and $\Psi(\cdot)$ is a penalty function on A_i to prevent treating all residuals as outliers and ensure that the minimization of $f(r_i, A_i)$ is equivalent to that of $f(r_i)$. Computing A_i can be thought of as deciding whether r_i is an outlier: $A_i = 0$ means r_i is an outlier and $A_i = 1$ otherwise. We will see in the following that σ determines the "crispness" of the outlier process.

According to [7], given an object function $\phi(\cdot)$, the penalty function $\Psi(\cdot)$ on A_i can be determined as

$$\Psi(A_i; \sigma) = \phi(\varphi(A_i; \sigma)) - A_i \varphi(A_i; \sigma), \tag{6.3}$$

where $\varphi = \dot{\phi}^{-1}$. Conversely, given an outlier process A_i and the associated penalty function $\Psi(\cdot)$, the object function is

$$\rho(r_i; \sigma) = \inf_{0 \leq A_i \leq 1} (A_i r_i^2 + \Psi(A_i; \sigma)). \tag{6.4}$$

Assume that the parameter vector $\boldsymbol{\theta}$ is to be estimated from the set of residuals $r_i = r_i(\boldsymbol{\theta})$. Equation (6.1) and Equation (6.2) can be rewritten as

$$f(\boldsymbol{\theta}) = \sum_i \rho(r_i(\boldsymbol{\theta}); \sigma) \tag{6.5}$$

and

$$f(\boldsymbol{\theta}, A_i) = \sum_i A_i r_i^2(\boldsymbol{\theta}) + \sum_i \Psi(A_i; \sigma), \tag{6.6}$$

respectively.

With the above formulation, robust estimation becomes a computational problem of minimizing the objective function (6.6) with respect to $\boldsymbol{\theta}$ and A_i. Applying

the alternative subspace minimization technique to Equation (6.6) on the $\boldsymbol{\theta}$ and A_i subspaces yields

$$r_i^2(\boldsymbol{\theta}^{(k)}) + \frac{\partial \Psi(A_i^{(k+1)}; \sigma)}{\partial A_i} = 0, \tag{6.7}$$

$$A_i^{(k+1)} r_i(\boldsymbol{\theta}^{(k+1)}) \frac{\partial r_i(\boldsymbol{\theta}^{(k+1)})}{\partial \boldsymbol{\theta}} = 0,$$

which is an IRLS algorithm. Using Equation (6.3) to solve for $A_i^{(k+1)}$, we have

$$A_i^{(k+1)} = \frac{\dot{\rho}(r_i(\boldsymbol{\theta}^{(k)}); \sigma)}{r_i(\boldsymbol{\theta}^{(k)})}, \tag{6.8}$$

which is the standard reweighting equation used in robust M-estimation.

6.2 A Continuation Method for Robust Estimation

We have left out the estimation of the scale parameter σ in the IRLS algorithm (6.7). It is reasonable to assume that the residuals have a contaminated Gaussian distribution, or using the outlier-process formulation, a Gaussian distribution with outlier processes to break the connections of outlier observations to the Gaussian. The scale parameter σ is the standard deviation of the underlying Gaussian distribution of the residuals. A robust estimate of σ can be computed from the median of the absolute residuals using:

$$\sigma^{(k)} = 1.4286 \text{ median}_i |r_i^{(k)}|. \tag{6.9}$$

This comes from the fact that the median of the absolute values of a large sample from a unit-variance one-dimensional Gaussian distribution is 1/1.4826 [58].

On the other hand, controlling σ can be considered as a computational technique for non-convex minimization of the objective function (6.5) or its outlier process equivalent (6.6). The scale parameter σ can be exploited to construct a local convex

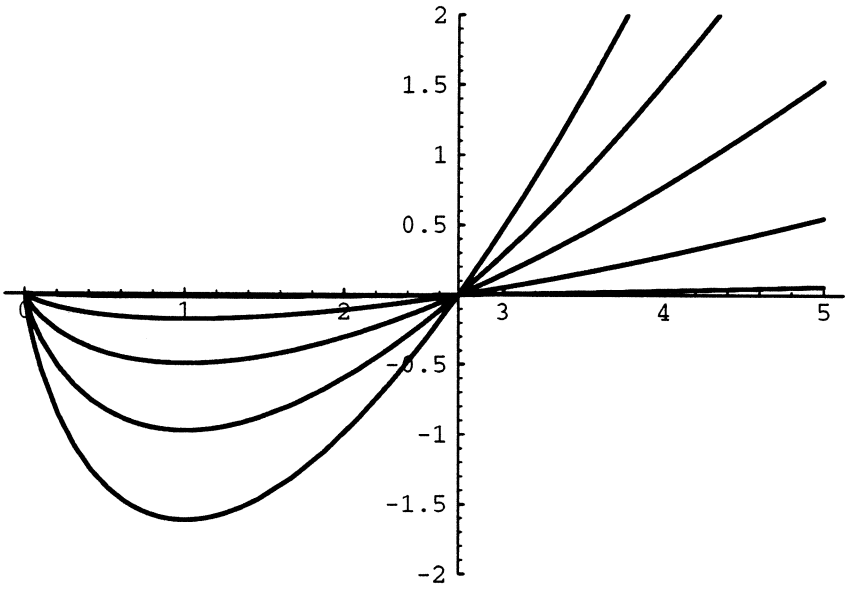

Figure 6.1: Plots of the penalty function $2\sigma^2 x(logx - 1)$ for $\sigma = 0.1, 0.3, 0.5, 0.7, 0.9$.

approximation to the objective function which can be readily minimized. For sufficiently large σ, the objective function is convex and has a single local minimum. From this point we start the search for the global maximum of the objective function with a slightly smaller σ, and the process repeats until the scale-adjusted objective function is very close to the original non-convex function. Such techniques are generally referred as *continuation methods*, or Graduated Non-Convexity (GNC) methods.

If we choose to use the Welsch estimator [7] $\rho(r_i;\sigma) = 1-e^{-r_i^2/2\sigma^2}$, then the penalty function for the corresponding outlier process A_i is $\Psi(A_i;\sigma) = 2\sigma^2 A_i(\log A_i - 1)$. Robust estimation of $\boldsymbol{\theta}$ can be formulated as that of minimizing

$$\sum_i (1 - e^{-r_i^2(\boldsymbol{\theta})/2\sigma^2}) \tag{6.10}$$

or equivalently

$$\sum_i A_i r_i^2(\boldsymbol{\theta}) + 2\sigma^2 \sum_i A_i(\log A_i - 1). \tag{6.11}$$

Using the IRLS algorithm formulated in Equation (6.7), the outlier process A_i can

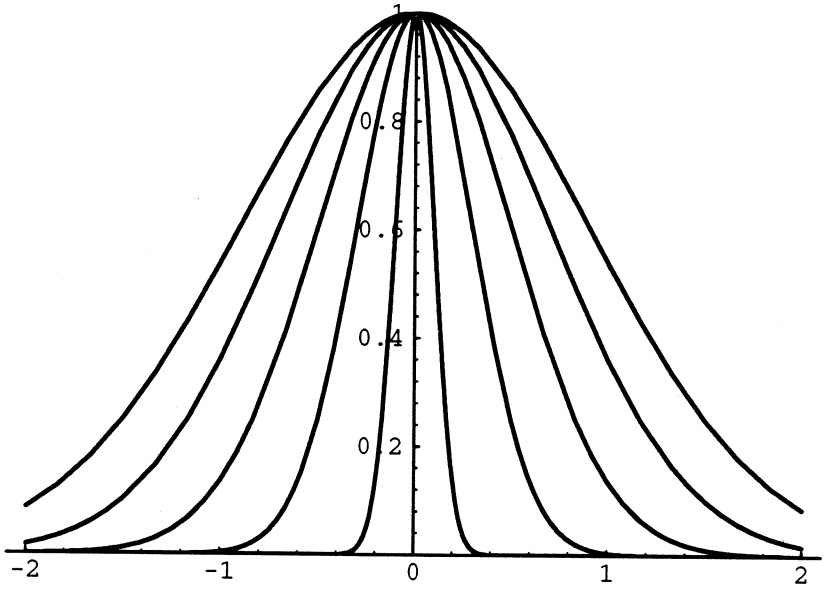

Figure 6.2: Plots of the soft-delta function $e^{-x^2/2\sigma^2}$ for $\sigma = 0.1, 0.3, 0.5, 0.7, 0.9$.

be computed by

$$A_i = e^{-r_i^2(\theta)/2\sigma^2}, \tag{6.12}$$

which gives a fuzzy decision (A_i between 0 and 1) whether r_i is an outlier. The scale parameter σ can be thought of controlling the fuzziness (or crispness) of the outlier processes. As $\sigma \to 0$, A_i becomes a binary variable. On the other extreme, when $\sigma = \infty$, $A_i \equiv 1$ meaning that the outlier process accepts the point correspondence without questioning. Leclerc [44] considered using a decreasing sequence of σ on the Welsch estimator as embedding the Dirac-delta function $\delta(r)$ into a scale space. In this aspect, the fuzzy outlier process (6.12) can be called a *soft delta* function.

We will use this algorithm has a skeleton method for robust absolute orientation and pose estimation. Our ASM algorithm and the Levenberg-Marquardt algorithm are used as the kernel in the inner loop.

6.3 Experiments

6.3.1 Absolute orientation

Assume a set of 3D model points $\mathbf{x}_i, i = 1, \ldots n$, and a set of corresponding 3D scene points $\mathbf{y}_i$ including potential outliers. The parameter vector to be estimated is $\boldsymbol{\theta} = (R, \mathbf{t})$. Ideally, each pair of model point and scene point is related by

$$\mathbf{y}_i = R\mathbf{x}_i + \mathbf{t}. \tag{6.13}$$

Given noisy observations $\tilde{\mathbf{x}}_i$ and $\tilde{\mathbf{y}}_i$, a deviation from the ideal case, or a residual is defined as the Mahalanobis distance:

$$r_i^2(R, \mathbf{t}) = (R\tilde{\mathbf{x}}_i + \mathbf{t} - \tilde{\mathbf{y}}_i)^t \boldsymbol{\Sigma}_i^{-1} (R\tilde{\mathbf{x}}_i + \mathbf{t} - \tilde{\mathbf{y}}_i), \tag{6.14}$$

where $\boldsymbol{\Sigma}_i = R\boldsymbol{\Sigma}_{\tilde{\mathbf{x}}_i} R^t + \boldsymbol{\Sigma}_{\tilde{\mathbf{y}}_i}$.

If we choose to use the Welsch or the Leclerc estimator, R and $\mathbf{t}$ can be solved robustly using the following objective function

$$\sum_i (1 - e^{-r_i^2(R,\mathbf{t})/2\sigma^2}), \tag{6.15}$$

which is equivalent to

$$\sum_i A_i r_i^2(R, \mathbf{t}) + 2\sigma^2 \sum_i (A_i \log A_i - A_i), \tag{6.16}$$

where A_i are outlier processes.

