DYNAMICALLY ESTIMATING MOBILE RANGE CAMERA POSE FROM INVARIANT FEATURE MEASUREMENTS

Duncan L. Baird

Department of Electrical Engineering
McGill University, Montréal

April 1994

A Thesis submitted to the Faculty of Graduate Studies and Research
in partial fulfilment of the requirements of the degree of
Master of Engineering

Abstract. Complex robotic tasks such as autonomous exploration and grasping demand the co-operation of sensors and actuators. In order to integrate sensor measurements and actuator control schemes we must determine the rigid body transformations that relate the native co-ordinate frames of these devices. Equivalently, we need to estimate the relative pose of sensors and actuators in the system.

We examine the problem of determining the pose of a robot-mounted range-finding camera, and present a class of solutions motivated by the idea that mobile camera calibration is best addressed by an ongoing dynamic estimation process. We use range measurements and known robot kinematics to provide the estimate of camera pose which is maximally consistent with the available data. Our scheme uses scene features that are often present in typical workcell scenes and that are easily and reliably extracted. We develop several formulations of the principle, and present experimental results for both simulated and real data sets.

Résumé. L'exploration autonome et la préhension, comme exemples de tâches robotiques complexes, requirent la coopération de plusieurs capteurs et de plusieurs moteurs. L'intégration de données visuelles dans le schéma de contrôle demande que l'on connaisse les transformations qui relient les coordonnées de base de chaque sous-système. Autrement dit, il faut estimer la pose relative des capteurs et manipulateurs composant le système.

Nous examinons l'estimation de la pose d'une caméra télémétrique manipulée par robot, et présentons une classe de solutions qui découlent de l'idée que la calibration est un procédé dynamique et continuel. Les mesures prises par la caméra et la connaissance de la cinématique du manipulateur permettent d'extraire estimation la plus consistante de la transformation caméra-robot. La méthode exploite des particularités géométriques qui sont présents dans la plupart des scènes typiques, et qui sont facilement identifiables. Nous développons plusieurs formulations du principe, et nous présentons les résultats d'essais avec des données simulées et réelles.

Acknowledgements

The author would like to acknowledge the valuable assistance provided by several individuals in the preparation of this work.

Gilbert Soucy and Pierre Tremblay created many of the tools used to control the robot and range-finder in our lab, and provided speedy technical assistance on numerous occasions. John Lloyd and Peter Whaite consistently provided keen insight and advice at critical moments, and their assistance has been invaluable. Prof. Vincent Hayward was patient and accommodating beyond all reasonable call as this thesis neared completion. Prof. Frank Ferrie offered thoughtful guidance and supervision, as well as timely encouragement when prospects looked grim. Lisa Ramsaran smoothed out a great deal of bad punctuation, and provided much-needed moral support.

Thank you.
dlb 94/04/19

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

Introduction

1. Motivation

Consider an image sensor, such as a conventional intensity camera or a range-finding sensor, mounted on the end effector of a mobile robot (Figure 1.1). The utility of this configuration lies in its ability to provide multiple views of a scene, which may yield information from object surfaces which would be occluded or otherwise ambiguous from a single viewpoint. In order to assimilate the information acquired from several viewpoints we need to map between the local co-ordinate frames of the sensor at each position. This requires knowledge of the relative motion between sensor positions; often we also find it useful to know the absolute sensor positions with respect to some global co-ordinate system.

In general we have some knowledge of end effector position from the kinematic model of the robot manipulator. The orientation of the sensor frame with respect to the robot end effector, often called the "hand-eye transformation", may not be known, and indeed it is often difficult to measure. This is largely because the camera co-ordinate system may be defined with respect to the camera's optical axes (which may be physically inaccessible and not indicated by any mechanical feature), or may be arbitrarily defined by a numerical calibration process. In this thesis we explore the problem of determining the end-effector-to-camera co-ordinate transformation for the case of a robot-mounted range-finding sensor, and develop a class of solution methods which address both the analytic and the practical difficulties inherent in the task.

Typically, the hand-eye calibration problem is regarded as a static computation that is performed once after assembly of a robotic system under highly constrained conditions and using optimally designed jigs or calibration targets. While this approach is expedient, it fails to address several important considerations that arise in a practical context. If the sensor is re-calibrated, removed and re-installed, or if its

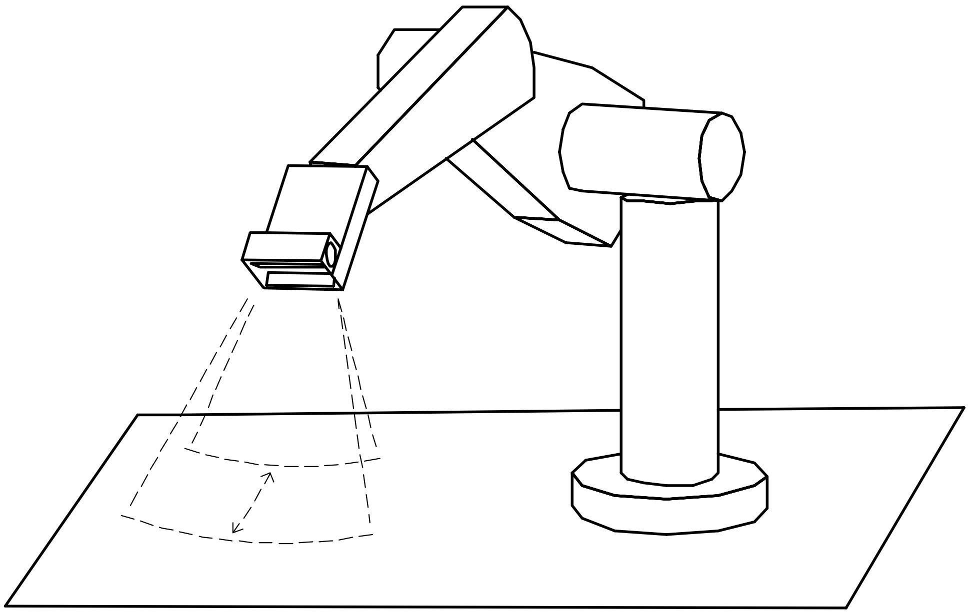

Figure 1.1. Industrial robot manipulating a scanning laser range-finding camera.

fixture to the robot changes in any way, the static calibration procedure must be repeated. This requires introduction of the calibration jig into the work environment and cessation of normal operations while the procedure is executed. Furthermore, the calibration generally requires that the robot kinematics be precisely known; in practice, positioning errors in industrial robots can be large, and are highly complex functions of configuration, static and dynamic loading conditions, and location in the workspace.

In this thesis we argue in favour of a dynamic calibration scheme. Such a procedure runs continuously during normal operation of the system, and uses data from the robot/sensor complex as well as feature correspondences indicated by higher level processes which either know or infer the structure of the environment. The premise of the calibration is to seek hand-eye transformation parameters which make measured data maximally consistent with known or inferred models of objects in the workspace. We cast the problem in a parameter estimation framework, and visualize the solution as a filter taking feature correspondence data as input and generating kinematic parameter estimates as output (Figure 1.2).

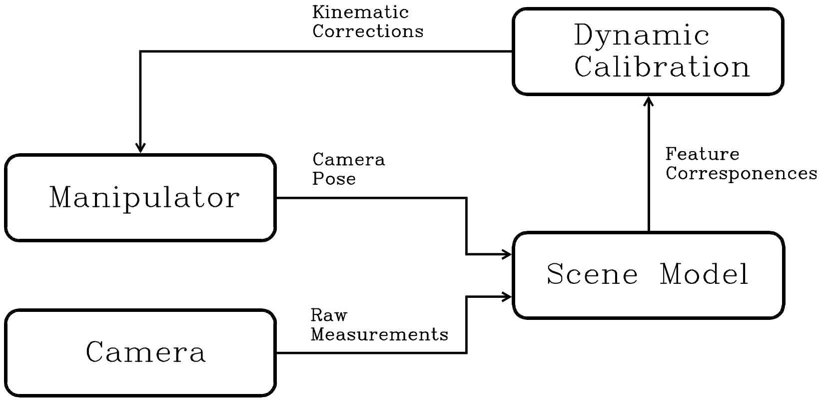

FIGURE 1.2. Dynamic calibration. Estimates of camera pose helps to establish feature correspondences between views, which in turn are used to refine the pose estimates.

2. Contribution

The major contributions of this thesis are (i) a novel dynamic formulation of the hand-eye calibration problem, and (ii) a demonstration that hand-eye calibration can be computed simultaneous with the execution of a robotic vision system's primary task. We select an invariant feature in the environment for which correspondence between views is easily and reliably established, and collect a sequence of views of that feature from different positions. We then compute the hand-eye transformation that is most consistent with the correspondence hypothesis, given the data currently available. Our approach is:

- dynamic, in the sense that it runs continuously and can adapt to changing conditions in the workspace
- capable of accommodating partial information, *i.e.* features that do not completely constrain rotations and translations in six degrees of freedom
- tolerant of errors in feature extraction and robot kinematic control
- economical in terms of its constraints on the structure of objects in the workspace, as well as in computational complexity
- applicable as a static calibration, requiring no special calibration jigs or manual measurements of the scene

Central to the practical utility of such a scheme is the selection of features which can be reliably estimated and tracked in typical task environments. We formulate our estimator for both point features and planar surfaces, as these represent two feature classes which are common and easily recovered in many real situations. In particular, we do not require that position of the invariant features be known in any global coordinate system, that more than one such feature be available in the environment, or that 6-DOF pose and position of an object be estimated from a single image frame.

3. Overview

We begin in Chapter 2 with a formal statement of our calibration problem, and with a discussion of the desired characteristics of the solution. We survey previous work of other researchers in this area, and explore the strong relationship of this problem to static camera calibration, robot kinematic calibration, motion estimation, and integration of multiple views.

In Chapter 3, we present a solution to the hand-eye calibration problem for mobile range-finding cameras based on correspondence of a single point feature across a sequence of views. The solution is a minimum mean squared error estimator that is linear in the data, but incorporates nonlinear orthonormality constraints. We also develop two solutions for the case where the invariant feature is a planar surface. The first of these follows directly from the point-feature development, but is based on a distance metric that exhibits several undesirable characteristics. The second formulation improves on the metric problem, but implies a somewhat more expensive computation. The formulation for plane features is particularly appealing in many practical contexts, since it is often very easy to find and extract an invariant planar surface (such as a floor or table-top) from the scene. In all cases, we emphasize computational methods which use finite memory resources, as this is an important characteristic of filter processes that run perpetually.

We demonstrate experimental results of the calibration processes in Chapter 4. We first assess accuracy and stability of the point-feature estimator using simulated data. Both plane-feature formulations are simulated in order to gain some insight into the impact of the metric problem inherent in the first estimator. Finally, we show the performance of the plane formulation with real data gathered from our experimental apparatus and compare with the simulated results.

In Chapter 5 we summarize the dynamic calibration technique proposed, and comment on several issues that remain for future study. We also remark briefly on other sensor calibration problems encountered in our laboratory which are solved by adaptations of our method.

CHAPTER 2

Problem Overview

Our discussion of the range-finding camera hand-eye calibration problem makes frequent reference to co-ordinate frames and transformations between frames, so we begin by introducing suitable notation. We then state our formulation of the calibration problem, and identify the desired characteristics of a solution. Finally, we survey the work of other researchers on this and other related problems, and gather a number of ideas that are useful in our development of solutions.

1. Background - Co-ordinate Frames and Transformations

A point in 3-space is represented by a 3-vector $\mathbf{v}_F$, where the components of $\mathbf{v}$ indicate displacements along three orthogonal basis directions provided by a co-ordinate frame F. The world co-ordinate frame W is arbitrarily defined to lie somewhere in space, and an un-subscripted vector $\mathbf{v}$ is conventionally assumed to be relative to W unless the context indicates otherwise.

A frame F is defined by the transformation $\mathcal{T}_{FW}$ which maps vectors in F into corresponding vectors in W, as $\mathbf{v}_W = \mathcal{T}_{FW}\mathbf{v}_F$. We interpret $\mathcal{T}_{FW}$ as an operator, and for the moment make no assumptions about its mathematical form or parameterization. Euler's theorem indicates that for rigid objects and motions the transformation can always be decomposed as a rotation of basis and a translation, which implies that $\mathcal{T}_{FW}$ expresses the position and orientation of F with respect to W.

It is convenient to generalize the notion of co-ordinate transformations to objects other than points in space. A parametric object such as a geometric solid, surface or direction in space is described with respect to frame W by a vector of its parameters $\mathbf{f}_W$. Its representation in another frame F is given by $\mathcal{T}_{WF}\mathbf{f}_W$, where we understand the operator $\mathcal{T}_{WF}$ to apply a transformation appropriate to the parameterization of $\mathbf{f}_W$. No ambiguity results as long as we restrict our analysis to a single object class.

We also introduce a natural notation for compositions and inverses of co-ordinate transformations. Given frames A, B and C, we have

$$\begin{aligned}\mathcal{T}_{AC} &= \mathcal{T}_{BC}\mathcal{T}_{AB}\\ \mathcal{T}_{AB} &= \mathcal{T}_{BA}{}^{-1}\end{aligned}$$

2. Robot Vision Workcell

While the calibration problem we describe is of a quite general nature, for clarity and concreteness we pose it in the context of the robot vision workcell found in our laboratory. Our workcell consists of an optical range-finding sensor mounted on the end-effector of a 6-DOF robot manipulator, as in Figure 1.1. Referring to Figure 2.3, we define the relevant co-ordinate frames and transformations.