The IRLS method Equation (6.7) is used in the inner loop of the continuation method described in Section 6.2.

We compare robust and non-robust absolute orientation methods using the synthetic data generated using the protocol described in Section 5.3. A fraction of the point correspondences are replaced by outliers. The rotation errors and the translation errors plotted against increasing percentages of outliers are summarized in Figure 6.3. We can see that the robust method produces much more accurate results for data

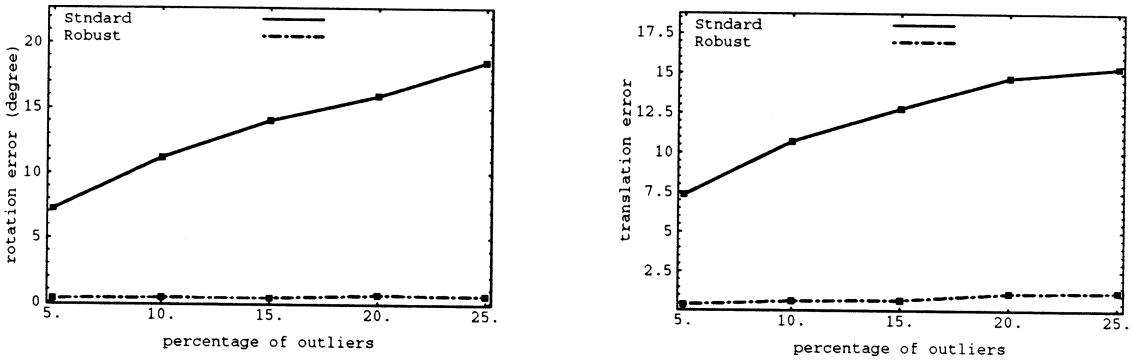

Figure 6.3: Comparing robust and non-robust absolute orientation methods against increasing percentages of outliers.

with up to 25% outliers. On the other hand, the results computed using non-robust method, with rotation errors greater than 5 degree, become useless even with as little as 5% outliers.

6.3.2 Object pose estimation

Most work on robust pose estimation uses classical methods as kernels of IRLS or other modified least squares methods for outlier rejection. It is assumed that a good initial guess is available so that a large portion of outliers can be detected by the first reweighting, and the following iterations will start with a reasonable assignment of weights. In the cases where good initial approximate solutions are not available, the kernel has to deal with large residuals caused by outliers. Methods that focus only on good local convergence, like the Gauss-Newton method, will certainly fail. More robust methods like the Levenberg-Marquardt method actually become a steepest descent method, and hence are very slow to converge. Classical optimization methods for the most part are designed for solving general problems. A specialized optimization method that exploits the inherent structure of the problem may better solve the problem, especially in the presence of noise and outliers.

Recall that the ASM method requires a scaling step. In robust estimation, we

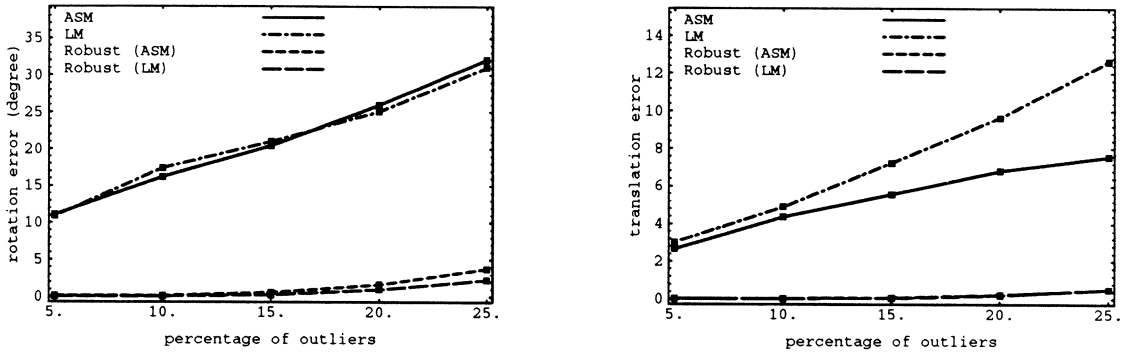

Figure 6.4: Comparing robust pose estimation methods using ASM and LM.

need a robust measure for scales. Define the robust MSDCs for $\tilde{\mathbf{x}}_i$ and $\tilde{\mathbf{y}}_i$ as

$$\text{MSDC}(\tilde{\mathbf{x}}_i) \stackrel{\text{def}}{=} \sqrt{\frac{\sum_i A_i |\tilde{\mathbf{x}}_i - \bar{\mathbf{x}}|^2}{\sum_i A_i}} \tag{6.17}$$

and

$$\text{MSDC}(\tilde{\mathbf{y}}_i) \stackrel{\text{def}}{=} \sqrt{\frac{\sum_i A_i |\tilde{\mathbf{y}}_i - \bar{\mathbf{y}}|^2}{\sum_i A_i}}, \tag{6.18}$$

where the robust centroids are

$$\bar{\mathbf{x}} = \frac{\sum_i A_i \tilde{\mathbf{x}}_i}{\sum_i A_i}, \text{ and } \bar{\mathbf{y}} = \frac{\sum_i A_i \tilde{\mathbf{y}}_i}{\sum_i A_i}. \tag{6.19}$$

Experiment **C2** in Section 5.3 was performed for the robust IRLS algorithm [38] using the Welsch estimator [9] with the scale-space continuation technique described in Section 6.2. Both ASM and LM were used in the IRLS inner loop.

As shown in Figure 6.4, ASM performs about the same as LM when used as an IRLS kernel. However, Figure 6.5 and 6.6 show that ASM makes IRLS more efficient. Notice that in Experiment **C2**, no prior initial approximate solution is given to the pose estimation methods. Outlier rejection is totally based on the given point correspondences.

6.3.3 Hand-eye calibration

Given the 3D coordinates of the model points and their corresponding camera projections, we compute the rotation and translation that relate the coordinate frame of

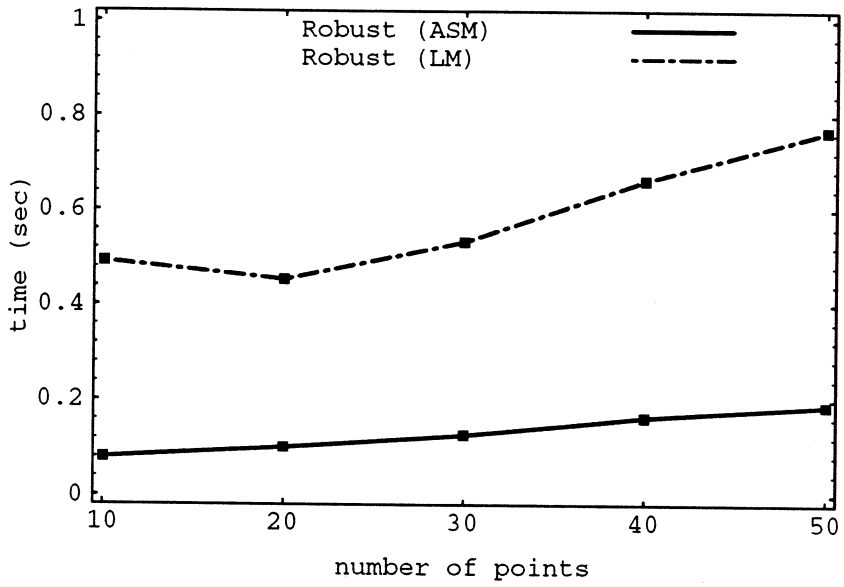

Figure 6.5: Average running times of the robust pose estimation methods. We choose PO = 20%.

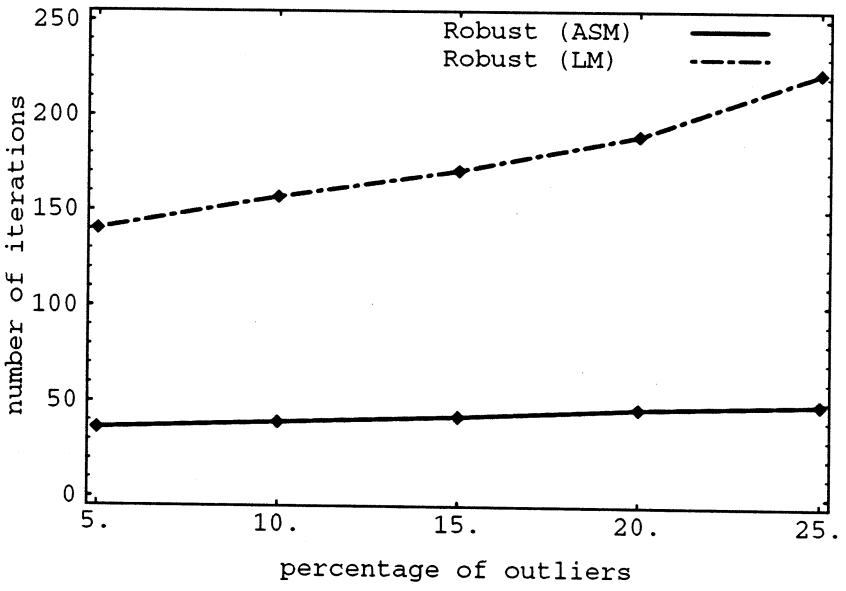

Figure 6.6: Average numbers of iterations of the robust pose estimation methods for different percentages of outliers.

a robot arm to that of a camera.

Experimental setting

Our experimental setting for hand-eye calibration consists of a Zebra Zero robot arm, a Cohu camera with an 8 mm lens, a Sony XC-77 camera with a 12.5 mm lens, and two Imaging Technologies digitizers attached to a Sun Sparc II workstation via a Solflower SBus-VME adapter. The size of the video image received from the cameras is 640-by-480. The intrinsic parameters of both cameras were determined offline using Tsai's two-step method [61].

The physical conditions are shown in Figure 6.7. The Sony XC-77 (middle, bottom) was positioned nearly aligned with the robot coordinate system and was tuned to have sharp images. The Cohu (left, top) was positioned more to the side, and delivered more defocused images. Data was acquired by moving the arm to 35 positions, and at each position compiling a data pair consisting of the absolute coordinates of a feature in the robot frame (computed from the robot inverse kinematics), and the image coordinates of the feature provided by tracking the lower right corner of the floppy disk. [1] This process was repeated 5 times to obtain 5 datasets for each camera. Another 5 datasets for each camera were obtained by adding one outlier.

Results and discussion

The results of the calibration methods are compared by computing object space error which is determined by comparing the reference points to their orthogonal projections on the respective lines of sight. The results for the five trials for both cameras are plotted in Figure 6.10.

Non-robust methods were tested on the outlier-free datasets. It turned out that the results given by the two-step methods are very close to those given by ASM, so only the ASM results are plotted. Given that the two-step methods required

[1] The tracking system is more fully described in [28].