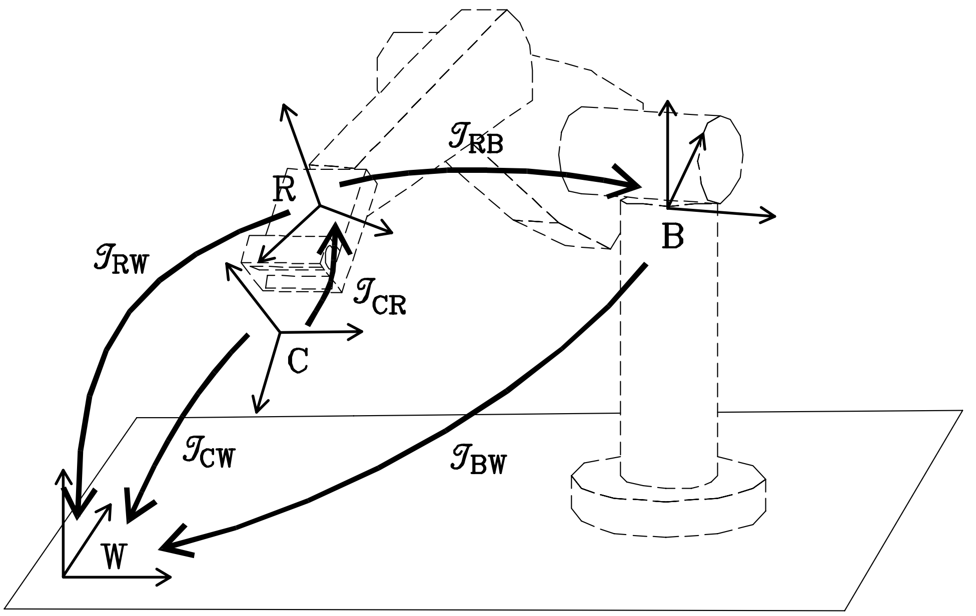

FIGURE 2.3. Coordinate frames and kinematic loops of the robot/range-finding camera complex.

2.1. Robot.

The robot base frame B is fixed with respect to W, and the transformation $\mathcal{T}_{BW}$ is assumed to be known. In general we are free to define W, and in much of this thesis find it convenient to identify W with B, in which case $\mathcal{T}_{BW}$ is the identity. Relationships between the coordinate frames of the robotic workcell are shown schematically in Figure 2.4.

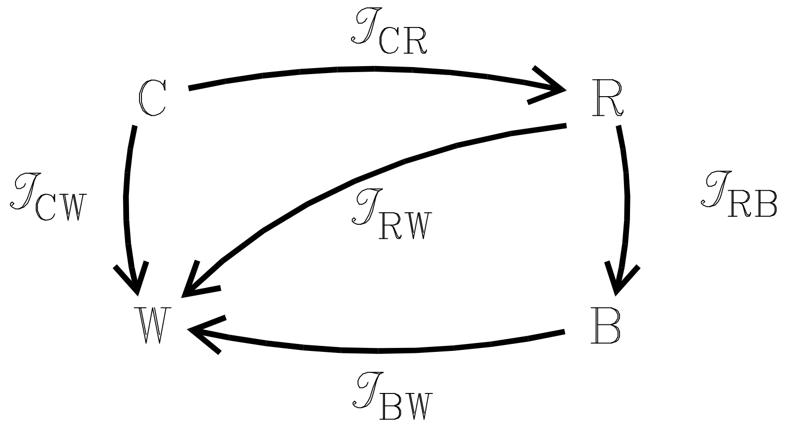

Figure 2.4. Schematic view of the workcell kinematic loops.

The robot end effector has a rigidly attached co-ordinate frame R. The position and pose of the end effector with respect to robot base $\mathcal{T}_{RB}$ can be computed for any robot position from the known robot kinematic model and measured joint encoder values [22]. In practice we control the robot by specifying values of $\mathcal{T}_{RB}$, and rely on robot control software to compute and achieve the required joint angles. We note that the measured quantity $\overline{\mathcal{T}_{RB}}$ may differ somewhat from the true value $\mathcal{T}_{RB}$, due to errors in the kinematic model and to un-modelled loading effects. The resolution of kinematic model errors is itself a significant problem in robotics [11, 24], and for our purposes we merely acknowledge that $\overline{\mathcal{T}_{RB}}$ is corrupted by a noise process.

2.2. Range-finding Camera. The range-finding camera measures distance to the nearest opaque surface along a ray emanating from its optical centre. The ray is swept in two axes, and a two-dimensional depth map of the scene is captured. Each pixel of the range image identifies the (x, y, z) co-ordinates of a point that lies in the surface of a scene object. These co-ordinates are expressed with respect to the camera's intrinsic co-ordinate frame C, which may be aligned with the optical axes of the camera itself, or may be completely arbitrary with respect to camera geometry.

Note that the camera measures radial distance from its optical centre to a surface, and that these measurements are mapped onto C by image acquisition software. The determination of this mapping is called intrinsic calibration [32], as it depends solely on optical and mechanical characteristics of the camera itself. We assume that an intrinsic calibration of the camera has been obtained previously by a static procedure, and we do not consider it further.

$\mathcal{T}_{CW}$ gives the location of the camera in world co-ordinates. In order to merge image data acquired from several robot positions $\mathcal{T}_{RBi}$, we need to know the actual camera positions $\mathcal{T}_{CWi}$ in the world frame [1]. This can be computed from the kinematic loop of Figure 2.4 as

$$\mathcal{T}_{CWi} = \mathcal{T}_{BW}\mathcal{T}_{RBi}\mathcal{T}_{CR} = \mathcal{T}_{RWi}\mathcal{T}_{CR} \tag{1}$$

where $\mathcal{T}_{CR}$ is the constant transformation relating the camera frame C to the robot end effector R. Since $\mathcal{T}_{RWi}$ can be obtained from the robot controller and the known transformation $\mathcal{T}_{BW}$, our problem is to determine $\mathcal{T}_{CR}$ from some appropriate measurement or series of measurements.

3. Problem Statement

We typically use an estimate of $\mathcal{T}_{CR}$, along with measurements of robot end-effector position, to map camera measurements from different view stations into a common reference frame W. It is therefore natural to formulate the estimator for $\mathcal{T}_{CR}$ to minimize the error in this mapping over a typical sample of measurements. In other words, we seek the estimate of $\mathcal{T}_{CR}$ which is *maximally consistent* with all observations. We can now state our approach to the hand-eye calibration problem as follows.

Given the kinematic configuration of Figure 2.4, we select a measurable invariant feature $\mathbf{f}_W$ in the workspace, some or all of the parameters of which may be unknown. Define a metric $\delta(\mathbf{f},\mathbf{g}) \geq 0$ which measures the distance between two features $\mathbf{f}$ and $\mathbf{g}$, where the notion of distance is appropriate to the feature class. Then we seek minimum mean-square error (MMSE) estimates $\overline{\mathcal{T}_{CR}}$ and $\overline{\mathbf{f}_W}$ which satisfy

$$\min_{\mathcal{T}_{CR}\in\Omega,\mathbf{f}_W} \sum_{i=1}^{n} w_i\delta^2(\mathcal{T}_{RWi}\mathcal{T}_{CR}\mathbf{f}_{Ci},\, \mathbf{f}_W) \tag{2}$$

[1] This is equivalent to knowing relative camera motions if we define the world co-ordinate frame to coincide with one of the camera positions.

Here, Ω is the set of rigid motion transformations, $\mathbf{f}_{Ci}$ is the feature measurement acquired from the i^{th} viewpoint, w_i is a weight reflecting confidence in the data acquired at viewpoint i, and n is the total number of available views. Note in particular that we minimize not only over transformations $\mathcal{T}_{CR}$, but also over parameters of the invariant feature $\mathbf{f}_W$.

This analytic statement alone does not completely characterize our calibration problem. We must identify several practical and computational goals that motivate our development of estimators for $\overline{\mathcal{T}_{CR}}$ and $\overline{\mathbf{f}_W}$.

(i) **Dynamic calibration**

We seek a calibration process that can run continuously during execution of robot vision system tasks, which may be inspection, exploration, grasping, or similar actions. We therefore do not assume that we have direct control over the placement or selection of objects in the workspace, or over the trajectory of the robot. Our procedure should provide the best possible estimate of the unknowns based on whatever data is available.

(ii) **Detectable features**

Invariant feature objects used by the calibration scheme should be chosen such that extraction from a range image and detection of correspondence across views can be performed easily and reliably. In particular, we would like to utilize feature objects that are likely to be found in the task environment, rather than to introduce to the scene a dedicated calibration target.

(iii) **Observability**

Observability of our system is equivalent to the requirement that the measurements we take be sufficient to constrain the estimated parameters. We must ensure that the chosen feature classes are sufficiently descriptive to permit observation of all modes of $\mathcal{T}_{CR}$. We show later that the selection of manipulator positions also governs observability. Since manipulator motions are directed primarily by task objectives (and not by the calibration process), we must detect conditions under which observability is marginal or totally lacking.

(iv) **Finite memory**

Computationally, we seek an estimator having the behavior of a digital filter.

Each time a new view of the reference feature is obtained, we compute an updated estimate for the unknowns based on the entire data set. In practice this demands that the estimator be implementable in finite memory, and places some strong constraints on admissible solution methods.

(v) **Physically realizable transforms**

The fact that our kinematic equations arise from rigid spatial relations implies that the estimate of $\mathcal{T}_{CR}$ should correspond to a physical rotation and translation of frames. Many useful parameterizations of co-ordinate transformations have more than the six degrees of freedom consistent with rotation and translation, and admit solutions that fall outside the space of physically realizable transformations. While we should ensure that our estimator produce physically realizable transformations, we believe that a non-realizable solution that is significantly better (in the sense of our projection error metric) than the best realizable one is useful in practice and may provide insight into other errors in the kinematic loop (*i.e.* static calibration of the range-finder and robot manipulator).

Having sufficiently motivated our efforts, we now turn to the literature and examine previous work on related problems.

4. Related Work

4.1. Representing Rotations. One of the significant issues arising in the development of our computational procedure is the selection of an appropriate representation for rigid co-ordinate transformations, and in particular for the rotation part. Spring provides a useful overview of this issue in [29], and we quickly summarize the high points here.

The rotation of a column 3-vector, $\mathbf{v}$, is a linear, length- and angle-preserving transformation that can be expressed as matrix multiplication by a 3×3 real orthonormal matrix R_{AB}, as $\mathbf{v}_B = R_{AB}\mathbf{v}_A$. This is a convenient representation for computing rotations, but is inappropriate as a parametrization of rotation because the number of parameters (9) exceeds the number of degrees of freedom in 3-space rotation (3). The column vectors of a rotation matrix R_{AB} are the unit basis directions of B expressed in A. The components of a rotation matrix satisfy six implicit constraint equations which enforce unit length and mutual orthogonality on these basis vectors.

General rotations may be parameterized by triples of Euler angles. The three parameters of a Euler angle representation specify successive rotations about defined basis directions. While the number of parameters equals the number of degrees of freedom in rotation, Euler angle representations suffer several limitations. There exist many different and incompatible Euler parameterizations according to the selection of rotation axes and the order of successive application, so it is important to specify which member of this family of representations is intended. The conversion of Euler angles to rotation matrices requires evaluation of trigonometric functions, which are computationally expensive and introduce analytic nonlinearity. Finally, Euler representations have singular configurations in which a set of representations are equivalent, which can introduce computational difficulties.

A representation for rotations which has received much attention in the literature is the quaternion, and in particular the unit quaternion [29, 27, 12, 41]. The quaternion is a 4-vector which can be viewed as the composition of a scalar and a 3-vector. The direction of the 3-vector identifies the axis of rotation, and the scalar encodes the magnitude of rotation about that axis. There exists a particularly elegant algebra for the manipulation of quaternions [10], and it is finding increasing application in both analytic and computational work in computer graphics, machine vision and robotics. Quaternions do, however, have several limitations. Use of a 4-vector still represents an over-parameterization of rotation, which often motivates a restriction to unit quaternions. Rotation of a vector is a nonlinear operation requiring two quaternion multiplications. Despite this, it is clear that quaternion algebra is a powerful analytic tool, and arguably provides the most natural mathematical framework for manipulating rotations.

4.2. Integrating Multiple Views. The construction of descriptive models of the environment from sensor data generally entails the need to merge data acquired from multiple viewpoints. This is because a single viewpoint rarely provides sufficient information to fully constrain the model building process. Successful view integration typically requires that registration of the image coordinate systems be accomplished with very high precision. The degree of kinematic precision available from many manipulator platforms (including both fixed-base industrial robots and free-roving mobile systems) is often inadequate to provide adequate registration based purely on sensor/manipulator calibration, and we must resort to refinements based on feature matching between views. Much work has been done on this problem; a review can be found in [28]. In general, establishing correspondence over a dense feature set is a complex optimization problem with many local minima, and the quality of the result is therefore highly dependent on having a good initial estimate of relative camera motion. Our aim in the present work is not to eliminate the need for motion parameter refinement based on local feature correspondences, but to ensure that the best possible initial estimate is provided to this process by the camera positioning complex.

4.3. Estimating Orientation in Photogrammetry. The problem of determining unknown solid rotations and translations in three dimensions was explored early on by the photogrammetric community, and their work significantly predates the recent contributions of robotics and computer vision researchers. Their problem generally involved estimating the relative position and pose of two airborne cameras (or a single moving camera), as well as the orientation of a multi-camera system in space. These are respectively called problems of relative and absolute orientation [31], although analytically these problems are equivalent (see Section 2.2).