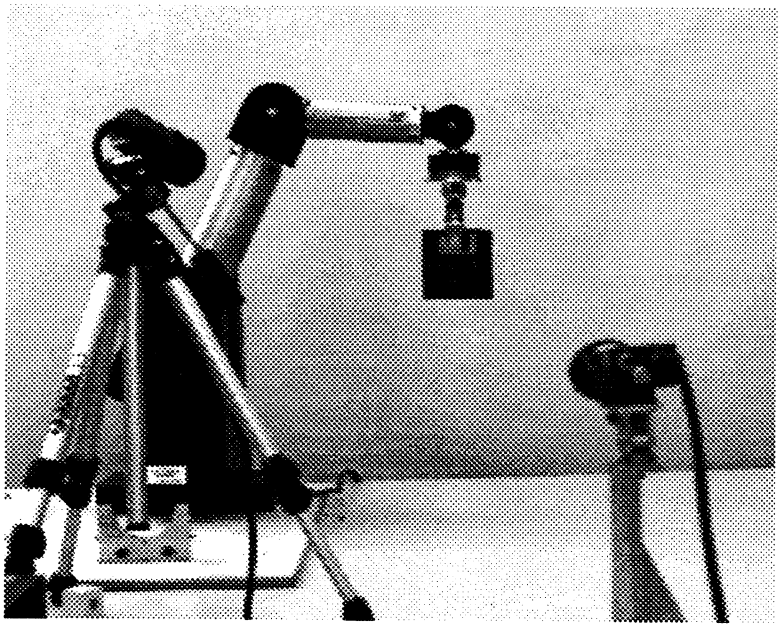

Figure 6.7: The experimental setup showing the positions of the two cameras relative to the robot arm.

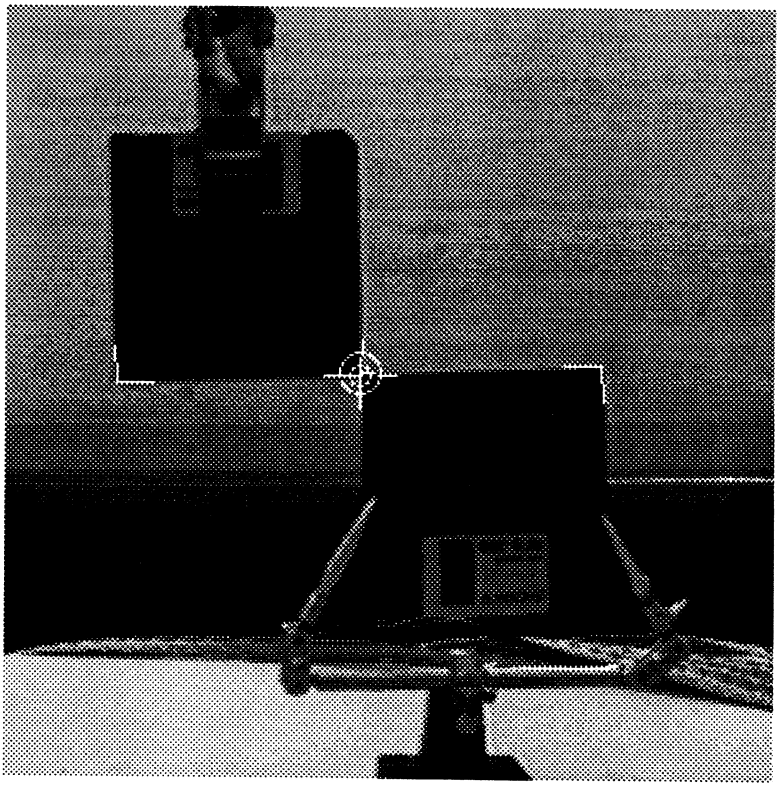

Figure 6.8: An image from the cameras showing the tracking used to generate image feature point data.

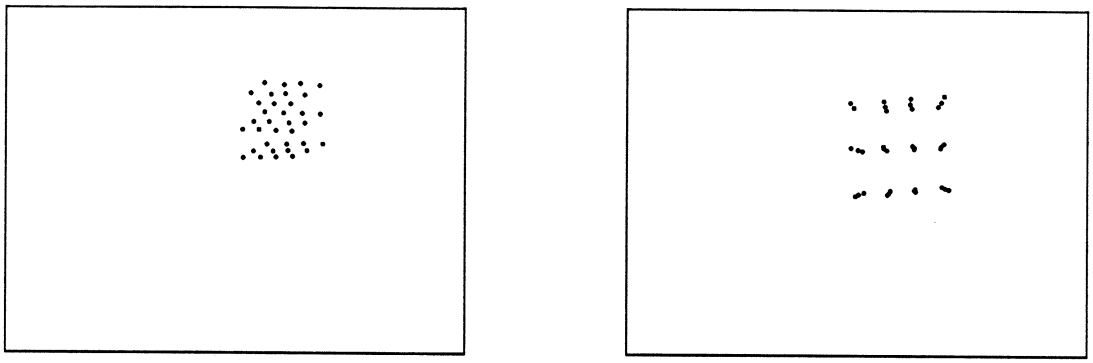

Figure 6.9: The projections of 35 model points as seen through: (*left*) Cohu camera and (*right*) Sony camera.

several times as long to converge (see Figure 6.5), ASM would clearly be preferred in these circumstances. The linear methods used in the simulation were used in this experiment. It is clear that ASM is more stable and accurate than the linear method.

The robust IRLS algorithm in Equation (6.7) using ASM was applied to the datasets with one extra outlier. It is interesting to note that the robust method discovered another outlier in the second dataset for each camera. The reason for that outlier was that the Zebra Zero robot arm reached its joint limit when generating the second dataset.

One difference between the simulations and these tests is that the errors in the model points are significant (on the order of up to a centimeter). Despite these errors, ASM appears to compute an accurate transformation.

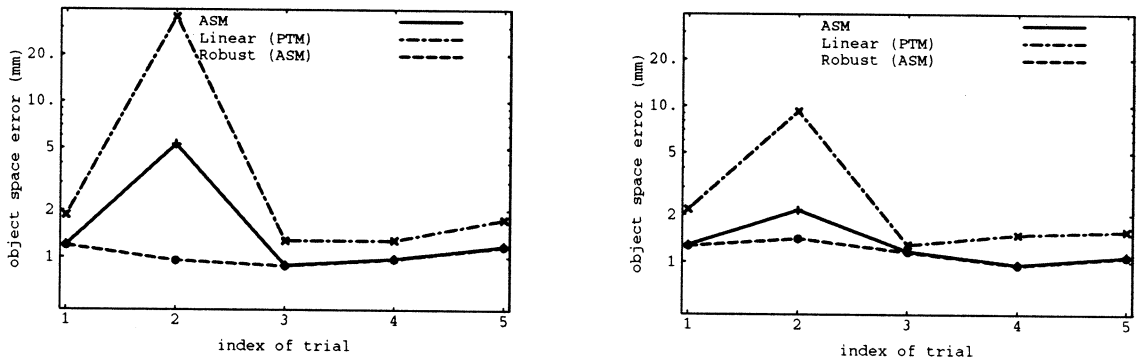

Figure 6.10: Results of the hand-eye calibration experiments for (*left*) Cohu and (*right*) Sony XC-77 cameras. Mean object space errors in pixels are measured for each data set.

Chapter 7

Model Matching

7.1 From Robust Estimation to Model Matching

Model matching and robust pose estimation are usually considered as two successive stages of problem solving. Model matching finds the correct correspondence between two feature sets. In pose estimation, the pose is computed from the feature correspondence assumed to be available from model matching.

The popular hypothesis-and-test paradigm of model matching [5,40,46] actually relies on efficient and accurate pose estimation to solve for the correspondence. It hypothesizes (estimates) the object pose which is most consistent with a given partial correspondence, and seeks supporting or negating evidence by comparing the data to the object transformed by the estimated pose. When matching is done, the correct pose is also found.

On the other hand, pose estimation has to take into account the problem of possible matching errors which introduce outliers in the point correspondences. Outlier rejection in robust methods can actually be considered as a further stage of correspondence solving.

It turns out that model matching and robust pose estimation have a considerable amount of overlap in functionalities. We present a new problem solving scheme that

better integrates the model matching stage and the pose estimation stage. In terms of robust methods, it can be considered as *aggressive*, since it not only tries to deny incorrect matches, but also tries to establish the correct ones. In terms of model matching, it is *robust*, since the correspondence result is gradually improved and verified with respect to the pose.

7.2 Correspondence Processes

Within the outlier rejection framework, when a scene feature is treated as an outlier, it is considered as unidentifiable and should be discarded. However, it is probable that such a scene feature actually comes from another model feature. We generalize outlier processes to *correspondence processes* A_{ia} such that each object feature can have more than one candidate scene feature and vice versa. The candidates can be selected with respect to feature attributes, such as color or grey level. In the most general case, in which the matching is purely *location-based*, a residual r_{ia} and a correspondence process A_{ia} are associated with each pair of object and scene points $(\mathbf{x}_i, \mathbf{y}_a), i = 1, \ldots, m, a = 1, \ldots, n$. $A_{ia} = 1$ means that $\mathbf{y}_a$ is identified with the model point $\mathbf{x}_i$. Otherwise, $\mathbf{y}_a$ should be considered as a spurious outlier, or coming from some model point other than $\mathbf{x}_i$. If the Welsch estimator is used again, the problems of model matching and pose estimation can be integrated into a single objective function using the algebraic transformation techniques described in [53]

$$\sum_{ia} A_{ia} r_{ia}^2(\boldsymbol{\theta}) + 2\sigma^2 \sum_{ia} (A_{ia} \log A_{ia} - A_{ia}), \tag{7.1}$$

or

$$\sum_{ia} (1 - e^{-r_{ia}^2(\boldsymbol{\theta})/2\sigma^2}). \tag{7.2}$$

As stated in [7], the outlier process formulations provide the advantage of being able to exploit interactions among outlier processes. This is especially true for

correspondence processes. We wish that the set of correspondence processes provide *feasible interpretation* which satisfies the following criterion:

> *Each scene feature is either identified with one and only one object feature or is considered as a spurious outlier. Each object feature either appears as one and only one scene feature or is considered deleted from the scene.*

Mathematically, this criterion can be written as a set of constraints on the correspondence matrix $A = (A_{ia})$:

$$\sum_i A_{ia} \leq 1 \qquad \text{(column constraint)} \tag{7.3}$$

$$\sum_a A_{ia} \leq 1 \qquad \text{(row constraint), and} \tag{7.4}$$

$$A_{ia} \in 0, 1 \qquad \text{(integrality constraint).} \tag{7.5}$$

When $\sum_i A_{ia} = 0$, the model point $\mathbf{x}_i$ is said to be *missing*. Similarly, when $\sum_a A_{ia} = 0$, the scene point $\mathbf{y}_a$ is said to be *spurious*.

7.3 A Continuation Method for Model Matching

The same robust IRIS algorithm described in Chapter 6 can be readily applied. The same continuation method developed in Section 6.2 for robust estimation can be easily adapted to model matching. Specifically, if we define $T = 2\sigma^2$ as a *temperature*, and call the matching objective functions *energy functions*, the continuation method can be referred to as *deterministic annealing* [55,59,70], which is a computational technique developed for solving combinatorial optimization problems with 0-1 variables.

At each fixed scale σ, minimizing objective function (7.1) subject to constraints (7.3), (7.4) and (7.5) on A can be done in two phases. In the first phase, A is determined using (6.7), and the feasible-interpretation constraint is satisfied by *iterative projective scaling* (IPS) [56,22], where row-column normalization is repeatedly applied to A until it converges. In the second phase, A is fixed, and the problem becomes a weighted least-squares problem.

On the other hand, when the constraints are not enforced, the objective function favors

$$\sum_{ia} A_{ia} \approx n \tag{7.6}$$

with some sufficiently small σ, which implies that there are approximately n matches among all possible matches. Mjolsness [52] has shown that Equation (7.2) can be derived from

$$\sum_{ia} A_{ia} r_{ia}^2(\boldsymbol{\theta}) \tag{7.7}$$

by approximately enforcing the constraint that

$$\sum_{ia} A_{ia} = n, \tag{7.8}$$

which is referred to as the *multinomial constraint.* An advantage of using (7.8) is that we do not have to deal with spurious or missing features explicitly; they are modeled by empty rows or columns of the A matrix. While this constraint is weaker, we can argue that it is a good approximation to real matching constraints. An entropy argument in favor of this constraint is that among matrices satisfying (7.8), the vast majority have low occupancy for most rows and columns. There is also an energy argument: multiple assignments are allowed but discouraged by the equivalent objective function (7.2) unless r_{ia} and $r_{ja}, i \neq j$, or r_{ia} and $r_{ib}, a \neq b$ happen to be very close (within σ) to each other. So (7.8) is plausible as the sole constraint on A.

The correspondence process formulation has been used in 2D-2D point matching [48]. An algebraic manipulated version of the point-matching objective function has been applied to 2D-2D line segment matching [49]. A similar objective function is also developed for 3D-3D model matching. For sufficiently small σ, minimizing (7.2) can be shown to be equivalent to the generalized Hough transform or template matching. A continuation method for minimizing such an objective function can be considered as a coarse-to-fine template matching.

7.4 2D-2D Point Matching

Consider the problem of locating a two-dimensional "model" object that is believed to appear in the "scene". Assume first that both the model and the scene are represented by a set of 2D "points" respectively, $\mathbf{x}_i$ and $\mathbf{y}_a$. The problem is to recover the actual transformation (translation and rotation) that relates the two sets of points. Following the framework developed above, such a problem can be solved by minimizing the following objective function

$$f(A,\theta,\mathbf{t}) = \sum_i A_{ia}\|R_\theta \mathbf{x}_i + \mathbf{t} - \mathbf{y}_a\|^2 + 2\sigma^2 \sum_{ia} A_{ia}(\log A_{ia} - 1), \tag{7.9}$$

where $A_{ia} = A$ represents the unknown correspondence, R_θ is a rotation matrix with rotation angle θ, and $\mathbf{t}$ is a 2D translation vector. The equivalent "robust" objective function obtained by applying algebraic transformations to 7.9 is

$$\sum_{ia}(1 - e^{-\|R_\theta \mathbf{x}_i + \mathbf{t} - \mathbf{y}_a\|^2/2\sigma^2}). \tag{7.10}$$

The problem then becomes that of maximizing

$$f_{point}(R,\mathbf{t}) = \sum_{ia} e^{-\|R_\theta \mathbf{x}_i + \mathbf{t} - \mathbf{y}_a\|^2/2\sigma^2}. \tag{7.11}$$

which in turn can be interpreted as minimizing the Euclidean distance between two Gaussian-blurred images containing the scene points $\mathbf{x}_i$ and a transformed version of the model points $\mathbf{y}_a$. Assuming that there is only translation between the model and the scene, each containing 20 points, figure 7.1 demonstrates the shape of (7.11) from coarse to fine scales for a simpler case in which only translation is applied to the model, and the objective function becomes

$$f_{point}(\mathbf{t}) = \sum_{ia} e^{-\frac{1}{2\sigma^2}\|\mathbf{x}_i + \mathbf{t} - \mathbf{y}_a\|^2}. \tag{7.12}$$

At $\sigma = 0.16$, there is only one single peak. We can see that the location of this peak falls in the convex region around the highest peak at $\sigma = 0.08$. As a consequence, the latter should be very easily reached via a local search from the former. Our

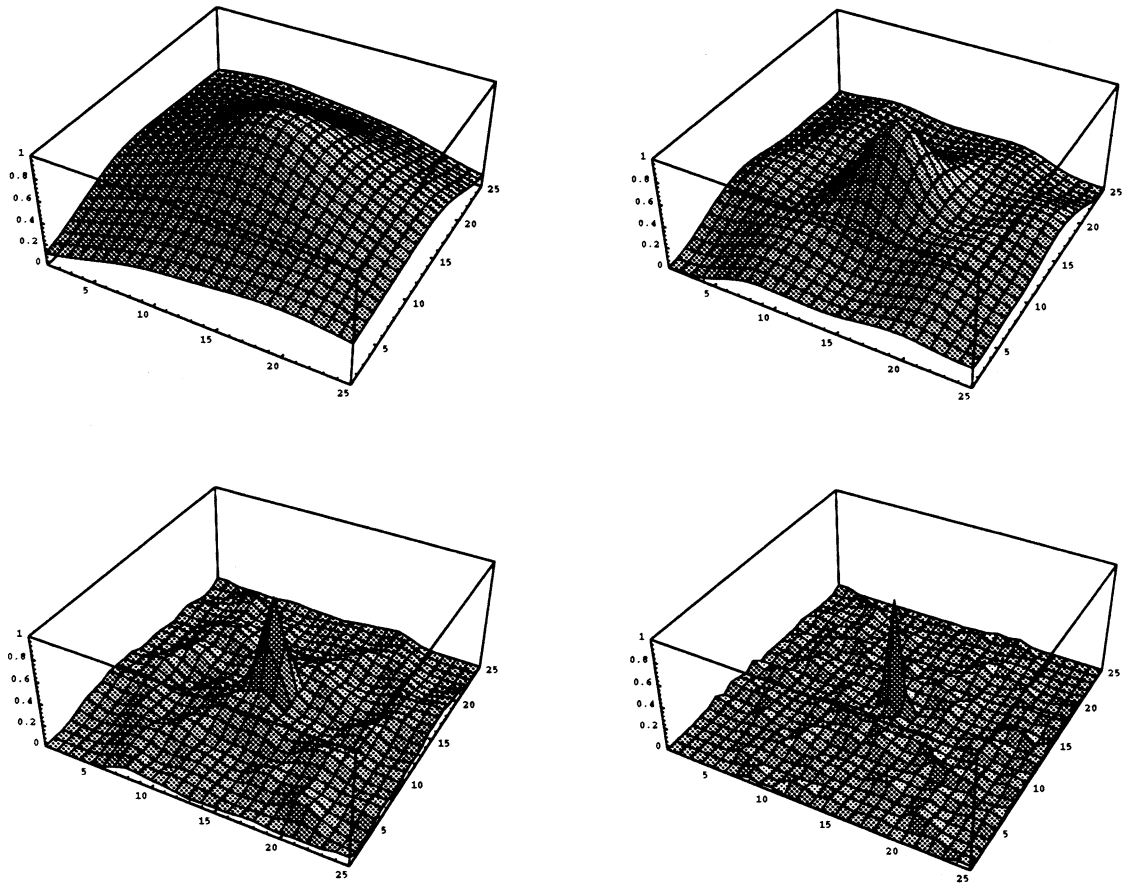

Figure 7.1: The objective functions for translation only at at different scales (σ): 0.16 (*top left*), 0.08 (*top right*), 0.04 (*bottom left*) and 0.02 (*bottom right*), are plotted against x and y components of translation.

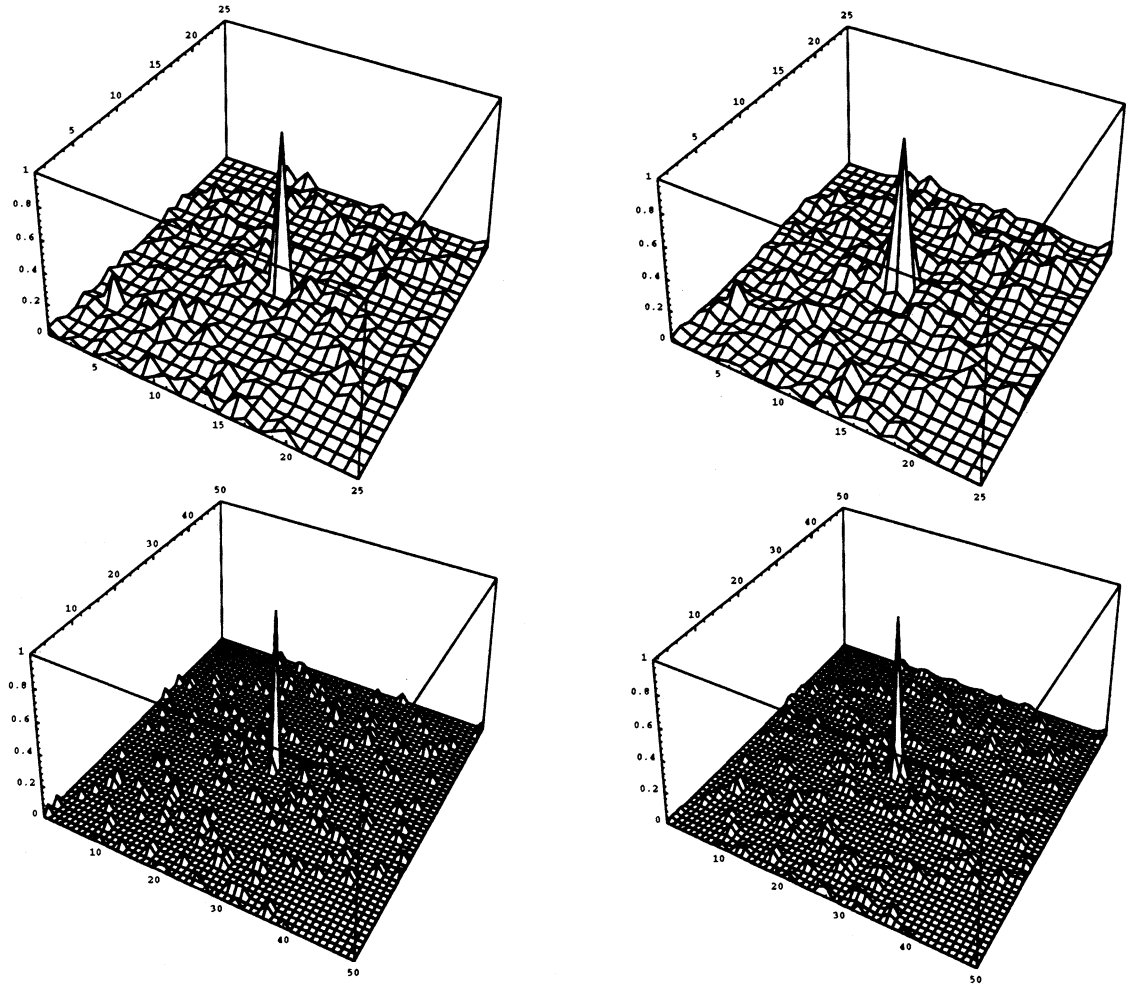

Figure 7.2: Comparing template matching (*left*) and optimization using the objective function $f_{point}(\mathbf{t})$ (*right*). We use 25-by-25 (*top*) bins and 50-by-50 (*bottom*) bins for translation. In (*left*), the vertical scale represents the number of votes collected at each bin, and the horizontal scales are indexes for translation bins. The width of each bin along either x or y direction is σ. In (*right*), the vertical scale represents the value of $f_{point}(\mathbf{t})$ at discrete points sampled at the center of each bin. The values are normalized so that the central spike in both plots are of height 1.