Solutions to absolute orientation have been proposed by many authors. Given a number of corresponding points in two frames, it is straightforward to solve for the translation between frames, as well as an optional scale factor [12]. Estimation of the rotational component is less obvious. Thompson [30] and Schut [25] obtain exact rotation parameters from three pairs of point correspondences and offer no means of incorporating more data in the formulation. Their methods are computationally expedient, but implicitly assume the data to be error-free. Furthermore, these methods use the data in an asymmetric manner, so that in the case of noisy data it is possible to obtain different results simply by re-ordering the data points.

Oswal and Balasubramanian [19] present a solution that accommodates redundant measurements, but which generates a rotation matrix that is not orthogonal. Their approach obtains the nine components of a 3×3 rotation matrix by solution of a set of linear equations. The authors attempt to orthogonalize the solution, but their method results in neither an orthogonal matrix nor a minimum error solution to the original estimation problem. Despite this, we can make two useful observations about their method:

- the over-parameterization of a rotation (3-DOF) by the components of a 3×3 matrix (9-DOF) is computationally convenient, because it results in a completely linear formulation
- if measurement errors are independent, identically distributed, normal and zero-mean, then as more data is collected the estimate will converge to the true, orthogonal, rotation matrix.

B.K.P. Horn solves absolute orientation in closed form using unit quaternions [12], and with orthonormal rotation matrices [13]. Both methods directly yield real rotations that solve the over-constrained case (many point correspondences) in a minimum mean-squared error sense, by solution of a low-order eigenvector problem. In each method, a minimum of three point correspondences are required between two views. Horn's solutions have the desirable properties that all data points contribute equally to the solution (unlike methods which imply an ordering of the data points), and that the solution is symmetric with respect to interchange of the two views.

Note that the estimation of absolute orientation from point matches is useful to us in two ways. First, by providing the camera motion parameters directly, it permits integration of data from multiple views without recourse to measured joint positions and kinematic models of the robot, or indeed to the camera hand-eye transform. While this is highly useful, we also need a method to establish camera orientation in the common case where scene ambiguity denies us the necessary point correspondences. Second, an estimate of relative camera motion between two views can be used in conjunction with an estimate of robot end effector motion, in order to determine the camera hand-eye transform. However, the solution for camera orientation is but one step of this nontrivial process, which we discuss in Section 4.5.

4.4. Motion Estimation. The problem of motion estimation posed by the computer vision and robotics community is basically the same as that of orientation in photogrammetry: given two views of a scene, extract the motion parameters of the camera. Equivalently, we may assume the camera to be stationary and objects in the scene to be in motion. The problem is generally approached by establishing point matches between image frames, and computing the rigid transformation which best maps between correspondences. The nature of solutions to this problem depend strongly on the projection model of the camera employed.

Motion estimation under perspective projection has been studied at length [6]. Characterization of conditions under which the problem can be solved and the number of point correspondences required is rather involved. A number of authors consider the case of orthographic projection [35, 15, 14], where the results are somewhat more straightforward. It is understood that an orthographic projection model may be a suitable substitute for perspective in the case of intensity cameras with long focal lengths. The extension of our formulation for range-finding cameras to intensity cameras is of some interest, but is beyond our present scope.

In three-dimensional motion estimation the measurement source is a range-finding camera or stereo-vision system, or in general a device capable of delivering point measurements in 3-space. Many authors have considered this problem (see [4] for a review). Estimation of an orthonormal matrix which best maps one set of vectors into another in the least squares sense is known as the *orthogonal Procrustes problem*, and is solved directly using the singular value decomposition [9]. This solution implicitly models errors as being confined to one view. When both views are corrupted by noise (which is generally the case), a solution based on the method of *total least squares* (TLS) [9, 8] is more appropriate, and is studied in [4]. It should be noted that the TLS solution generates a rotation matrix that is orthogonal, but not orthonormal.

4.5. Camera Hand-eye Calibration. At this point we have acquired a battery of methods for establishing camera motion parameters from point correspondences in two views. Several methods for computing the hand-eye calibration of a mobile camera are built on this procedure, as we show presently.

Suppose that we have captured two views of static scene, and have successfully established the camera motion by one of the methods of sections 4.3 or 4.4. Referring to Figure 2.5, we can write the closed kinematic loop equation

$$\mathcal{T}_{C_1 C_2}\mathcal{T}_{CR} = \mathcal{T}_{CR}\mathcal{T}_{R_1 R_2} \tag{3}$$

This expression is obtained by equating transformations from camera frame C_1 to robot frame R_2 via different paths around the kinematic loop. We are interested in solutions $\mathcal{T}_{CR}$ satisfying (3), given that we can measure the camera motion $\mathcal{T}_{C_1 C_2}$ as above, and obtain the robot manipulator motion $\mathcal{T}_{R_1 R_2}$ from known robot kinematics and joint measurements. This problem is addressed by Shiu and Ahmad [26] using homogeneous representations of rigid motion, and rather more directly by Zhuang and Roth [41] using a quaternion formulation. This provides another compelling example of the power of quaternion representations, for their application results in a far simpler presentation, as well as additional insight that was not forthcoming in the homogeneous transform representation. Both authors conclude that two pairs of views satisfying some weak conditions are required for a unique solution to $\mathcal{T}_{CR}$. Zhuang and Roth further point out that their method is easily extended to accommodate over-constrained data in a least-squares framework.

Angeles [1] presents a solution based on vector invariants of rotation matrices, which he applies to the hand-eye calibration problem. The solution uses Gramm-Schmidt orthogonalization and results in an extremely simple computational procedure. While the solution is exact and makes no attempt to account for measurements corrupted by noise, it appears that extension of this methodology to over-constrained data should be straightforward.

Tsai and Lenz propose a similar solution to the hand-eye problem [33, 34] as part of a series of camera calibration solutions for mobile CCD cameras [32, 16, 34]. A linear algebraic equation which must be satisfied by transformations between pairs

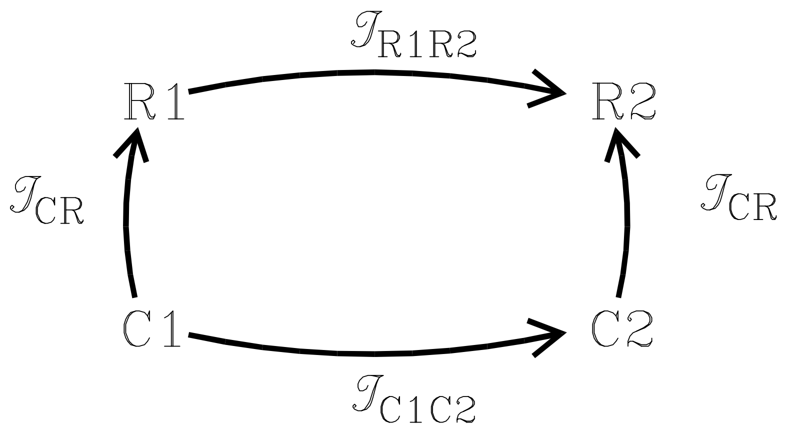

Figure 2.5. Kinematic loops for two camera positions $C1$ and $C2$. The relative end effector motion $\mathcal{T}_{R1R2}$ is known from robot kinematics, and the relative camera motion $\mathcal{T}_{C1C2}$ is the displacement of the feature in camera co-ordinates between the two views.

of viewpoints is derived, and a system of several such pairs (at least two) is solved for the minimizing rotations and translations comprising $\mathcal{T}_{CR}$. The authors provide a thorough analysis of error propagation and show quite impressive practical results. However, since pairs of camera stations comprise a datum in their minimization framework, it is necessary to consider all combinations of view stations in order to fully utilize the information present in the data set (pairing only of stations adjacent in time does not completely utilize the data). Thus it is necessary to store the data from each view independently, and the memory requirement of the procedure grows with time.

We should point out that all of these procedures require that the position and pose of the camera with respect to a fixed calibration target or a previous camera station has been obtained for each view. This requires either the solution of a motion estimation problem (sections 4.3 and 4.4), or the use of a known target from which camera position and pose can be estimated unambiguously. In [34], a precision optical target is employed, and the solution of camera orientation relative to the target is given in [32] as part of the calibration of intrinsic camera parameters.

4.6. Robot Kinematic Calibration. We digress briefly to explore work in the area of robot kinematic calibration. All attempts to estimate $\mathcal{T}_{CR}$ using computed end-effector orientations are subject to errors in the robot kinematic model, so some attention to this issue is called for. Furthermore, robot calibration bears several interesting similarities to our camera calibration problem.

The forward kinematics of a serial link manipulator are conveniently represented by a composition of co-ordinate transformations between successive links. Each transformation is commonly parameterized in terms of Denavit-Hartenberg (D-H) parameters [5, 22], which represent two rotations and two translations for each link. In the case of a revolute joint, one of the angular D-H parameters is controlled and measured via a joint encoder, while the others are fixed design parameters. These values are subject to error from a variety of sources: machining and assembly tolerances, gear backlash, and link flex due to static or dynamic loading. The compound effect of small errors can be quite significant in a serial manipulator, and the highly non-linear nature of the error model has given rise to a significant research problem. Much of the work in this area is reviewed in [11].

While we are not directly concerned with the correction of robot kinematic errors, we have noted two approaches to this problem which are of interest. Bennett and Hollerbach [3] propose a methodology for self-calibration of a redundant serial manipulator. A robot with greater than six degrees of freedom may be formed into a mobile closed kinematic loop by constraining its end-effector to be fixed with respect to its base. By exercising the robot joints and recording all joint measurements, it is possible to recover the kinematic corrections by nonlinear optimization. We observe that a range-finding camera mounted on the end-effector of a 6-DOF robot from which we can determine position and pose of a fixed calibration target can be viewed as a single redundant manipulator with end-point constraints, where the camera-target transformation provides the parameters of the final link. Thus we expect that we could calibrate the entire robot/camera complex following their approach, provided that good initial estimates of the parameters (including the unknown $\mathcal{T}_{CR}$) are available.

A less rigorous approach to kinematic correction is proposed by Foulloy and Kelly [7]. They assume that within a small volume of the workspace, end-effector position errors can be adequately modeled as a homogeneous linear function of measured position. This is a rather doubtful assumption, for they describe robot end-effector position with a 6-vector composed of Euler angle and translation components, and compute corrections using linear combinations of these elements. There is no reason to believe, for instance, that the error in a rotational parameter should be a linear function of its mean value, which is measured with respect to a completely arbitrary zero.

Following their line of thought, however, it is probably reasonable and sufficient to suppose that in a small volume of the workspace and for a given robot configuration (*i.e.* a small volume of joint space) the end effector orientation error is nearly constant. To correct for this, we need to compute a corrective transform at the end effector, which we expect to adjust as the robot traverses its workspace. If we can obtain a convenient procedure for dynamically estimating $\mathcal{T}_{CR}$ of our robot-mounted camera, then our estimate will incorporate whatever constant end-effector correction is appropriate to that region of the workspace. This is appealing since the global positioning accuracy for typical industrial manipulators can be quite poor, while small relative motions can often be made with high precision.

CHAPTER 3

Theory

We begin our development of estimators for the range camera hand-eye transform with the case in which a single point is detectable as an invariant (*i.e.* stationary) feature in the scene. Such a feature might be a corner of an object, the center of a solid object with known symmetries such as a sphere, ellipsoid or cube, or any other unambiguously identifiable point in the scene. We then proceed to examine the case where a single planar surface is used as the invariant. The plane is assumed infinite in the sense that we make no attempt to determine its boundary, as is appropriate in the case of a floor or tabletop forming the background of a scene. We find the development for plane features somewhat more troublesome than that for points, but assert that the effort is justified given the practical appeal of this method.

1. Point Features

Suppose that we can identify in the environment a stationary point $\mathbf{p}_W$, described with respect to the world frame by its Cartesian co-ordinates $[\ x\ y\ z\ ]$. Since $\mathbf{p}_W$ is unknown, we will develop an estimate for it which we call $\overline{\mathbf{p}_W}$. For several robot positions $\mathcal{T}_{RWi}$, we acquire a measurement $\mathbf{p}_{Ci}$ expressed in the local camera frame. Intuitively we expect that for a sufficiently large number of different robot/camera positions, and supposing that the measured data is perfect, there should exist a unique rigid transformation $\mathcal{T}_{CR}$ that is consistent with all of the measurements and with the hypothesis that $\mathbf{p}_W$ is indeed stationary. Our task is to determine the conditions under which $\mathcal{T}_{CR}$ is uniquely determined, and to recover the estimate $\overline{\mathcal{T}_{CR}}$ that is most consistent with a given noisy set of measurements.

In order to solve the optimization problem (2) we require:

- a representation of coordinate transformations to project point features from camera frame to world frame
- a metric with which to evaluate the quality of this projection for a given datum and parameter vector in the domain of the optimization

1.1. Representing Transformations. A useful representation of coordinate transformations is the homogeneous transformation matrix [22]. Homogeneous transformation matrices may also encode scaling, mirroring and perspective projection operations, but our interest is limited to those representing rigid motions. We therefore consider a subset of homogeneous transform matrices consisting of a 3×3 orthonormal rotation matrix R and a 3×1 translation vector $\mathbf{r}_t$, as,

$$T = \begin{bmatrix} & R & & \mathbf{r}_t \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

A point $\mathbf{p}$ is described by the augmented vector $\mathbf{p} = [\ x\ y\ z\ 1\ ]'$, so that $\mathbf{p}$ is rotated and translated by a single matrix multiplication, and transforms are compounded as,

$$\mathbf{p}_C = T_{BC} T_{AB} \mathbf{p}_A$$

Here we have adopted the same subscripting convention as for generic coordinate transformations.

The homogeneous transform matrix above, when considered component-wise, represents an over-parameterization of the space of rigid motions, as 12 numbers are required to describe 6 degrees of spatial freedom. While this causes some difficulty in the formulation of our minimization, it is attractive as it gives rise to both a useful linear approximation and a computationally convenient iterative path to the optimal solution.

1.2. Error metric. We project the camera frame measurement of each feature point $\mathbf{p}_{Ci}$ into the world frame as $\mathbf{p}_{Wi}$, and obtain the projection error as its distance to the estimated world frame feature $\mathbf{p}_W$. Since both $\mathbf{p}_{Wi}$ and $\mathbf{p}_W$ are points in Cartesian space, a natural distance metric δ_i is the 2-norm of the difference vector

$$\begin{aligned} \mathbf{e}_i &= \mathbf{p}_{Wi} - \mathbf{p}_W \\ \delta_i^2 &= \|\mathbf{e}_i\|_2^2 = \mathbf{e}_i' \mathbf{e}_i \end{aligned} \tag{4}$$

Following the kinematic loop of Figure 2.4, we can write the projected feature point using homogeneous coordinate transformations as

$$\mathbf{p}_{W\,i} = T_{RW\,i}\, T_{CR}\, \mathbf{p}_{C\,i} \tag{5}$$

We now write the Cartesian error vector defined in (4) as,

$$\mathbf{e}_i = T_{RW\,i}\, T_{CR}\, \mathbf{p}_{C\,i} - \mathbf{p}_W \tag{6}$$

Clearly, this expression for $\mathbf{d}_i$ is linear in the unknown parameters, which are the 12 components of the homogeneous transformation matrix T_{CR} and the 3 components of the feature point position $\mathbf{p}_W$. We introduce a shorthand notation for the terms of (6), as

$$T_{RW\,i} = \begin{bmatrix} W & \mathbf{w}_t \\ (\mathbf{0}^3)' & 1 \end{bmatrix} \qquad R_{CR} = \begin{bmatrix} \mathbf{r}_1 & \mathbf{r}_2 & \mathbf{r}_3 \end{bmatrix} \qquad T_{CR} = \begin{bmatrix} R_{CR} & \mathbf{r}_t \\ (\mathbf{0}^3)' & 1 \end{bmatrix}$$
$$\mathbf{p} = \begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix} \qquad \mathbf{p}_{C\,i} = \begin{bmatrix} \mathbf{p} \\ 1 \end{bmatrix} \qquad \mathbf{p}_W = \begin{bmatrix} \mathbf{p}_w \\ 1 \end{bmatrix} \tag{7}$$

where W is the 3×3 rotation sub-matrix of $T_{RW\,i}$, $\mathbf{r}_j$ are the column 3-vectors of T_{CR}, and $\mathbf{p}_w$ is also a 3-vector. Now we can rewrite (6) in a familiar form by defining the 15×1 vector of unknown parameters $\mathbf{x}$ as

$$\mathbf{x} = \begin{bmatrix} \mathbf{r}_1' & \mathbf{r}_2' & \mathbf{r}_3' & \mathbf{r}_t' & \mathbf{p}_w' \end{bmatrix}' \tag{8}$$

and write (6) in the standard linear form

$$\mathbf{e}_i = A_i\, \mathbf{x} + \mathbf{b}_i \tag{9}$$

where the 3×15 matrix A_i and 3×1 vector $\mathbf{b}_i$ are defined as

$$A_i = \begin{bmatrix} W p_x & W p_y & W p_z & W & -I^3 \end{bmatrix} \qquad \mathbf{b}_i = \mathbf{w}_t \tag{10}$$

Here, as throughout, I^3 is the 3×3 identity matrix. The minimum mean-squared error estimate of the camera transform and of the unknown feature point location is then the vector $\overline{\mathbf{x}}$ that minimizes the objective function

$$\Delta = \sum_i (A_i\, \mathbf{x} + \mathbf{b}_i)'(A_i\, \mathbf{x} + \mathbf{b}_i) \tag{11}$$

over all views i, subject to admissibility constraints on $\mathbf{x}$. We define a symmetric matrix H, vector $\mathbf{c}$ and scalar k, and expand the above to obtain,

$$\begin{aligned} H &= \sum_i A_i' A_i \\ \mathbf{c} &= \sum_i A_i' \mathbf{b}_i \\ k &= \sum_i \mathbf{b}_i' \mathbf{b}_i \\ \Delta &= \mathbf{x}' H \mathbf{x} + 2\mathbf{x}'\mathbf{c} + k \end{aligned} \tag{12}$$

1.3. Approximate Linear Solution. The objective function defined by (12) is a linear quadratic function to which the minimizing solution is well known from least squares estimation. Our development parallels the standard derivation of the normal equations of the linear least-squares problem [23], for which we can directly write the solution

$$\overline{\mathbf{x}}_l = -H^{-1}\mathbf{c} \tag{13}$$

The solution is uniquely determined if H is positive definite; since H is symmetric, this is equivalent to H having full rank. Since there are 15 unknowns and each view provides 3 independent constraint equations, no less than 5 viewpoints will be required to constrain this solution. The matter of selecting exactly 5 viewpoints so as to ensure that 15 linearly independent constraints are generated is difficult, and remains for further study. However, we can certainly detect degenerate viewpoint combinations by monitoring the condition of H. Furthermore, our experimental work has given rise to several heuristics that are useful in guiding viewpoint selection, which we discuss in Chapter 4.

It is well known from numerical analysis that inversion of H is not the most numerically stable means of solution to the linear least squares problem [9, 23]. This is largely because the condition number of H is proportional to the square of the condition number of the so-called "design matrix" $A = [\, A_0' \ldots A_n' \,]'$. The condition number of a matrix is the ratio of the largest and smallest singular values, which are the gains of the matrix along various prescribed basis directions. Since numerical accuracy of the solution is typically proportional to the condition number, a more stable computation can be performed by operating directly on A. QR factorization or singular value decomposition (SVD) of A, for example, both offer numerically robust solutions to the linear least squares problem [9]. However, the cost of this approach is evident in the fact that while H is finite-dimensional (15×15 in our example), A grows as the number of available views increases. Thus, if the number of views is small, we are advised to solve this system by one of the methods operating directly on A. Since in our case we desire a solution that can be implemented in finite memory for an infinite number of views, we prefer to form H and exercise caution when computing its inverse.

Equation (13) can be conveniently solved by any number of methods, notably Cholesky factorization. In our experimental work, we prefer the more costly singular value decomposition (SVD) for the diagnostic insight it offers into the condition of H. We may of course also weight each viewpoint dataset while accumulating H, $\mathbf{c}$ and k, in order to bias the estimate according to confidence in the data [23], although to do so in a meaningful way we need to model kinematic errors in the manipulator, which is beyond our present scope.

While the above solution generates values for $\overline{T_{CR}}$ and $\overline{\mathbf{p}_W}$ which result in the global minimum mean-squared error Δ for a given data set, we are not at all certain that $\overline{T_{CR}}$ corresponds to a real rigid motion. We must enforce on $\overline{\mathbf{x}}$ the constraint that it give rise to a real rotation in $\overline{T_{CR}}$, to which we now turn our attention.

1.4. Enforcing Orthonormality Constraints. A 3×3 matrix R is orthonormal if and only if its columns (and, equivalently, its rows) form a mutually orthogonal set of unit vectors. This requirement is compactly expressed by the condition $R'R = I$. Column-wise expansion of the rotation sub-matrix R of T_{CR} yields a set of 6 independent equations that enforce orthogonality and unit length, as

$$\begin{aligned}
\mathbf{r}_1'\mathbf{r}_1 &= 1 \\
\mathbf{r}_2'\mathbf{r}_2 &= 1 \\
\mathbf{r}_3'\mathbf{r}_3 &= 1 \\
\mathbf{r}_1'\mathbf{r}_2 &= 0 \\
\mathbf{r}_1'\mathbf{r}_3 &= 0 \\
\mathbf{r}_2'\mathbf{r}_3 &= 0
\end{aligned} \tag{14}$$

In addition to satisfying equations (14), the columns of a matrix representing rigid rotation also satisfy $\mathbf{r}_1 \times \mathbf{r}_2 = \mathbf{r}_3$, or $\det(R) = +1$, in order to disallow transformations representing reflection. In our case we find it unnecessary to enforce this last constraint, as reflective transformations are likely to produce a good fit only in the case of extremely poor data. We write the orthogonality constraint equations in the form of a vector equation,

$$\mathbf{G}(\mathbf{x}) = \mathbf{0}^6 \tag{15}$$

where $\mathbf{x}$ is as defined in (8), and the components of $\mathbf{G}(\mathbf{x})$ are understood to be ordered as in equations (14).

Returning to the original optimization of equation (2), we now have a linear quadratic objective function to minimize subject to a set of quadratic constraints on the optimization parameters. Formally, we solve

$$\min_{\mathbf{x}} (\mathbf{x}'H\mathbf{x} + 2\mathbf{x}'\mathbf{c} + k),\ \mathbf{G}(\mathbf{x}) = \mathbf{0} \tag{16}$$

This is easily solved by the method of Lagrange multipliers [18]. We form the Lagrangian

$$l(\mathbf{x}, \lambda) = \mathbf{x}'H\mathbf{x} + 2\mathbf{x}'\mathbf{c} + k + \lambda'\mathbf{G}(\mathbf{x}) \tag{17}$$

where λ is the 6×1 vector of Lagrange multipliers corresponding to each of the (ordered) equations (14). Then the solution of the constrained optimization (16) also minimizes the Lagrangian $l(\mathbf{x}, \lambda)$. The first order sufficiency conditions for a minimum of $l(\mathbf{x}, \lambda)$ are

$$\begin{aligned} \frac{\partial l(\mathbf{x}, \lambda)}{\partial \mathbf{x}} &= \mathbf{0} \\ \frac{\partial l(\mathbf{x}, \lambda)}{\partial \lambda} = \mathbf{G}(\mathbf{x}) &= \mathbf{0} \end{aligned} \tag{18}$$

which are precisely the conditions for a solution of (16).

Since $l(\mathbf{x}, \lambda)$ contains terms of the form $\lambda_i x_j x_k$ arising from the constraint equations, the partial derivatives appearing in (18) are *not* linear in the unknowns and an analytic solution for $(\overline{\mathbf{x}}, \overline{\lambda})$ is impossible. Instead, we apply Newton's method to derive an appropriate iteration. Expanding the above partial derivatives in a power series about $(\mathbf{x}_0, \lambda_0)$, discarding second and higher order terms, and combining with (17), we obtain

$$\begin{aligned}
\frac{\partial l(\mathbf{x},\lambda)}{\partial \mathbf{x}} &\approx \frac{\partial l(\mathbf{x}_0,\lambda_0)}{\partial \mathbf{x}} + (\mathbf{x}-\mathbf{x}_0)'\frac{\partial^2 l(\mathbf{x}_0,\lambda_0)}{\partial \mathbf{x}^2} + (\lambda-\lambda_0)'\frac{\partial^2 l(\mathbf{x}_0,\lambda_0)}{\partial \lambda \partial \mathbf{x}} \\
&= 2\mathbf{x}_0' H + 2\mathbf{c}' + \lambda_0' \frac{\partial \mathbf{G}(\mathbf{x}_0)}{\partial \mathbf{x}} + (\mathbf{x}-\mathbf{x}_0)'(2H + \frac{\partial^2(\lambda_0'\mathbf{G}(\mathbf{x}_0))}{\partial \mathbf{x}^2}) \\
&\quad +(\lambda-\lambda_0)'\frac{\partial \mathbf{G}(\mathbf{x}_0)}{\partial \mathbf{x}} \qquad (19) \\
\frac{\partial l(\mathbf{x},\lambda)}{\partial \lambda} &\approx \frac{\partial l(\mathbf{x}_0,\lambda_0)}{\partial \lambda} + (\mathbf{x}-\mathbf{x}_0)\frac{\partial^2 l(\mathbf{x}_0,\lambda_0)}{\partial \lambda \partial \mathbf{x}} \\
&= \mathbf{G}(\mathbf{x}_0) + (\mathbf{x}-\mathbf{x}_0)'\frac{\partial \mathbf{G}(\mathbf{x}_0)}{\partial \mathbf{x}} \qquad (20)
\end{aligned}$$

Equating the right hand sides above to zero and writing $\mathbf{dx} = (\mathbf{x}-\mathbf{x}_0)$, $\mathbf{d}\lambda = (\lambda-\lambda_0)$, we find the next iteration step $[\,\mathbf{dx}'\ \mathbf{d}\lambda'\,]'$ by solving the linear system

$$\begin{bmatrix} 2H + \frac{\partial^2(\lambda_0'\mathbf{G}(\mathbf{x}_0))}{\partial \mathbf{x}^2} & \frac{\partial \mathbf{G}(\mathbf{x}_0)}{\partial \mathbf{x}}' \\ \frac{\partial \mathbf{G}(\mathbf{x}_0)}{\partial \mathbf{x}} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{dx} \\ \mathbf{d}\lambda \end{bmatrix} + \begin{bmatrix} 2H\mathbf{x}_0 + 2\mathbf{c} + \frac{\partial \mathbf{G}(\mathbf{x}_0)}{\partial \mathbf{x}}'\lambda_0 \\ \mathbf{G}(\mathbf{x}_0) \end{bmatrix} = \mathbf{0} \qquad (21)$$