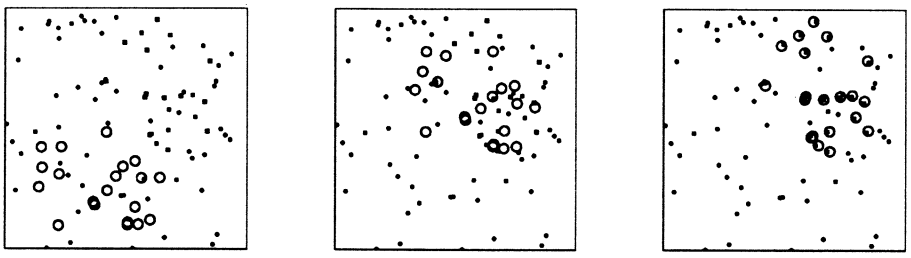

Figure 7.3: Shown here is an example of matching a 20-point model to a scene with 66.7% spurious outliers. The model is represented by circles. The set of square dots is an instance of the model in the scene. All other dots are outliers. From left to right are configurations at the annealing steps 1, 10, and 51, respectively.

continuation method works by decreasing σ slowing so that the current highest peak can always be reached from the location of the highest peak at previous σ. Hopefully, the central peak at $\sigma = 0.02$ can be reached even though there are many spurious peaks in the original objective function. Figure 7.2 shows the relation of maximizing f_{point} to template matching, which maximizes

$$\sum_{ia} \text{Indicator}((\mathbf{x}_i + \mathbf{t}_{kl} - \mathbf{y}_a) \in [-\epsilon_t, \epsilon_t] \times [-\epsilon_t, \epsilon_t]) \tag{7.13}$$

over k, l, where the translation space is partitioned into a 2D array of bins. $\mathbf{t}_{kl}$ are centers of the bins, and ϵ_t is the width of each bin. By associating the bin width with σ, optimizing a model matching objective function such as (7.11) works like template matching.

A typical run-time behavior of the algorithm is illustrated in Figure 7.3.

7.5 2D-2D Line-Segment Matching

In many vision problems, representation of images by line segments has the advantage of compactness and subpixel accuracy along the direction transverse to the line. However, such a representation of an object may vary substantially from image to image due to occlusions and different illumination conditions.

7.5.1 Indexing points on line segments

The problem of matching line segments can be thought of as a point matching problem in which each line segment is treated as a dense collection of points. Assume now that both the scene and the model are represented by a set of line segments $\mathbf{s}_i$ and $\mathbf{m}_a$, respectively. Both the model and the scene line segments are represented by their endpoints as $\mathbf{s}_i = (\mathbf{p}_i, \mathbf{p}'_i)$ and $\mathbf{m}_a = (\mathbf{q}_a, \mathbf{q}'_a)$, where $\mathbf{p}_i, \mathbf{p}'_i$, and $\mathbf{q}_a, \mathbf{q}'_a$ are the endpoints of the ith scene segment and the ath model segment, respectively. The locations of the points on each scene segment and model segments can be parameterized as

$$\mathbf{x_i} = \mathbf{s}_i(u) = \mathbf{p}_i + u(\mathbf{p}'_i - \mathbf{p}_i),\ u \in [0,1] \text{ and} \tag{7.14}$$

$$\mathbf{y_a} = \mathbf{m}_a(v) = \mathbf{q}_a + v(\mathbf{q}'_a - \mathbf{q}_a),\ v \in [0,1]. \tag{7.15}$$

Now the model points and the scene points can be though of as indexed by $\mathbf{i} = (i, u)$ and $\mathbf{a} = (a, v)$. Using this indexing, we have $\sum_{\mathbf{i}} \propto \sum_i l_i \int_0^1 du$ and $\sum_{\mathbf{a}} \propto \sum_a l_a \int_0^1 dv$, where $l_i = \|\mathbf{p}_i - \mathbf{p}'_i\|$ and $l_a = \|\mathbf{q}_a - \mathbf{q}'_a\|$. The point matching objective function (7.11) can be specialized to line segment matching as the following objective function

$$f_{seg}(\theta, \mathbf{t}) = \sum_{ia} l_i l_a \int_0^1 \int_0^1 e^{-\frac{1}{2\sigma^2}\|R_\theta \mathbf{m}_a(v) + \mathbf{t} - \mathbf{s}_i(u)\|^2} du\ dv, \tag{7.16}$$

originally developed in [52]. Here we make a correction to the original objective function by adding the length terms l_i and l_a of model and scene line segments, respectively.

As a special case of point matching objective function, (7.16) can readily be optimized by the algorithm previously developed for point matching problem.

7.5.2 Gaussian sum approximation

The finite integrals in (7.16) are simplified as infinite Gaussian integrals by approximating the box function $\Theta(t)$ with a sum of three Gaussian as shown in Figure 7.4:

$$\Theta(t) \equiv \begin{cases} 1 & \text{if } t \in [0,1] \\ 0 & \text{otherwise} \end{cases} \approx \sum_{k=1}^{3} A_k \exp -\frac{1}{2}\frac{(c_k - t)^2}{\sigma_k^2}. \tag{7.17}$$

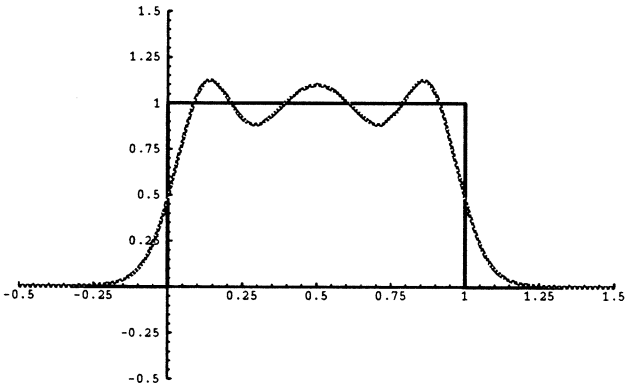

Figure 7.4: Approximating $\Theta(t)$ by a sum of 3 Gaussian.

By numerical minimization of the Euclidean distance between these two functions of t, the parameters may be chosen as $A_1 = A_3 = 0.800673$, $A_2 = 1.09862$,

$\sigma_1 = \sigma_3 = 0.0929032$, $\sigma_2 = 0.237033$, $c_1 = 1 - c_3 = 0.116807$, and $c_2 = 0.5$, as computed in [52].

Using this approximation, each finite double integral in (7.16) can be replaced by

$$\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} e^{-\frac{1}{2\sigma^2}\|R_\theta \mathbf{m}_a(v)+\mathbf{t}-\mathbf{s}_i(u)\|^2}\,\Theta(u)\Theta(v)\,du\,dv \approx$$

(7.18)

$$\sum_{k,l=1}^{3} A_k A_l \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} e^{-\frac{1}{2\sigma_k^2}(c_k-u)^2} e^{-\frac{1}{2\sigma_l^2}(c_l-v)^2} e^{-\frac{1}{2\sigma^2}\|R_\theta \mathbf{m}_a(v)+\mathbf{t}-\mathbf{s}_i(u)\|^2} du\,dv.$$

Each of these nine Gaussian integrals can be done exactly. Defining

(7.19) $$\mathbf{v}_{iakl} = \mathbf{s}_i(c_k) - R_\theta \mathbf{m}_a(c_l) - \mathbf{t}$$

(7.20) $$\hat{\mathbf{p}}_i = \mathbf{p}_i' - \mathbf{p}_i,\ \hat{\mathbf{q}}_a = R_\theta(\mathbf{q}_a' - \mathbf{q}_a),$$

(7.18) becomes

$$2\pi\sigma^2 l_i l_a \sum_{k,l=1}^{3} \frac{A_k A_l \sigma_k \sigma_l}{\sqrt{(\sigma^2+\hat{\mathbf{p}}_i^2\sigma_k^2)(\sigma^2+\hat{\mathbf{q}}_a^2\sigma_l^2) - \sigma_i^2\sigma_j^2(\hat{\mathbf{p}}_i\cdot\hat{\mathbf{q}}_a)^2}}$$

(7.21) $$\times \exp -\frac{1}{2}\frac{\mathbf{v}_{iakl}^2\sigma^2 + (\mathbf{v}_{iakl}\times\hat{\mathbf{p}}_i)^2\sigma_k^2 + (\mathbf{v}_{iakl}\times\hat{\mathbf{q}}_a)^2\sigma_l^2}{(\sigma^2+\hat{\mathbf{p}}_i^2\sigma_k^2)(\sigma^2+\hat{\mathbf{q}}_a^2\sigma_l^2) - \sigma_k^2\sigma_l^2(\hat{\mathbf{p}}_i\cdot\hat{\mathbf{q}}_a)^2}.$$

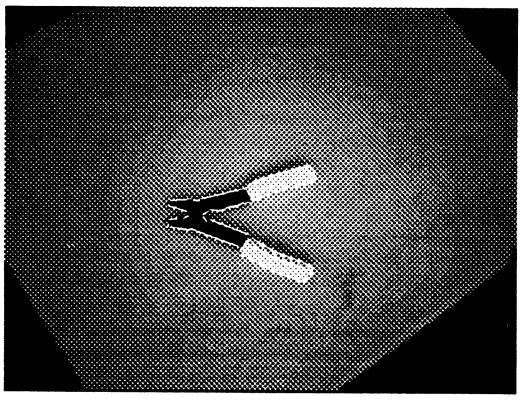

Figure 7.5: The polygonal representation of the model is overlayed on the image from which the line segments are extracted.

From the Gaussian sum approximation, we get a closed form objective function which can be readily optimized to give a solution to the line segment matching problem.

7.5.3 Results and discussions

The line segment matching algorithm described in this paper was tested on scenes captured by a CCD camera producing 640×480 images, which were then processed by an edge detector. Line segments were extracted using a polygonal approximation to the edge images. The model line segments were extracted from a scene containing a canonically positioned model object (Figure 7.5). They were then matched to those extracted from a scene containing differently positioned and partially occluded model object (Figure 7.6). The result of matching is shown in Figure 7.7.

7.6 3D-3D Point Matching

Extending the robust objective function for absolute orientation (6.16), we have

$$\sum_{ia} A_{ia} \|R\mathbf{x}_i + \mathbf{t} - \mathbf{y}_a\|^2 + 2\sigma^2 \sum_{ia} (A_{ia} \log A_{ia} - A_{ia}). \tag{7.22}$$

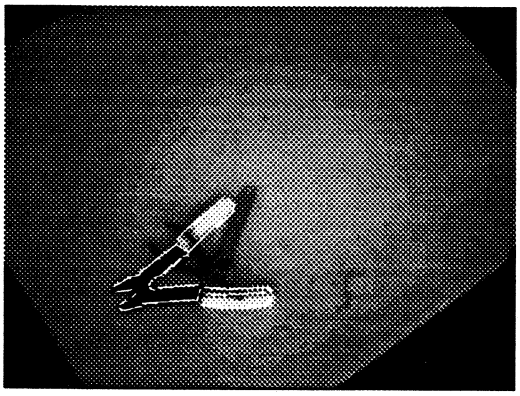

Figure 7.6: The model line segments, which are transformed with the optimal parameter found by the matching algorithm, are overlayed on the scene image. It shows that our algorithm has successfully located the model object in the scene.

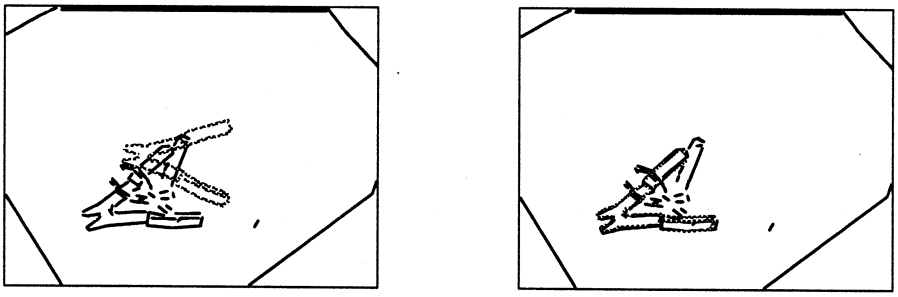

Figure 7.7: Shows how the model line segments (gray) and the scene segments (black) are matched. The model line segments, which are transformed with the optimal parameter found by the matching algorithm, are overlayed on the scene line segments with which they are matched. Most of the the endpoints and the lengths of the line segments are different. Furthermore, one long segment frequently corresponds to several short ones. However, our matching algorithm is robust enough to uncover the underlying rigid transformation from the incomplete and ambiguous data.

If the robust IRLS algorithm (6.7) and the continuation method in Section 7.3 are used to minimize (7.22), we need to solve a doubly-weighted absolute orientation problem

$$\sum_{ia} A_{ia}^{(k)} \|R\mathbf{x}_i + \mathbf{t} - \mathbf{y}_a\|^2 \tag{7.23}$$

at the $(k+1)$th iteration.

We extend the dual quaterion algorithm described in Appendix A to solve for the correspondence as well as the rotation and the translation.

Replacing the single sums over i with the double sum over both i and a, and the weight w_i with the correspondence process A_{ia}, the objective function becomes

$$f(\boldsymbol{r}, \boldsymbol{s}) = \sum_{ia} A_{ia} \|W(\boldsymbol{r})^t Q(\boldsymbol{r}) \boldsymbol{x}_i + 2W(\boldsymbol{r})^t \boldsymbol{s} - \boldsymbol{y}_a\|^2. \tag{7.24}$$

The problem can be reformulated as minimizing

$$f(\boldsymbol{r}, \boldsymbol{s}) = \boldsymbol{r}^t C_1 \boldsymbol{r} + \boldsymbol{s}^t C_2 \boldsymbol{s} + \boldsymbol{s}^t C_3 \boldsymbol{r} \tag{7.25}$$

subject to $\boldsymbol{r}^t \boldsymbol{r} = 1, \boldsymbol{s}^t \boldsymbol{r} = 0$, where

$$C_1 = -\sum_{ia} A_{ia} Q(\boldsymbol{y}_a)^t W(\boldsymbol{x}_i) \tag{7.26}$$

$$C_2 = \frac{1}{2} \sum_{ia} A_{ia} I \tag{7.27}$$

$$C_3 = \sum_{ia} A_{ia} (W(\boldsymbol{x}_i) - Q(\boldsymbol{y}_a)). \tag{7.28}$$

All the constraint information, including the current fuzzy estimate of the correspondence A is absorbed into the three 4-by-4 matrices C_1, C_2, C_3 in (2). The optimal solution $(\boldsymbol{r}^*, \boldsymbol{s}^*)$ can be found by using exactly the same method described in Appendix 2.3.

7.6.1 Experiments

A test instance for 3D point matching involves generating a random 3D point set as a model, and then generating a test scene by applying a random transformation,

adding noise consisting of independent Gaussian jitter and then randomly adding and deleting points.

A set of 20 3D points for $\mathbf{x}_i$ are generated uniformly within a box defined by $x_i,\ y_i,\ z_i \in [-5, 5]$. In order to generate a 3D rotation R, a unit quaterion is uniformly selected from a unit 4-sphere. The resulting distribution of 3D rotations is also uniform [14]. For translation $\mathbf{t}$, t_1 and t_2 are uniformly selected from $[5, 15]$, and t_3 from $[20, 50]$. Gaussian noise $\mathcal{N}(0, \sigma)$ is added to three component of each $R\mathbf{x}_i + \mathbf{t}$ to generate $\mathbf{y}_i$. The variance σ is related to SNR_{mod} by $\text{SNR}_{mod} = -20\log(\sigma/10)$ dB. A fraction ($= PO_m\%$) of model points are missing, and a fraction ($= PO_s\%$) of spurious points are added to scene points. The objective then is to recover the translation and the rotation and to find the correspondence between this and the original point set. The results are summarized in Figures 7.8 and 7.9. For each combination of (σ-PO_m-PO_s) 250 test instances were generated.

Figure 7.8 shows the results of Experiments **C1** and **C2** with different assumed knowledge of correspondence. When the correspondence is assumed to be known exactly (even if some of the points are really outliers), we use the standard absolute orientation algorithm. When there are potential outliers presents, we use the robust algorithm described in Chapter 6. For the case that we do not have any information about the correspondence, we apply the point matching algorithm described in this section.

The matching algorithm can not be better than standard algorithm when the correspondence information given to the standard algorithm is correct, as for the case in Figure 7.8 *left* (Experiment **C1**). However, we find that the performance of the matching algorithm is very close to the other two algorithms, especially when the signal-to-noise ratio is high. This shows that the matching algorithm solves the correspondence well. Figure 7.8 *right* reports the case that there are some outliers in the given correspondence. The results produced by the standard algorithm becomes useless, while the matching algorithm performs well when the percentage of outliers is below 10%. The reason that the robust algorithm outperforms the matching algorithm

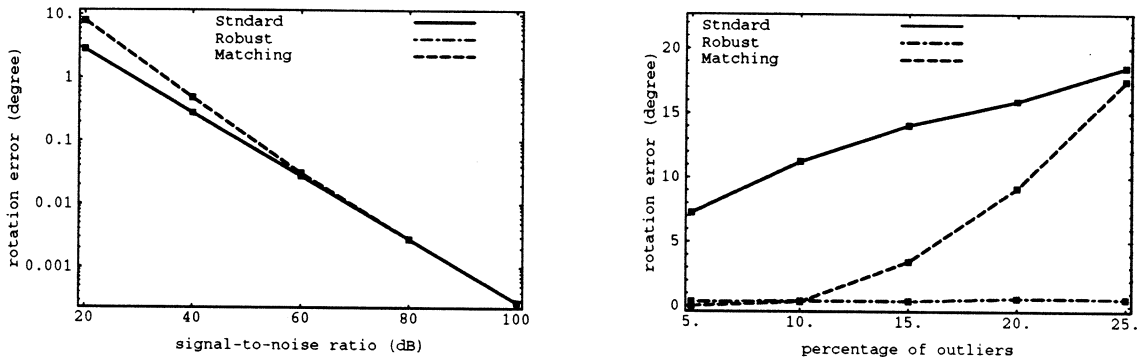

Figure 7.8: Results of Experiments **C1** (*left*) and **C2**(*right*) with different assumed knowledge of correspondence: totally known, partial known (having potential outliers), and unknown. Only rotation error is shown.

is that the given correspondence is not completely wrong.

In the case that the correspondence is completely unknown, the standard and the robust algorithm fail most of the time given an arbitrarily selected correspondence, unless the former gets the correct one, and the latter get one that is close enough. On the other hand, Figure 7.9 shows that the matching algorithm degrades gracefully when the percentages of missing model points and spurious scene points increase.

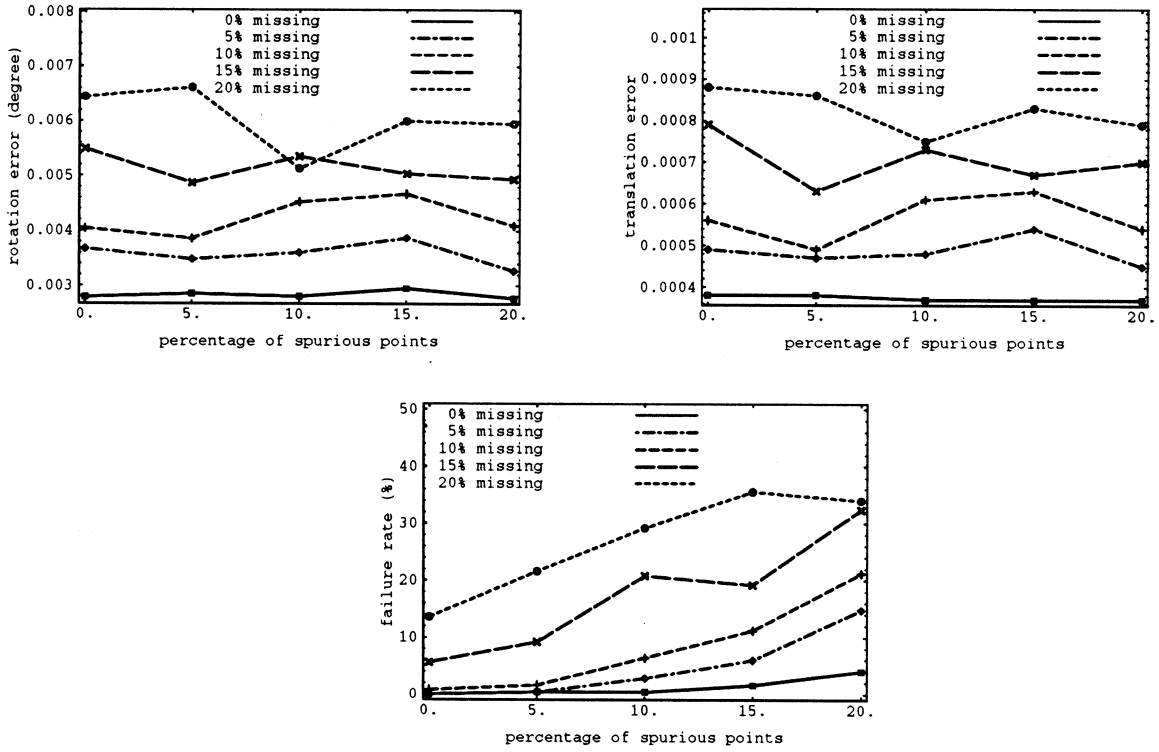

Figure 7.9: Results of 3D-3D point matching. The rotation and translation errors are averaged over correct matches. No initial guess using correspondence processes is given to the point matching algorithm.