Note that vector equation (21) is composed of the data accumulators H and $\mathbf{c}$ computed in the linear approximative solution, and derivatives of the constraint equations $\mathbf{G}(\mathbf{x})$ which are independent of the data. This is highly convenient from a computational viewpoint, since at each iteration we need only evaluate several simple functions of the constraints at the current solution $[\,\mathbf{x}_0'\ \lambda_0'\,]'$. In addition to $\mathbf{G}(\mathbf{x})$, we also need its gradient and the Hessian of $\lambda_0'\mathbf{G}(\mathbf{x})$ with respect to the optimization parameters $\mathbf{x}$,

$$\frac{\partial \mathbf{G}(\mathbf{x})}{\partial \mathbf{x}} = \left[\begin{array}{ccc|c} 2\mathbf{r}_1' & \mathbf{0} & \mathbf{0} & \\ \mathbf{0} & 2\mathbf{r}_2' & \mathbf{0} & \\ \mathbf{0} & \mathbf{0} & 2\mathbf{r}_3' & \mathbf{0}^{6\times 6} \\ \mathbf{r}_2' & \mathbf{r}_1' & \mathbf{0} & \\ \mathbf{r}_3' & \mathbf{0} & \mathbf{r}_1' & \\ \mathbf{0} & \mathbf{r}_3' & \mathbf{r}_2' & \end{array}\right]$$

$$\frac{\partial^2(\lambda'\mathbf{G}(\mathbf{x}))}{\partial \mathbf{x}^2} = \left[\begin{array}{ccc|c} 2\lambda_1 I^3 & \lambda_4 I^3 & \lambda_5 I^3 & \\ \lambda_4 I^3 & 2\lambda_2 I^3 & \lambda_6 I^3 & \mathbf{0}^{9\times 6} \\ \lambda_5 I^3 & \lambda_6 I^3 & 2\lambda_3 I^3 & \\ \hline & \mathbf{0}^{6\times 9} & & \mathbf{0}^{6\times 6} \end{array}\right] \qquad (22)$$

which are 6×15 and 15×15 respectively. Note that $\mathbf{r}_j$ in the above refer to column vectors of the unknown transformation T_{CR}, which themselves are elements of the present value of the solution $\mathbf{x}_0$. The symmetric system (21) can now be solved by standard techniques for the next iteration step. We thus obtain an iterative solution for $\overline{T_{CR}}$ and $\overline{\mathbf{p}_W}$ that locally minimizes the objective and satisfies the stated orthogonality constraints.

The successful solution of general nonlinear optimization depends on the availability of a good initial estimate for the optimal parameters, both to minimize the number of iterations required for convergence and to avoid falling into local minima. We have just such an estimate available in the form of the linear solution to the unconstrained problem. The practical estimation procedure can therefore be accomplished by the following prescription. For each view i of the feature point,

(i) Form the matrix A_i and vector $\mathbf{b}_i$, and accumulate the partial sums $H = \sum_i w_i A_i' A_i$, $\mathbf{c} = \sum_i w_i A_i' \mathbf{b}_i$, and $k = \sum_i w_i \mathbf{b}_i' \mathbf{b}_i$. w_i is an optional weighting term which can be used to bias the solution in favour of more certain data, and is inversely proportional to the expected value of the mean squared projection error δ_i. We often assume all data to be equally reliable, and set $w_i = 1$ for all i.

(ii) If $i >= 5$, check the numerical condition of H. If H is near singular, then insufficient independent constraints have been generated by the available viewpoints, and more data is required before proceeding.

(iii) Obtain an initial estimate for the unknown parameters $\overline{\mathbf{x}}$. This may be known *a-priori*, it may be the solution obtained from previous views, or it may be the unconstrained linear approximation to the optimal parameters. In the latter case, compute the approximation $\overline{\mathbf{x}}_l = -H^{-1}\mathbf{c}$.

(iv) Using the initial estimate obtained above, iterate according to equation (21) until convergence.

At convergence, we expect that the orthonormality constraints on $\overline{T_{CR}}$ are satisfied, and that gradient of the objective function is zero. Numerically, of course, we cannot expect exact convergence, and must instead set thresholds on some suitable convergence measures. We use the magnitudes of the constraint error $\|\mathbf{G}(\mathbf{x})\|$ and of the step size $\|\mathbf{dx}\|$. When both of these measures fall below limits based on desired accuracy and machine precision, convergence is inferred. Under this scheme, each time we acquire new data we immediately generate an update estimate that utilizes all information available thus far.

1.5. Observations on the Point Feature Formulation. Suppose that we have at least five views of the feature point which give rise to 15 independent constraints on the solution vector. In the ideal case of noiseless data, all camera measurements will be exactly consistent and the total projection error $\Delta_{\min}$ will be zero. Since the minimum of a linear optimization with positive definite H is unique, the minimizing parameters will be the true values. The true value T_{CR} clearly has an orthonormal rotation component, and therefore the minimizing parameters $\overline{\mathbf{x}}_l$ will satisfy $\mathbf{G}(\overline{\mathbf{x}}_l) = \mathbf{0}$.

As we increase measurement error in the data, we expect the world space projections of feature points to be become inconsistent (*i.e.* $\Delta_{\min} > 0$), and $\overline{\mathbf{x}}_l$ to move away from the true solution. $\overline{\mathbf{x}}_l$ is itself a random quantity which can be characterized by statistical measures. For a given data set, $\overline{\mathbf{x}}_l$ will likely generate a non-orthonormal rotation sub-matrix, and we will iterate to find the orthonormal solution $\overline{\mathbf{x}}$ near to $\overline{\mathbf{x}}_l$ which minimizes the projection error. In principle, the constrained minimization may have multiple solutions due to local minima. Furthermore, if the initial value used in the iteration is far from the true solution, the iteration may not converge. We therefore require that the deviation of $\overline{\mathbf{x}}_l$ from the minimum error constrained estimate $\overline{\mathbf{x}}$ be small. In practice this means that extremely noisy data may yield a linear estimate that cannot adequately seed the iteration, and result in either a spurious solution or a divergent iteration.

If a good prior estimate of the unknown parameters is available, it may be used in place of the linear solution to seed the constrained minimization. In cases of extremely noisy data, we expect that the linear solution may behave poorly and an independent estimate may be required. Since the orthonormality equations provide constraints over 6 of the 15 degrees of freedom in the solution vector, and since each view of the feature point generates three equations, we infer that the constrained optimization is solvable given at least 3 viewpoints. This number reflects the fundamental geometry of the problem, in contrast to the 5 views required for the linear solution as an artifact of the over-parameterization of rotations.

While the constrained estimate provides the best fit to the data based on a rigid body model of the camera/robot system, we note that in general this estimate results in a greater mean prediction error $\sqrt{\Delta/n}$ than the linear estimate. This is hardly surprising since the linear estimate is based on a model that offers 6 more degrees of freedom which are adjusted to obtain a better fit to the data. However, following the above discussion, we expect that the mean prediction errors associated with the two models should not differ drastically, provided that the rigid transformation model is appropriate to the system. In particular, if the camera intrinsic calibration or the robot kinematic model are very poor we expect the constrained estimate to perform significantly worse than the unconstrained one, in terms of mean prediction errors. If we observe such a situation, we can infer that the kinematic loop comprising camera, robot and feature point is not well modeled by the rigid transformation model described in Figure 2.4.

In this case, we may wish to look more closely at the unconstrained estimator. This solution incorporates the degrees of freedom required to model and compensate for several linear distortion modes of the camera, such as anisotropic co-ordinate scaling. If the unconstrained result is stable over several datasets and is significantly better than the constrained result in terms of projection error, we should consider discarding some of the orthonormality constraint equations or using the linear estimate directly.

Finally, we emphasize that our estimator makes use of all of the available data in a symmetric fashion. Furthermore, it does so in finite memory, so that it may be applied as a filter operating on an infinite input data stream, and continuously delivering updated best-fit transformation parameters. As such, it provides a means of dynamically computing a first order correction to the robot kinematic model, along the lines suggested by Foulloy and Kelly [7] and discussed in Section 4.6 of Chapter 2. We would like the estimates to adapt quickly as the manipulator moves between regions of the workspace, and this can be achieved by varying the weight assigned to each datum according to some criterion. The design of our estimator admits several such schemes at very low computational cost. We now briefly explore two adaptations of our basic approach to illustrate this idea.

1.6. Viewpoint weighting schemes. The estimator presented above makes no assumption as to spatial or temporal ordering of the available viewpoints. While this is appropriate for the case of a static calibration where the goal is to minimize projection errors over the entire workspace, it is not always a desirable behavior. Consider for instance the case where our prime interest is in assuring the best possible mapping between images captured from two viewpoints adjacent in time. Here we intend that adjacency in time implies adjacency in space, as in the case of a manipulator following a simple trajectory and stopping periodically to capture the scene. In such a case we are not overly concerned with the quality of mapping over the entire robot workspace, but in recovering a relative motion with the highest possible precision.

For this purpose we modify the error metric (4) so as to reflect only the error incurred in mapping between adjacent frames, as

$$\mathbf{e}_i = \mathbf{p}_{W\,i} - \mathbf{p}_{W\,i-1} \tag{23}$$

We call this the differential form of the point-based estimator. Notice that in this form we impose no penalty on a solution which maps poorly between viewpoints more than one frame apart. We also completely eliminate from the unknowns the components of $\overline{\mathbf{p}_W}$, thus reducing the number of variables for which we need to solve. Now the vector of unknown parameters $\mathbf{x}$ contains only the 12 components of T_{CR}, and we accumulate at each viewpoint the components

$$\begin{aligned} H &= \sum_i (A_i - A_{i-1})'(A_i - A_{i-1}) \\ \mathbf{c} &= \sum_i (A_i - A_{i-1})'(\mathbf{b}_i - \mathbf{b}_{i-1}) \\ k &= \sum_i (\mathbf{b}_i - \mathbf{b}_{i-1})'(\mathbf{b}_i - \mathbf{b}_{i-1}) \end{aligned} \tag{24}$$

where the last three columns of A_i have been truncated with respect to equation (10). The remainder of the solution now proceeds as before. If the manipulator trajectory and image capture schedule are such that adjacency in time implies closeness in both space and manipulator configuration, then this formulation will reject large-scale positioning errors of the manipulator and provide an estimate of T_{CR} that is optimized for merging nearby views by relative motions.

A similar behavior can be obtained by embedding view-aging in the estimator, based on the idea that what happened long ago is less important than what is happening now. Returning to the original formulation, if we accumulate data matrices according to the rule

$$H_i = aH_{i-1} + A_i'A_i \tag{25}$$

and similarly for $\mathbf{c}$ and k, with $0 < a < 1$, then the weight of data acquired at a given viewpoint will fall off according to a first order decay controlled by the time constant a. Our rough notion of time in this example is defined in terms of the availability of new viewpoint data on an ongoing basis, though in principle the decay of a data set could equally well be referred to clock time, or indeed to any other monotonically increasing quantity.

2. Plane Features

Our development of an estimator for the camera mounting transform using single points is motivated by our aim to minimize complexity of the image analysis on which feature extraction depends, and to correspondingly broaden the class of scenes which can provide the necessary constraints to our estimation process. Often we can extract the location of a point feature with good reliability and relatively little computational burden, where determining the full 6-DOF position and pose of a feature might be difficult - or impossible, if the feature object possesses inherent symmetry.

Following this argument, we observe that a robot vision workcell is frequently composed of a collection of objects placed on a planar surface, such as a floor or table-top. This background surface is often considered uninteresting and ignored, except to the degree required for placement of objects *on* the surface and for collision avoidance. In cases where a background plane is known to exist and to be immobile with respect to the manipulator base frame, we are tempted ask whether it can be used as the invariant feature in our dynamic calibration process.

It is not immediately obvious that measurements of arbitrary points in an infinite plane can provide sufficient information to constrain an estimate of the hand-eye transform. Note that an infinite plane in R^3 is characterized by three symmetries: two directions of translation in the plane, and one of rotation about the normal. Equivalently, we think of it as providing three constraints over the 6-DOF space of rigid motions. Dimensionally speaking, this is no worse than the single point feature which offers three translational constraints and three rotational symmetries, so we are encouraged to proceed by analogy to the point feature development.

2.1. Plane Representations. A plane $\mathcal{Q}$ embedded in R^3 is defined as a set of points $\mathbf{v} = [\, x\ y\ z \,]'$ satisfying the linear relation $f(\mathbf{v}) = ax + by + cz + d = 0$. We speak of a planar surface as being parameterized by the vector of coefficients of this expression $\mathbf{q} = [\, a\ b\ c\ d \,]'$. Recall that since $\nabla f(\mathbf{v}) = [\, a\ b\ c \,]$, the first three parameters are the components of a vector normal to the plane, and we will often write $\mathbf{q} = [\, \mathbf{u}'\ d \,]'$. The parameterization of the plane by $\mathbf{q}$ is not unique, since $\mathbf{q}$ and $k\mathbf{q}$ describe the same plane. However, if we choose k such that $k^2(a^2 + b^2 + c^2) = 1$, then the representation is unique up to the sign of the normal. This last ambiguity is resolved for planes which do not pass through the origin if we also, arbitrarily, insist that $kd < 0$. We therefore define the canonical representation of a plane $\mathcal{Q}$ as $\mathbf{q} = [\, a\ b\ c\ d \,]'$, such that

$$\begin{aligned} ax + by + cz + d &= 0 \quad \forall [\, x\ y\ z \,] \in \mathcal{Q} \\ a^2 + b^2 + c^2 &= 1 \\ d &\leq 0 \end{aligned} \tag{26}$$

with the understanding that when $d = 0$ the representation is singular. This implies that for planes containing the origin, sign ambiguity of the normal remains.