Chapter 8

Conclusion and Future Work

8.1 What Has Been Done

We have presented a new framework and new algorithms for online pose estimation and model matching. It combines a descent method that exhibits both rapid convergence and robustness and elegant process-based algorithms for outlier rejection and model matching.

Unlike traditional approaches, our methods operates in 3D object space, not 2D image space, together with a systematic error measurements and propagation framework. This turned out to be the key point to the success of our methods. It is suited to applications where initial pose is not available such as visual-servoing and model-based recognition. The method's efficiency also makes it a good candidate for real-time model-based tracking of 3D objects.

The new outlier rejection and model matching framework and algorithms are especially suited to online applications. The matching algorithm gradually improves the solutions, so that when it is stopped in the middle, the result may still provide useful information.

In the following, we summarize the key ingredients of our work presented in this dissertation.

Minimum variance estimation and error fusion The alternating subspace minimization method and the 3D object space approach cannot be successful without appropriate error measurements and propagation. Based on the minimum variance estimation framework, we propagate the error measures in the observations to the estimations. For the alternating subspace minimization method which operates in two separate phases, the subset of variables computed in each phase are treated as observations in the other phase. The algorithm uses error measures in each phase to appropriately weight the "intermediate" observations.

Alternating subspace minimization The subspace minimization method provides an economical computational mechanism to optimize an objective function over variables as numerous as the observations themselves. It divides a large problems into subproblems with fewer variables. When each subproblem can be solved easily by itself, we can solve them alternatively. In our case, the pose estimation problem has been decomposed into an absolute orientation problem and iteratively linearized 3D reconstruction. Both have simple and efficient solutions.

Outlier and correspondence processes Outlier and correspondence processes model the observations using simple a Gaussian distribution with additional "processes" to model the connection of the data to the underlying Gaussian distribution. With this formulation, continuation methods are fully exploited, and interactions between processes are facilitated to solve the difficult problems of outlier rejection and correspondence establishment.

8.2 Future Work

In this section, we discuss some possible extension to the work.

Recursive motion estimation Our pose estimation algorithms can be very easily modified to do structure-from-motion, in which the structure reconstructed using previous frames are used as the observed model. In particular, our methods always attach the solutions with error measures in the form of covariance matrices. Such error measures can be propagated to next structure-from-motion computation resulting in a Kalman-like temporal filtering scheme.

Model-based visual tracking Model-based visual tracking involves tracking a moving object with known geometry through a video sequence on which the object is captured. The pose of the object must be constantly updated and used by a feature tracker to capture the object in the next frame. Preliminary work has been done in applying the proposed pose estimation method to track an object with a video camera in real-time.

Locating and tracking articulated objects We believe that the basic pose estimation for simple rigid objects can be extended to articulated objects such as human bodies in vision-based human-computer interactions. An articulated object can be represented by an *affixment graph*, in which the nodes representing individual rigid components of the articulated object are interconnected through the nodes representing relative positions and orientations between connecting components. As argued in [46], estimating all inter-component rigid transformations is no more difficult than estimating the single pose.

Bibliography

[1] Y. I. Abdel-Aziz and H. M. Karara, *Direct linear transformation into object space coordinates in close-range photogrammetry*, Symposium on Close-Range Photogrammetry (Urbana-Champaign, IL), Jan 1971, pp. 1–18.

[2] T. D. Alter, *3D pose from corresponding points under weak-perspective projection*, Tech. Report A.I. Memo No. 1378, MIT Artificial Intelligence Lab., 1992.

[3] T. W. Anderson, *Introduction to multivariate statistical analysis*, second ed., John Wiley and Sons, 1984.

[4] K. S. Arun, T. S. Huang, and S. D. Blostein, *Least-squares fitting of two 3-D point sets*, IEEE Trans. Pat. Anal. Machine Intell. **9** (1987), 698–700.

[5] N. Ayache and O. D. Faugeras, *HYPER:A new approach for the recognition and positioning of two-dimensional objects*, IEEE Trans. Pat. Anal. Machine Intell. **8** (1986), no. 1, 44–54.

[6] M. Bajura, H. Fuchs, and R. Ohbuchi, *Merging virtual objects with the real world: Seeing ultrasound imagery within the patient*, Proc. SIGGRAPH, July 1992, pp. 203–210.

[7] M. Black and A. Rangarajan, *On line processes, outlier rejection, and robust statistics*, Tech. Report YALEU/DCS/RR-993, Department of Computer Science, Yale University, 1993, to appear in Intl. J. Computer Vision.

[8] A. Blake and A. Zisserman, *Visual reconstruction*, The MIT Press, 1987.

[9] R. H. Byrd and D. A. Pyne, *Convergence of the iteratively reweighted least squares algorithm for robust regression*, Tech. Report Technical Report No. 313, Department of Mathematical Science, The Johns Hopkins University, 1992.

[10] W.Z. Chen, U.A. Korde, and S.B. Skaar, *Position control experiments using vision*, Intl. J. Rob. Res. **13** (1994), no. 3, 199–208.

[11] N. Cui, J. J. Weng, and P. Cohen, *Recursive-batch estimation of motion and structure from monocular image sequences*, CVGIP: Image Understanding **59** (1994), no. 2, 154–170.

[12] D. DeMenthon and L. S. Davis, *Exact and approximate solutions of the perspective-three-point problem*, IEEE Trans. Pat. Anal. Machine Intell. (1992), no. 11, 1100–1105.

[13] M. Dhome, M. Richetin, J. Lapresté, and G. Rives, *Determination of the attitude of 3-D objects from a single perspective view*, IEEE Trans. Pat. Anal. Machine Intell. (1989), no. 12, 1265–1278.

[14] D. Kirk Ed., Graphics Gems III, 124–132, Academic Press, 1992, pp. 124–132.

[15] R. M. Haralick et. al., *Pose estimation from corresponding point data*, IEEE Trans. Sys. Man Cyber. **19** (1989), no. 6, 1426–1446.

[16] W. E. L. Grimson et. al., *An automatic registration method for frameless stereotaxy, image guided surgery, and enhanced reality visualization*, Proc. IEEE Conf. Computer Vision Pat. Rec., 1994, pp. 430–436.

[17] O. D. Faugeras and G. Toscani, *Calibration problem for stereo*, Proc. IEEE Conf. Computer Vision Pat. Rec., June 1986, pp. 15–20.

[18] O.D. Faugeras and M. Hebert, *The representation, recognition and locatng of 3-D objects*, Intl. J. Robotics Res. **5** (1986), no. 3, 27–52.

[19] M. Fischler and R. C. Bolles, *Random sample consensus: A paradigm for model fitting and automatic cartography*, Commun. ACM (1981), no. 6, 381–395.

[20] S. Ganapathy, *Decomposition of transformation matrices for robot vision*, Pattern Recognition Letters (1989), 401–412.

[21] S. K. Ghosh, *Analytical Photogrammetry*, Pregamon Press, New York, 1988.

[22] S. Gold, *Matching structural and spatial representations*, Ph.D. thesis, Yale University, 1995.

[23] R. Goldberg, *Pose determination of parametrized object models from a monocular image*, Image Vision Computing (1993), no. 1, 49–62.

[24] ———, *Constrained pose refinement of parametric objects*, Intl. J. Computer Vision (1994), no. 2, 181–211.

[25] E. Grimson and T. Lozano-Pérez, *Model-based recognition and localization from sparse range or tactile data*, Intl. J. Robotics Res. **3** (1984), 3–35.

[26] G. D. Hager, *Real-time feature tracking and projective invariance as a basis for hand-eye coordination*, Proc. IEEE Conf. Computer Vision Pat. Rec., IEEE Computer Society Press, 1994, pp. 533–539.

[27] G. D. Hager, G. Grunwald, and G. Hirzinger, *Feature-based visual servoing and its application to telerobotics*, DCS RR-1010, Yale University, New Haven, CT, January 1994, To appear at the 1994 IROS Conference.

[28] G. D. Hager, S. Puri, and K. Toyama, *A framework for real-time window-based tracking using off-the-shelf-hardware*, Tech. Report YALEU/DCS/RR-988, Department of Computer Science, Yale University, October 1993.

[29] R. M. Haralick, *Propagating covariance in computer vision*, Proc. IAPR Intl. Conf. Pattern Recognitions, 1994, pp. 493–498.

[30] R. M. Haralick, C. Lee, K. Ottenberg, and M. Nolle, *Analysis and solutions of the three point perspective pose estimation problem*, Proc. IEEE Conf. Computer Vision Pat. Rec., 1991, pp. 592–598.

[31] R. M. Haralick and L. G. Shapiro, Computer and Robot Vision, ch. 14, p. 132, Addison-Wesley Publishing Company, Reading, Massachusetts, 1993, p. 132.

[32] N. Hollinghurst and R. Cipolla, *Uncalibrated stereo hand eye coordination*, Tech. Report TR-126, Cambridge University, Dept. of Engineering, September 1993.

[33] R. Horaud, *New methods for matching 3-D objects with single perspective views*, IEEE Trans. Pat. Anal. Machine Intell. (1987), no. 3, 401–412.

[34] R. Horaud, B. Canio, and O. Leboullenx, *An analytic solution for the perspective 4-point problem*, Computer Vis. Graphics. Image Process (1989), no. 1, 33–44.

[35] B. K. P. Horn, *Closed-form solution of absolute orientation using unit quaternion*, J. Opt. Soc. Amer. **A-4** (1987), 629–642.

[36] B. K. P. Horn, H. M. Hilden, and S. Negahdaripour, *Closed-form solution of absolute orientation using orthonomal matrices*, J. Opt. Soc. Amer. **A-5** (198), 1127–1135.

[37] T. S. Huang and A. N. Netravali, *Motion and structure from feature correspondences: A review*, IEEE Proceeding **82** (1994), no. 2, 252–268.

[38] P. J. Huber, *Robust Statistics*, John Wiley and Sons, 1981.

[39] D. P. Huttenlocher and S. Ullman, *Recognizing solid objects by alignment with an image*, Intl. J. Computer Vision **5** (1990), no. 2, 195–212.

[40] ———, *Recognizing solid objects by alignment with an image*, Intl. J. Computer Vision **5** (1990), no. 2, 195–212.

[41] A. M. Jazwinsky, *Stochastic Processes and Filtering Theory*, The Academic Press, 1970.