With this parameterization, the perpendicular distance η of a point $\mathbf{w}$ to the plane is simply $|f(\mathbf{w})|$. Referring to Figure 3.6, we see that for any point $\mathbf{v}$ in the plane, and with unit normal to the plane $\mathbf{u} = [\, a\ b\ c \,]'$, we have

$$\eta = |(\mathbf{w} - \mathbf{v}) \cdot \mathbf{u}| = |\mathbf{w} \cdot \mathbf{u} + d| = |f(\mathbf{w})|$$

We immediately also see that the parameter d is the (negative) perpendicular distance from the plane to the origin.

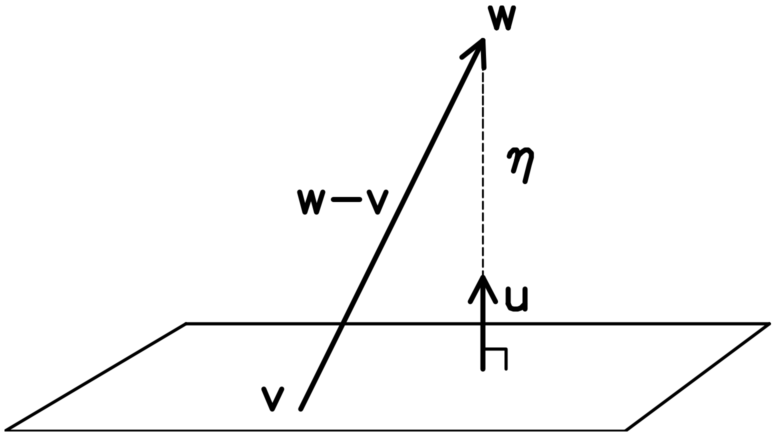

FIGURE 3.6. Perpendicular distance from a point to a plane

A mechanism for transforming plane representations between coordinate frames is also required. Using the homogeneous coordinate representation for a point $\mathbf{p} = [\mathbf{v}'1]'$ and our canonical representation of a plane $\mathbf{q}$, we can write the plane equation in vector form simply as $\mathbf{q}'\mathbf{p} = 0$. Considering two coordinate frames A and B related by the homogeneous transform matrix T_{AB}, we seek a transformation $\mathcal{T}_{AB}$ which maps the plane representation $\mathbf{q}_A$ into $\mathbf{q}_B$. Since we must have $\mathbf{q}'_A\mathbf{p}_A = 0$ and $\mathbf{q}'_B\mathbf{p}_B = 0$, we can write

$$\begin{aligned} \mathbf{q}'_B(T_{AB}\mathbf{p}_A) &= \mathbf{q}'_A\mathbf{p}_A = 0 \\ \mathbf{q}'_B T_{AB} &= \mathbf{q}'_A \\ \mathbf{q}_B &= {T'_{AB}}^{-1}\mathbf{q}_A \end{aligned} \tag{27}$$

We see that planes are transformed in a manner analogous to vectors by applying the inverse transpose of the homogeneous transform matrix, *i.e.* $\mathcal{T}_{AB} = (T'_{AB})^{-1} = T'_{BA}$. Since the rotation part R_{AB} of T_{AB} is orthonormal, we have ${R_{AB}}^{-1} = R'_{AB}$. A unit normal vector transformed by $(R'_{AB})^{-1} = R_{AB}$ will therefore also have unit length. We often write the plane transformation $\mathcal{T}_{AB}$ in terms of the components R and $\mathbf{r}_t$ of the natural (*i.e.* point-based) homogeneous transform, as

$$\mathcal{T}_{AB} = \begin{bmatrix} R_{AB} & \mathbf{0}^3 \\ -\mathbf{r}'_t R_{AB} & 1 \end{bmatrix} \tag{28}$$

The above discussion shows that the first two conditions of (26) are preserved by homogeneous transformations of plane representations. The last condition ($d \leq 0$) is not satisfied in general, but in our case we can avoid the problem by ensuring that all co-ordinate systems in use have their origins on one side of the plane. This ensures that the sign of d is transformed uniformly in all cases, and usually occurs implicitly when the plane of interest is the background. If a transformation displaces the local coordinate origin across the plane, we simply multiply $\mathbf{q}$ through by -1 to preserve the convention.

2.2. Error Metric. We need to define a metric describing the similarity between two plane representations in order to formulate an optimization of the form of equation (2). The canonical plane representation ensures uniqueness except in the case where the plane contains the origin, and this can always be avoided by application of an arbitrary constant transform. We therefore use the magnitude of the vector difference between plane representations, as we did for point features,

$$\begin{aligned} \mathbf{e}_i &= \mathbf{q}_{Wi} - \mathbf{q}_W \\ \delta_i^2 &= \|\mathbf{e}_i\|_2^2 = \mathbf{e}_i'\mathbf{e}_i \end{aligned} \tag{29}$$

As in (4), $\mathbf{q}_{Wi}$ and $\mathbf{q}_W$ are 4-vectors, although here they represent plane surfaces instead of points in space.

While this has some of the desired properties of a metric, there is a dimensional problem due to anisotropy of the plane representation. In taking the 2-norm, we add squared dimensionless quantities arising from the plane normal components, to d^2 which has dimensions of squared length. This is equivalent to the observation that two planes can differ in two distinct modes according to whether or not the planes intersect. In general, there is no fundamental way to decide how to weigh each of these modes in a single difference metric, and this manifests itself in the fact that behavior of the metric is *not* independent of our choice of inches or millimeters for the expression of d in the plane representation.

This is a serious problem for our metric, for which we will show a more general solution in Section 3. To continue in the theme of the point formulation, however, we introduce a scaling constant to balance the penalty applied for parallel planes which differ by a normal translation, against those which intersect but differ in terms of the normal orientation. We define the difference vector

$$\mathbf{e}_i = \begin{bmatrix} \mathbf{u}_{Wi} - \mathbf{u}_W \\ \kappa(d_{Wi} - d_W) \end{bmatrix} \tag{30}$$

where $\mathbf{u}$ represents the first three components of the canonical plane representation and d the last. The scale constant κ is chosen such that the two components of (30) are comparable in magnitude for planes that are similar in the region of the workspace over which we can acquire image data. We refer to the metric defined in this way as the direct plane formulation.

2.3. Approximate Linear Solution. We now have a problem in exactly the same format as the points formulation, and we can proceed to a solution by the same means. The projection of the camera frame plane to the world frame is

$$\mathbf{q}_{Wi} = (T_{RW}{}'_i)^{-1}(T_{CR}{}')^{-1}\mathbf{q}_{Ci} \tag{31}$$

which we combine with (30) to obtain the objective function. We write the components of $\mathbf{e}_i$ as

$$(T_{RW}{}'_i)^{-1} = \begin{bmatrix} W & \mathbf{0}^3 \\ -\mathbf{w}'_t W & 1 \end{bmatrix} \qquad R_{CR} = \begin{bmatrix} \mathbf{r}_1 & \mathbf{r}_2 & \mathbf{r}_3 \end{bmatrix} \qquad (T_{CR}{}')^{-1} = \begin{bmatrix} R_{CR} & \mathbf{0}^3 \\ \tilde{\mathbf{r}}_t{}' & 1 \end{bmatrix}$$
$$\mathbf{u} = \begin{bmatrix} u_x \\ u_y \\ u_z \end{bmatrix} \qquad \mathbf{q}_{Ci} = \begin{bmatrix} \mathbf{u} \\ d \end{bmatrix} \qquad \mathbf{q}_W = \begin{bmatrix} \mathbf{u}_w \\ d_w \end{bmatrix} \tag{32}$$

where the translation component $\tilde{\mathbf{r}}_t$ is defined by $\tilde{\mathbf{r}}_t = -R_{CR}{}'\mathbf{r}_t$ as in (28). The vector of unknown parameters $\mathbf{x}$ now has 16 components,

$$\mathbf{x} = \begin{bmatrix} \mathbf{r}'_1 & \mathbf{r}'_2 & \mathbf{r}'_3 & \tilde{\mathbf{r}}'_t & \mathbf{u}'_w & \kappa d_w \end{bmatrix}' \tag{33}$$

and the error vector can be written as $\mathbf{e}_i = A_i\mathbf{x} + \mathbf{b}_i$, with

$$A_i = \begin{bmatrix} Wu_x & Wu_y & Wu_z & \mathbf{0}^3 & -I^3 & 0 \\ -\mathbf{w}'_t W u_x & -\mathbf{w}'_t W u_y & -\mathbf{w}'_t W u_z & \mathbf{u}' & \mathbf{0}^3 & -1 \end{bmatrix}$$
$$b_i = \begin{bmatrix} \mathbf{0}^3 \\ \kappa d \end{bmatrix} \tag{34}$$

Note that we have one more equation than in the point formulation, so that A_i and $\mathbf{b}_i$ are 4×16 and 4×1 respectively. Once a solution vector $\overline{\mathbf{x}}$ is obtained, we find the natural translation component of T_{CR} as $\mathbf{r}_t = -R_{CR}\tilde{\mathbf{r}}_t$.

Existence of the linear approximative solution, as before, depends on the rank of $H = \sum A_i' A_i$. Since we have four independent equations for each viewpoint and 16 unknowns, we expect no less than 4 views to be required.

2.4. Constraints. Once we have an initial estimate (via the linear solution or by other means), we can proceed to enforce constraints. The iteration of equation (21) applies almost verbatim, except that the constraint vector $\mathbf{G}(\mathbf{x})$ is augmented by the requirement that the normal of the estimated feature plane have unit length, *i.e.* $\mathbf{u}_w' \mathbf{u}_w = 1$. This yields a total of 16 unknowns and 7 constraints.

If we already have available good estimates of the unknown parameters, these can be used in place of the linear solution vector. In this case, the components of $\mathbf{G}(\mathbf{x})$ constrain 7 of the 16 degrees of freedom, which implies that the system can be solved uniquely given at least 3 views.

Provided that the scale factor κ is chosen so that the metric reflects a reasonable balance between orientation and translation errors, and that viewpoints are chosen such that H is well conditioned, we obtain estimates of the unknown transformation $\overline{T_{CR}}$ and plane parameters $\overline{\mathbf{p}_W}$ by a short iteration. Although good results are possible with this technique (as we report in Chapter 4), the need for the scaling factor κ indicates a fundamental problem with our distance metric over the space of planes.

The issue of identifying metrics over the spaces of rigid rotations and translations has been treated in the literature of theoretical kinematics. The set of Euclidean rotations in 3-space has the properties of an algebraic group, and is commonly referred to as the special orthogonal group $SO(3)$ [2]. Similarly, the group of combined rotations and translations (*i.e.* rigid motions) is referred to as $SE(3)$. While it is possible to define a distance metric over $SO(3)$, it has been shown that it is difficult to define a useful metric over $SE(3)$ [20, 21]. In particular, there exists no metric on $SE(3)$ having the property of *bi-invariance*, meaning invariance with respect to displacements of both world (*i.e.* inertial) and local (*i.e.* moving) frames of reference. Furthermore, efforts to devise a metric invariant with respect to one frame or the other requires the introduction of an arbitrary scaling factor relating rotational distances to translational ones. Our problem with plane metrics is subsumed by that of metrics on $SE(3)$, as planes are parameterized in a space composed of two degrees of rotation and one of translation.

3. Partitioned Plane Formulation

The foregoing development of the direct estimator for plane features proceeds by close analogy to that for points, with little attention to the information content of a plane representation derived from range data. Closer examination of the transformation constraints provided by a detected plane allows us to develop an estimator that circumvents the metric scaling problem discussed above.

Examining the linear objective function expanded in equation (34), we observe that the first three components of $\mathbf{e}_i$ are independent of the translation part of T_{CR} and of the d element of the world plane vector $\mathbf{q}_W$, as well as of the translation parameters of the data T_{RWi} and $\mathbf{q}_{Ci}$. We expect this since the plane normal measured in camera frame is transformed to its representation in the world frame by a sequence of pure rotations. The translational components, however, depend strongly on the rotational ones.

This suggests a natural partition of the procedure for plane features whereby we first solve for the best camera rotation on the basis of consistent mapping of normals, and then fix the unknown length parameters on the basis of the optimal rotation. Although partitioning an optimization in this manner will not generally yield a globally optimal solution, there exists no definitive measure of optimality in the absence of a meaningful invariant metric over the entire space of plane representations. We therefore begin by solving for the camera hand-eye rotation which most consistently maps all observed unit normal vectors to a constant unknown normal in the world frame.

3.1. Solving the Optimal Rotation. We seek the rotation sub-matrix R of the camera hand-eye transformation T_{CR} which best maps each camera frame representation of the plane normal to a single world frame vector. A suitable metric for evaluating the distance between transformed unit normals is required.

The natural metric for unit vectors is geodesic distance on the unit sphere, which is simply the angle θ between two vectors. If instead we consider the straight-line Euclidean distance, or magnitude of the vector difference, we obtain a metric that varies as $\sqrt{1-\cos\theta}$ with respect to the geodesic metric. Since this relation is monotonic over the range 0 to π, we use the more convenient difference metric in our formulation.

As in previous developments we obtain a linear quadratic expression for the objective function for which we can directly compute the minimizing parameters, and then iterate to enforce the constraints. The components of the objective function are written in terms of the measured data and the unknown parameters, via the kinematic loop of Figure 2.4. For each view i we have, with definitions as in (32),

$$\begin{aligned}\mathbf{e}_i =& WR\mathbf{u} - \mathbf{u}_w \\ =& u_x W\mathbf{r}_1 + u_y W\mathbf{r}_2 + u_z W\mathbf{r}_3 - \mathbf{u}_w\end{aligned} \tag{35}$$

and wish to minimize

$$\min_{\substack{R\in\Omega \\ \mathbf{u}_w\in\Gamma}} \sum_{i=1}^{n} \mathbf{e}_i'\mathbf{e}_i \tag{36}$$

Here Ω is the set of Euclidean rotations, and Γ the set of unit vectors in R^3. Equation (35) is linear in the 12 components of the unknowns R and $\mathbf{u}_w$, so we cast this expression in the standard form by defining the vector of unknown parameters as

$$\mathbf{x}_\mathbf{r} = \begin{bmatrix}\mathbf{r}_1 \\ \mathbf{r}_2 \\ \mathbf{r}_3 \\ \mathbf{u}_w\end{bmatrix} \tag{37}$$

This results in a homogeneous linear formulation $\mathbf{e}_i = A_{\mathrm{r}i}\mathbf{x}_\mathbf{r}$ at each viewpoint, with

$$A_{\mathrm{r}i} = \begin{bmatrix} Wu_x & Wu_y & Wu_z & -I_3 \end{bmatrix} \tag{38}$$

Considering all available viewpoints we now minimize

$$\Delta = \sum_i \mathbf{e}_i'\mathbf{e}_i = \sum_i \mathbf{x}_\mathbf{r}' A_{\mathrm{r}i}' A_{\mathrm{r}i}\mathbf{x}_\mathbf{r} \tag{39}$$

This expression has a minimum for $\overline{\mathbf{x_r}}$ such that $(\sum_i A_{\mathrm{r}i}' A_{\mathrm{r}i})\overline{\mathbf{x_r}} = H_{\mathrm{r}}\overline{\mathbf{x_r}} = 0$. This implies that we seek a nonzero vector $\overline{\mathbf{x_r}}$ in the null space of H_{r}, or equivalently in the null space of the *design matrix* $A_{\mathrm{r}} = [A_{\mathrm{r}0}' \ldots A_{\mathrm{r}n}']'$.

This poses a more difficult problem than the non-homogeneous linear systems previously encountered. One difficulty is that solutions are not unique, except in the degenerate case where the system matrix has full rank and the solution is the zero vector. In general, we have a family of solutions of the form $\mathbf{x} = \sum c_k \boldsymbol{\psi}_k$, where the set of vectors $\boldsymbol{\psi}_k$ form a basis of the null space. If the system matrix has a nullity[1] greater than one, then construction of the best solution (in terms of the objective) that also satisfies some constraints may be difficult. If the system matrix has a nullity of one and we know that a solution exists, then the solution must be $c\boldsymbol{\psi}$, where $\boldsymbol{\psi}$ is the unique null direction of the system matrix. It then suffices to choose c so as to satisfy the constraints, which must be possible if a solution exists.

We first consider the case of perfectly consistent, noiseless data. It can be shown by construction that there exist sets of four or more viewpoints such that the nullity of H_{r} is one. We also know that a solution with R_{CR} orthogonal and $\mathbf{u}_w$ unit-length must exist, so H_{r} cannot have full rank for any collection of consistent views. We can therefore construct a solution $\overline{\mathbf{x_r}}_l = c\boldsymbol{\psi}$ by finding the null space basis vector $\boldsymbol{\psi}$ and choosing c so as to satisfy constraints on $\overline{\mathbf{x_r}}_l$.

If the data are noisy, then H_{r} will generally have full rank. However, the nonzero $\overline{\mathbf{x_r}}_l$ which minimizes $\|H_{\mathrm{r}}\overline{\mathbf{x_r}}\|_2/\|\overline{\mathbf{x_r}}\|_2$ lies in the direction $\boldsymbol{\psi}$ that is closest to the null space of H_{r}. Furthermore, we expect that unless the data are extremely noisy, we can choose c so that $\overline{\mathbf{x_r}}_l$ is close to the constraint surface, and use it as the initial estimate for iterative enforcement of the constraints.

The vector closest to the null space of H_{r} is obtained in a reliable way by the singular value decomposition [9, 23]. The SVD of H_{r} is the triple of matrices U, Λ, V, defined such that

$$U \Lambda V' = H_{\mathrm{r}} \tag{40}$$

where U and V are orthonormal, and Λ is diagonal. The elements of Λ are the singular values σ_j of H_{r}. A useful property of the SVD is that the columns of U

[1]The *nullity* of matrix A is defined as the dimension of the null space of A.

corresponding to nonzero singular values form an orthonormal basis for the range of H_r, and the columns of V corresponding to zero singular values form a basis for the null space. If H_r has full rank, then the column vector $\boldsymbol{\psi}$ of V corresponding to the smallest singular value is closest to the null space, in the sense that $\overline{\mathbf{x_r}} = \boldsymbol{\psi}$ minimizes $\|H_r\overline{\mathbf{x_r}}\|_2/\|\overline{\mathbf{x_r}}\|_2$.

We compute the SVD of H_r (or of A_r if high numerical precision is preferred over finite memory usage), and find the unit 12-vector $\boldsymbol{\psi}$ corresponding to the smallest singular value. Since by convention singular values are ordered in descending magnitude, the condition number of H_r is σ_{12}/σ_1, and indicates how close the matrix is to rank-deficiency. This is an indicator of the degree to which the data is consistent with the projection model (recall that, in this case, we *desire* a singular matrix). Another useful indicator is the ratio σ_{12}/σ_{11}, which measures how clearly we can discriminate the null direction from other directions in parameter space. We call this indicator the *detectability* of the unique null direction.

Having determined $\boldsymbol{\psi}$, we need to choose k such that with $\overline{\mathbf{x_r}}_l = k\boldsymbol{\psi}$ the constraints are most nearly satisfied. Evidently we have no control over the mutual orthogonality of the columns of R_{CR} which comprise $\overline{\mathbf{x_r}}_l$, nor indeed over their relative magnitudes. The best we can do is to note that if $\mathbf{x_r}$ satisfies the constraints, then $\|\mathbf{x_r}\|_2 = 4$. Since $\|\boldsymbol{\psi}\|_2 = 1$, we set $k = \pm 2$, choosing the sign such that $\det(R_{CR}) > 0$.

3.2. Enforcing Constraints. The initial estimate obtained above may be taken as the final solution, following the discussion of Section 1.5. More often we will use it to seed an iterative procedure to enforce the parameter space constraints of our projection model. As in the first formulation for plane features, we have seven constraint equations forming the vector equation $\mathbf{G}(\mathbf{x}) = \mathbf{0}^7$. Six of these enforce orthonormality on R_{CR}, and one assures unit length of the world-frame normal $\mathbf{u}_w$. We use the iteration of equation (21), with $\mathbf{b} = 0$ and the gradient and Hessian of the constraints constructed according to the definition of the vector of unknowns $\mathbf{x}$.

Provided that the initial estimate is sufficiently close to the minimum of the constrained objective function, the optimal estimate of the rotation part R_{CR} and the world plane normal vector $\mathbf{u}_w$ is obtained at convergence. Recall that these estimates are obtained without use of any data having units of length, and that we have yet to fix estimates for the unknown translational parameters.

3.3. Solving the Optimal Translation. Here we present two methods for estimating the translational components of the unknown parameters. Method 1 follows directly from the direct estimator for plane features. Method 2 addresses some of the inherent weaknesses of Method 1, but to our great surprise often generates estimates that are inferior to those obtained by Method 1 in simulation. We do not as yet fully understand why this occurs, and leave this issue for future investigation. For completeness we report both methods, and comment further on the failure of Method 2 in Chapter 4.

3.3.1. *Plane translations - Method 1.* Examining equations (34) we see that the equation corresponding to the fourth element of $\mathbf{e}_i$ can be used to constrain the unknown translational parameters. Once the rotational parameters R_{CR} and $\mathbf{u}_W$ are fixed, we are left with an expression for the translational error that is linear in the unknowns $\tilde{\mathbf{r}}_t$ and d_w. Defining the vector of unknowns $\mathbf{x}$ as

$$\mathbf{x} = \begin{bmatrix} \tilde{\mathbf{r}}_t \\ d_w \end{bmatrix} \tag{41}$$

we rewrite the fourth row of equation (34) as $e_i = A_i\mathbf{x} + b_i$, with

$$A_i = \begin{bmatrix} \mathbf{u}' & -1 \end{bmatrix} \qquad b_i = -\,\mathbf{w}_t W R\mathbf{u} + d \tag{42}$$

with $\mathbf{u}$, $\mathbf{w}_t$, W and R defined as in (32). The minimizing parameters are then found as usual by computing $\overline{\mathbf{x}} = \left(\sum_i A_i' A_i\right)^{-1} \sum_i A_i' b_i$.

This computation can be performed in finite memory, but the fact that the terms of $\sum_i A_i' A_i$ and $\sum_i A_i' b_i$ depend on the rotational estimates (which change as more views are acquired) makes this difficult. It is possible to factor the rotational estimates out of the above expressions so that summation is performed exclusively over measurement data. This comes at the cost of storing a larger number of independent scalar quantities than would be required for the direct evaluation of $\overline{\mathbf{x}}$, but allows us to achieve a finite memory implementation of the estimator. We discuss this factorization further in connection with the finite memory implementation of Method 2 in Section 3.4, where the problem is much more difficult.

3.3.2. *Plane translations - Method 2.* In the direct formulation for plane features we used the homogeneous 4-vector derived from the plane equation to obtain a simple algebraic metric for comparing planes. This metric is a weighted sum of two components: the magnitude of the vector difference of unit normals (which we also used in the improved formulation), and the magnitude of the difference in the translational components d_i of canonical plane representations. Geometrically, the fourth component of the canonical plane vector is the perpendicular distance from the plane to the coordinate origin. Obviously d_i is not frame invariant, and neither is the difference $|d_i - d_j|$ between two distinct planes. This is easily seen by considering two perpendicular planes, and moving the origin along directions normal to each plane. Also, d_i may represent a distance from the origin to some point on the measured plane that is distant from the region that was visible to the camera, as will be the case if the surface is steeply inclined with respect to the camera view direction and the camera sits at the origin. In this situation, d_i is not well supported by the measured data, and its variance is much higher than that of less oblique views. This effect should be compensated for, but thus far we have no means of doing so.

We can obtain a metric that is both frame invariant and well supported by the data in a natural way by using the perpendicular distance of a point to a plane. For a canonical plane $\mathbf{q}$ and a point $\mathbf{p}$, this distance is simply $|\mathbf{q}'\mathbf{p}|$. If we have a point in the camera frame known to lie in a plane, we transform this point to the world frame and measure the perpendicular distance to the world representation of the plane. The metric thus has the form

$$\delta_i = \mathbf{q}'_W T_{RWi} T_{CR} \mathbf{p}_i \tag{43}$$

Writing the constituent transforms in terms of rotation and translation components as

$$T_{RWi} = \begin{bmatrix} W_i & \mathbf{w}_{ti} \\ \mathbf{0} & 1 \end{bmatrix} \qquad T_{CR} = \begin{bmatrix} R & \mathbf{r}_t \\ \mathbf{0} & 1 \end{bmatrix} \qquad \mathbf{q}_W = \begin{bmatrix} \mathbf{u}_w \\ d_w \end{bmatrix} \qquad \mathbf{p}_i = \begin{bmatrix} \mathbf{v}_i \\ 1 \end{bmatrix}$$

we expand the metric equation to obtain

$$\delta_i = \mathbf{u}'_w(W_i R\mathbf{v}_i + \mathbf{w}_{ti}) + \mathbf{u}'_w W_i \mathbf{r}_t + d_w \tag{44}$$

The metric is linear in the unknowns and is written for each viewpoint i in standard form $A_i\mathbf{x} + b_i = 0$ with

$$\mathbf{x} = \begin{bmatrix} \mathbf{r}_t \\ d_w \end{bmatrix} \qquad A_i = \begin{bmatrix} \mathbf{u}'_w W_i & 1 \end{bmatrix} \qquad b_i = \mathbf{u}'_w(W_i R\mathbf{v}_i + \mathbf{w}_{ti}) \tag{45}$$

The total error $\Delta = \sum_i \|A_i\mathbf{x} + b_i\|_2^2$ is minimized by linear least squares, and the solution is $\overline{\mathbf{x}} = -(\sum_i A'_i A_i)^{-1} \sum_i A'_i b_i$.

It now remains to determine a suitable point $\mathbf{p}_i$ at each viewpoint. Since we aim to choose a point that is well supported by the camera measurements, a reasonable selection is a point at the centre of the camera field of view. We call the line containing such points the line of sight. Recall that the data we acquire at each viewpoint consist of a set of samples of a surface from which we extract by segmentation those which correspond to the feature plane. We then find the plane parameters $\mathbf{q}_{Ci}$ which best fit the data - typically, in a least-squares sense - and use the resulting canonical plane representation as an atomic datum in our estimation process. Since we generally know the gaze direction of the camera expressed in its *own* frame, it is a simple matter to intersect the camera line of sight with the measured plane. We thus obtain the camera frame coordinates of a virtual point on the plane that is centered in the field of view. This point is known more precisely than any range data point because it is derived from many independent point measurements, and it is guaranteed to exist for any plane not containing a vector along the line of sight.

In our camera, the line of sight centred in the camera frame is simply the z-axis of the camera coordinate system. We therefore fix $v_x = v_y = 0$ and solve the plane equation for v_z. Defining $\mathbf{q}_{Ci} = [a_i \quad b_i \quad c_i \quad d_i]'$ and $\mathbf{v}_i = [0 \quad 0 \quad z_i]'$, we find $z_i = -(d_i/c_i)$. We can always introduce a change of coordinates such that any camera satisfies this model, provided that the line of sight is known in its natural frame. This will be useful in the following section, where we make use of the fact that $\mathbf{v}_i$ reduces to the scalar quantity z_i.