[42] T. Kaunungo and R. M. Haralick, *Multivariate hypothesis testing for gaussian data: Theory and software*, Tech. Report ISL-TR-95-05, Intelligence System Lab. Dept. of Electrical Eng. University of Washington, 1995.

[43] S. M. Kiang, R. J. Chou, and J. K. Aggarwal, *Triangulation errors in stereo algorithms*, Proc. IEEE Workshop Computer Vision, 1987, pp. 72–78.

[44] Y. G. Leclerc, *Constructing simple stable descriptions for image partitioning*, Intl. J. Computer Vision (1989), no. 3, 73–102.

[45] R. K. Lenz and R. Y. Tsai, *Techniques for calibration of the scale factor and image center for high accuracy 3-D machine vision metrology*, IEEE Trans. Pat. Anal. Machine Intell. **10** (1988), no. 3, 713–720.

[46] D. G. Lowe, *Three-dimensional object recognition from single two-dimensional image*, Artificial Intelligence (1987), no. 31, 355–395.

[47] ———, *Fitting parametrized three-dimensional models to images*, IEEE Trans. Pat. Anal. Machine Intell. (1991), no. 5, 441–450.

[48] C.-P. Lu and E. Mjolsness, *Mean field point matching by vernier network and by generalized Hough transform*, Proc. World Congress on Neural Networks, 1993, pp. 674–684.

[49] ———, *Two-dimensional object localization by coarse-to-fine correlation matching*, Advances in Neual Information Processing Systems 6, 1993, pp. 985–992.

[50] J. M. McCarthy, *Introduction to Theoretical Kinematics*, The MIT Press, 1990.

[51] E. Mjolsness and W. L. Miranker, *Greedy Lagrangians for neural networks: Three levels of optimization in relaxation dynamics*, Tech. Report YALEU/DCS/TR-945, Department of Computer Science, Yale University, January 1993.

[52] Eric Mjolsness, *Bayesian inference on visual grammars by neural nets that optimize*, SPIE Science of Artificial Neural Networks, April 1992, pp. 63–85.

[53] Eric Mjolsness and Charles Garrett, *Algebraic transformations of objective functions*, Neural Networks **3** (1990), 651–669.

[54] H. P. Moravec, *Obstacle avoidance and navigation in the real world by a seeing robot rover*, Ph.D. thesis, Stanford University, 1980.

[55] Carsten Peterson and Bo Söderberg, *A new method for mapping optimization problems onto neural networks*, International Journal of Neural Systems **1** (1989), no. 1, 3–22.

[56] A. Rangarajan, S. Gold, and E. Mjolsness, *A novel optimizing network architecture with applications*, submitted to Neural Computation.

[57] G. H. Rosenfield, *The problem of exterior orientation in photogrammetry*, Photogrammetric Engineering (1959), 536–553.

[58] P. J. Rousseeuw and A. M. Leroy, *Robust Regression and Outlier Detection*, John Wiley and Sons, 1987.

[59] Petar D. Simic, *Statistical mechanics as the underlying theory of 'elastic' and neural optimisations*, Network **1** (1990), 89–103.

[60] E. H. Tompson, *The projective theory of relative orientation*, Photogrammetria (1968), 67–75.

[61] R. Y. Tsai, *An effecient and accurate camera calibration technique for 3D machine vision*, Proc. IEEE Conf. Computer Vision Pat. Rec., 1986, pp. 364–374.

[62] S. Ullman, *An approach to object recognition*, Tech. Report AI Memo No. 931, MIT, Artificial Intelligence Lab., 1987.

[63] M. W. Walker, L. Shao, and R. A. Volz, *Estimating 3-D location parameters using dual number quaternions*, CVGIP: Image Understanding **54** (1991), no. 3, 358–367.

[64] Z. Wang and A. Jepson, *A new closed-form solution for absolute orientation*, Proc. IEEE Conf. Computer Vision Pat. Rec., 1994, pp. 129–134.

[65] J. Weng, N. Ahuja, and T. S. Huang, *Optimal motion and structure estimation*, IEEE Trans. Pat. Anal. Machine Intell. **15** (1993), no. 9, 864–884.

[66] J. Weng, P. Cohen, and M. Herniou, *Camera calibration with distortion models and accuracy evaluation*, IEEE Trans. Pat. Anal. Machine Intell. **10** (1992), no. 14, 965–980.

[67] J. Weng, P. Cohen, and N. Rebibo, *Motion and structure estimation from stereo image sequences*, IEEE Trans. Robotics and Automation **8** (1992), no. 3, 362–382.

[68] S.W. Wijesoma, D.F.H Wolfe, and R.J. Richards, *Eye-to-hand coordination for vision-guided robot control applications*, Intl. J. Rob. Res. **12** (1993), no. 1, 65–78.

[69] Y. Yakimovsky and R. Cunningham, *A system for extracting three-dimensional measurements from a stereo pair of TV cameras*, Computer Graphics and Image Processing **7** (1978), 195–210.

[70] Alan L. Yuille, *Generalized deformable models, statistical physics, and matching problems*, Neural Computation **2** (1990), no. 1, 1–24.

Appendix A

Solving Absolute Orientation Using Dual Quaternions

The absolute orientation solution in [63] is briefly described as follows:

Let the rotation and the translation be represented by a dual quaternion $(\boldsymbol{r}, \boldsymbol{s})$, $\boldsymbol{r}^t\boldsymbol{r} = 1, \boldsymbol{r}^t\boldsymbol{s} = 0$, which corresponds to a screw coordinate transform [50]. Treating quaternions as 4-vectors, the rotation can be represented by a homogeneous transformation matrix $W(\boldsymbol{r})^tQ(\boldsymbol{r})$ and the translation by $2W(\boldsymbol{r})^t\boldsymbol{s}$, where $\boldsymbol{r} = (q_1, q_2, q_3, q_4)^t$ is the quaternion for representing rotation, and

$$W(\boldsymbol{r}) = \begin{pmatrix} q_4 & q_3 & -q_2 & q_1 \\ -q_3 & q_4 & q_1 & q_2 \\ q_2 & -q_1 & q_4 & q_3 \\ -q_1 & -q_2 & -q_3 & q_4 \end{pmatrix} \tag{A.1}$$

$$Q(\boldsymbol{r}) = \begin{pmatrix} q_4 & -q_3 & q_2 & q_1 \\ q_3 & q_4 & -q_1 & q_2 \\ -q_2 & q_1 & q_4 & q_3 \\ -q_1 & -q_2 & -q_3 & q_4 \end{pmatrix}. \tag{A.2}$$

The rotation R can be written in terms of q_1, q_2, q_3, and q_4 as

$$R = \begin{pmatrix} q_4^2 + q_1^2 - q_2^2 - q_3^2 & 2(q_1q_2 - q_3q_4) & 2(q_1q_3 + q_2q_4) \\ 2(q_1q_2 + q_3q_4) & q_4^2 - q_1^2 + q_2^2 - q_3^2 & 2(q_2q_3 - q_1q_4) \\ 2(q_1q_3 - q_2q_4) & 2(q_2q_3 + q_1q_4) & q_4^2 - q_1^2 - q_2^2 + q_3^2 \end{pmatrix}. \tag{A.3}$$

With the dual quaternion representation, the objective function becomes

$$f(\boldsymbol{r}, \boldsymbol{s}) = \sum_i w_i \| W(\boldsymbol{r})^t Q(\boldsymbol{r}) \boldsymbol{x}_i + 2W(\boldsymbol{r})^t \boldsymbol{s} - \boldsymbol{y}_i \|^2 \tag{A.4}$$

where $\boldsymbol{x}_i = (\tilde{\mathbf{x}}_i, 1)^t$ and $\boldsymbol{y}_i = (\mathbf{y}_i, 1)^t$ are the homogeneous coordinates of the model point $\tilde{\mathbf{x}}_i$ and the scene point $\mathbf{y}_i$, respectively. Using the properties that $Q(\boldsymbol{a})\boldsymbol{b} = W(\boldsymbol{b})\boldsymbol{a}$ and $Q(\boldsymbol{a})^t\boldsymbol{a} = W(\boldsymbol{a})^t\boldsymbol{a} = (\boldsymbol{a}^t\boldsymbol{a})I$, the objective function can be reformulated as

$$f(\boldsymbol{r}, \boldsymbol{s}) = \boldsymbol{r}^t C_1 \boldsymbol{r} + \boldsymbol{s}^t C_2 \boldsymbol{s} + \boldsymbol{s}^t C_3 \boldsymbol{r} \tag{A.5}$$

subject to $\boldsymbol{r}^t\boldsymbol{r} = 1, \boldsymbol{s}^t\boldsymbol{r} = 0$, where

$$C_1 = -\sum_i w_i Q(\boldsymbol{y}_i)^t W(\boldsymbol{x}_i) \tag{A.6}$$

$$C_2 = \frac{1}{2} \sum_{ia} w_i I \tag{A.7}$$

$$C_3 = \sum_i w_i (W(\boldsymbol{x}_i) - Q(\boldsymbol{y}_i)). \tag{A.8}$$

Using Lagrange multipliers for the constraints yields

$$f(\boldsymbol{r}, \boldsymbol{s}) = \boldsymbol{r}^t C_1 \boldsymbol{r} + \boldsymbol{s}^t C_2 \boldsymbol{s} + \boldsymbol{s}^t C_3 \boldsymbol{r} + \lambda_1(\boldsymbol{r}^t\boldsymbol{r} - 1) + \lambda_2 \boldsymbol{r}^t\boldsymbol{s}. \tag{A.9}$$

The optimal solution $(\boldsymbol{r}^*, \boldsymbol{s}^*)$ can be found by solving for the Lagrange multipliers $\boldsymbol{\Sigma}_1$ and $\boldsymbol{\Sigma}_2$. $\boldsymbol{r}^*$ is the eigenvector of

$$C_3^t C_2^{-1} C_3 - \frac{1}{2}(C_1 + C_1^t) \tag{A.10}$$

with the largest eigenvalue, and $\boldsymbol{s}^*$ is equal to $-2C_2^{-1}C_3\boldsymbol{r}^*$. $\boldsymbol{r}^*$ and $\boldsymbol{s}^*$ are then used to determined the rotation and translation.

Appendix B

Fitting Orthonormal Matrices

For any 3-by-3 matrix $M = (\mathbf{m}^{(1)}, \mathbf{m}^{(2)}, \mathbf{m}^{(3)})$, the closest orthonormal matrix R to M and the associated scale factor s can be found by minimizing

$$\|M - sR\|_F, \tag{B.1}$$

where $\|\cdot\|_F$ is the Frobenius norm. It can be rewritten as

$$\sum_{i=1}^{3} \|\mathbf{m}^{(i)} - sR\mathbf{e}^{(i)}\|^2, \tag{B.2}$$

where $\mathbf{e}^{(i)}$ is the ith column vector of 3-by-3 identity matrix. This problem is equivalent to solving a 4-point absolute orientation problem with an extra point correspondence $((0,0,0)^t, (0,0,0)^t)$ for ensuring zero translation.