3.4. Computing Translation in Finite Memory. We now consider finite memory estimation of translation for the partitioned plane formulation. In the following discussion we derive the finite memory version of Method 2, as this is somewhat more challenging than that of Method 1. For the sake of brevity, we merely state that the same approach is used to derive a finite memory implementation of Method 1.

The least squares solution to system (45) is easily computed if we have available all of the original data T_{RWi} and $\mathbf{p}_i$ at each viewpoint at the time when we fix R and $\mathbf{u}_W$ from the rotational estimator. To execute this computation in constant memory for any number of views we must we perform the summations $\sum_i A_i'A_i$, $\sum_i A_i'b_i$ and $\sum_i b_i^2$ as the viewpoint data arrives and discard the raw data. The fact that the solution of the rotation estimator $[R \quad \mathbf{u}_W]$ appears in A_i and b_i makes this somewhat difficult. The terms required are

$$\begin{aligned}
A_i'A_i &= \begin{bmatrix} W_i'\mathbf{u}_W\mathbf{u}_W'W_i & W_i'\mathbf{u}_W \\ \mathbf{u}_W'W_i & 1 \end{bmatrix} \\
A_i'b_i &= \begin{bmatrix} W_i'\mathbf{u}_W\mathbf{u}_W'W_iR\mathbf{v}_i + W_i'\mathbf{u}_W\mathbf{u}_W'\mathbf{w}_{ti} \\ \mathbf{u}_W'W_iR\mathbf{v}_i + \mathbf{u}_W'\mathbf{w}_{ti} \end{bmatrix} \\
b_i^2 &= (\mathbf{u}_W'W_iR\mathbf{v}_i)^2 + 2(\mathbf{u}_W'W_iR\mathbf{v}_i)(\mathbf{u}_W'\mathbf{w}_{ti}) + (\mathbf{u}_W'\mathbf{w}_{ti})^2
\end{aligned} \tag{46}$$

where b_i^2 is used only if we desire to know the RMS projection error at the optimal solution. We would like to bring $\mathbf{u}_W$ and R outside the summation so that these functions can be evaluated on demand in a constant number of operations. The required factorization is not immediately obvious, but it can be done succinctly based on the following result.

In general, given matrices M, Q and vector $\mathbf{n}$, which are $n \times m$, $n \times q$ and $n \times 1$ respectively, we form the product

$$P = M'\mathbf{n}\mathbf{n}'Q$$

Expand the elements of P in terms of the column vectors $\mathbf{m}_j$ and $\mathbf{q}_k$ of M and Q, and use the fact that

$$(\mathbf{m}_j'\mathbf{n})(\mathbf{n}'\mathbf{q}_k) = (\mathbf{n}'\mathbf{q}_k)(\mathbf{m}_j'\mathbf{n}) = \mathbf{n}'(\mathbf{q}_k\mathbf{m}_j')\mathbf{n}$$

since scalar multiplication commutes. Now P can be written component-wise as

$$P_{jk} = \mathbf{n}'(\mathbf{q}_k \mathbf{m}'_j)\mathbf{n} \tag{47}$$

Using this result we can compute $\sum_i P_i$ component-wise as $\sum_i (P_i)_{jk} = \mathbf{n}'(\sum_i \mathbf{q}_k \mathbf{m}'_j)\mathbf{n}$, which is what we require for a finite memory computation of the estimated translation.

We also use the fact that the term $R\mathbf{v}_i$ collapses to a vector-scalar multiplication under our construction of $\mathbf{p}_{Ci}$. Since $\mathbf{v}_i = [0 \quad 0 \quad z_i]$ with $z_i = -d_i/c_i$, we have

$$R\mathbf{v}_i = \mathbf{r}_3 z_i$$

Now we can proceed to factor (46) and obtain

$$\begin{aligned}
\sum_i A'_i A_i &= \begin{bmatrix} M & (\sum_i W'_i)\mathbf{u}_W \\ \mathbf{u}_W{}'(\sum_i W_i) & n \end{bmatrix} \\
\sum_i A'_i b_i &= \begin{bmatrix} E\mathbf{r}_3 + \mathbf{e} \\ \mathbf{u}_W{}'(\sum_i z_i W_i)\mathbf{r}_3 + \mathbf{u}_W{}' \sum_i \mathbf{w}_{ti} \end{bmatrix} \\
\sum_i b_i^2 &= \mathbf{r}'_3 F \mathbf{r}_3 + 2\mathbf{r}'_3 \mathbf{f} + \mathbf{u}_W{}'(\sum_i \mathbf{w}_{ti} \mathbf{w}_{ti}')\mathbf{u}_W
\end{aligned} \tag{48}$$

where n is the number of views, and matrices M, E, F and vectors $\mathbf{e}, \mathbf{f}$ are defined component-wise as

$$\begin{aligned}
M_{jk} &= \mathbf{u}_W{}'(\sum_i \mathbf{w}_k \mathbf{w}'_j)\mathbf{u}_W \\
E_{jk} &= \mathbf{u}_W{}'(\sum_i z_i \mathbf{w}_k \mathbf{w}'_j)\mathbf{u}_W \\
F_{jk} &= \mathbf{u}_W{}'(\sum_i z_i^2 \mathbf{w}_k \mathbf{w}'_j)\mathbf{u}_W \\
\mathbf{e}_j &= \mathbf{u}_W{}'(\sum_i \mathbf{w}_{ti} \mathbf{w}'_j)\mathbf{u}_W && (49) \\
\mathbf{f}_j &= \mathbf{u}_W{}'(\sum_i z_i \mathbf{w}_{ti} \mathbf{w}'_j)\mathbf{u}_W && (50)
\end{aligned}$$

All summands in the above expressions are functions of the data at view i only. By accumulating the appropriate sums we can easily evaluate the minimizing solution $[\mathbf{r}'_t \quad d_w]$ and the RMS error $\sqrt{\Delta/n}$ for any given R and $\mathbf{u}_W$, using only finite computing resources for any number of views.

CHAPTER 4

Experimental Results

In the preceding chapter we have developed three estimation schemes for the range camera hand-eye transform. Here we report the results of our experimental analysis of estimator performance. Simulation is used extensively to characterize the behavior of the estimators against a known ground truth model. We then demonstrate the partitioned estimator for plane features using real data gathered from the active vision workcell in our laboratory, where the true parameters are unknown.

1. Selecting Viewpoints

An important issue in maximizing estimator accuracy is the selection of a set of viewpoints which adequately constrain the solution. Given an some current set of viewpoints, we seek a rule to determine the next viewpoint so as to best constrain those modes of the estimate which are most uncertain. Equivalently, we would like to choose a trajectory that maximizes the observability of the parameters we are attempting to estimate. This style of viewpoint selection strategy has been proposed and explored at length by Whaite and Ferrie [38, 39, 40, 37, 36] in the context of autonomous exploration. In their work the unknown model parameters describe objects in the workspace, and the goal is to collect data from a set of viewpoints so as to most rapidly reduce parametric uncertainties. The similarity to our present viewpoint selection problem is immediately apparent, but application of these methods is nontrivial.

A minimal requirement for generating a useful viewpoint trajectory is avoidance of degenerate viewpoint combinations, *i.e.* trajectories which constrain the estimates very poorly. Intuitively we expect that recovery of 6 DOF rigid transformations requires that we also drive the camera/robot complex through 6 DOF motions. Experimentally we have found that trajectories composed only of pure rotations, pure translations, rotations about a fixed axis or similar simple geometries are typically degenerate, in the sense that numerical condition of the resulting estimate is very poor. By contrast, randomly chosen trajectories tend to constrain the estimates well, since the probability of any particular geometric degeneracy is infinitesimally small. An appropriate heuristic is therefore to choose random viewpoints subject to feasibility constraints, and to avoid trajectories which are confined to simple geometric surfaces such as planes or spheres.

2. Generating Semi-random Trajectories

Our simulation of the camera/robot system requires that we generate a series of camera viewpoints that suitably constrain the estimators. We ensure that viewpoints generated by the simulated trajectory are actually feasible for a real camera, by disallowing views where the feature target is outside the range of view of a *reasonable* camera model. The fact that one side of an object is often occluded by the surface on which it rests is also incorporated. Finally, the resulting displacements approximate the feasible displacements of a real manipulator. Within the bounds of these constraints, the desired trajectory is determined randomly.

The semi-random viewpoint generator is constructed as follows. We begin by choosing the true world co-ordinates of the target feature, which is either a point or a plane. We also choose a true value for the simulated camera hand-eye transformation T_{CR}. For each desired viewpoint we place the simulated camera on the surface of a hemisphere centered at the feature point, or in the case of plane features we arrange for the centre of the hemisphere to lie in the target plane. The hemisphere radius is randomly selected in an interval appropriate to the manipulator and camera envisaged. Longitude and elevation angles for the camera position on the hemisphere are selected from uniform distributions over feasible ranges. Finally, we perturb the camera orientation by random tilt, pan and twist angles with respect to a nominal radial gaze direction. With this determination of camera orientation and position it is a simple matter to compute the robot end-effector position from T_{CR}, as well as the camera frame projection of the target feature. The resulting semi-random sequence of views typically constrains the estimator very well, while respecting many of the feasibility constraints imposed on real manipulators and sensors. The actual values used in our simulation appear in Table 4.1.

Parameter	Value	Units
true camera orientation R_{CR} (Euler angles)	-83.0, -1.9, -91.00	deg
true camera translation $\mathbf{t}_{CR}$	47.0, 37.0, 233.0	mm
true feature point coordinates	100.0, -200.0, 150.0	mm
true feature plane normal	-0.1078, 0.2157, -0.9705	-
true feature plane offset	-1078.3	mm
viewpoint longitude	0 - 360	deg
viewpoint elevation	25 - 90	deg
view sphere radius	250.0 - 750.0	mm
camera tilt	-20 - 20	deg
camera pan	-20 - 20	deg
camera twist	0 - 360	deg

Table 4.1. Simulation parameters

In order to simulate model and measurement error in the kinematic loop, we introduce an additional homogeneous transform between the robot end-effector and the simulated camera. This transform is nominally the identity, and represents the difference between the actual and observed positions of the end-effector. Translational disturbances are created by adding a random translation vector to the transform, where this vector is derived from a multivariate normal distribution. To simulate rotational disturbances we derive an axis of rotation from uniform latitude and longitude angle distributions, and then rotate about that axis by a normally distributed random angle. Note that the magnitude of the total disturbance is fully described by two quantities: the variance of the translational disturbance $\sigma_t^2 = \sigma_x^2 + \sigma_y^2 + \sigma_z^2$, and the variance of the angle of rotation σ_r^2.

3. Simulation Results

3.1. Condition Indicators. The condition number of matrix H in equation (12) is an indicator of how well a given set of viewpoints constrains the estimated parameter set. Figure 4.7 shows typical values of the condition number σ_n/σ_1 for the point formulation under both noiseless and moderately noisy simulated conditions. With fewer than five viewpoints, the numerical condition of the system is extremely poor, and we make no attempt to obtain a solution. The condition improves to greater than 10^{-10} after five views, which is sufficient for computation of a stable linear estimate. The direct plane formulation exhibits a similar behavior, rising to 10^{-5} after five views. From a strictly computational point of view these are rather low condition numbers, and we generally apply the technique of column-weighting before attempting to invert matrix H [9]. We have observed improvements of numerical condition up to five orders of magnitude by this technique, resulting in a more stable computation. We also use double precision arithmetic throughout to avoid loss of precision due to numeric underflow.

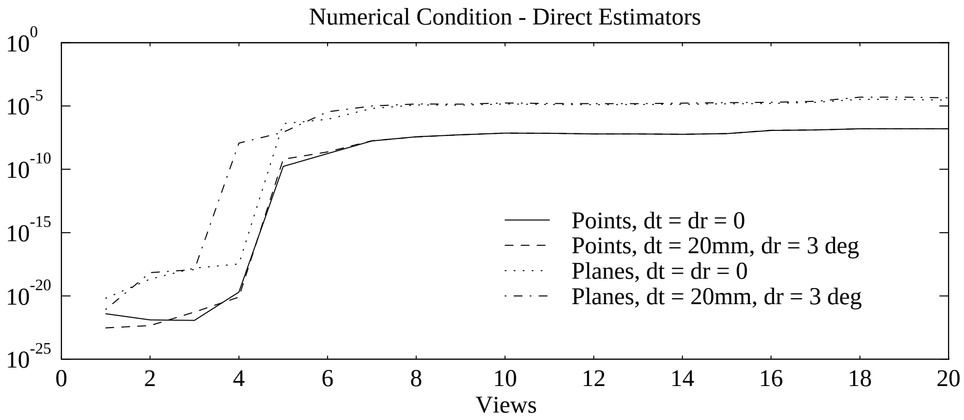

Figure 4.7. Numerical condition (σ_n/σ_1) of the direct estimators, for simulated data. *dt* is the standard deviation of the magnitude of the translational disturbance signal, and *dr* is the equivalent measure for the rotational disturbance (see text). The number of viewpoints supporting the estimate appears on the horizontal axis. Note that views are discrete events, and that we show connected curves for clarity of presentation.

The condition behavior of the direct plane formulation is also a function of the metric scaling constant κ, which suggests that we might attempt to choose κ so as to maximize the condition number for a given view combination. Unfortunately this rule does not generally yield good estimates. We have found empirically that a more useful guide to the selection of κ is obtained by minimizing the constraint error $\|\mathbf{G}(\mathbf{x}_l)\|$ obtained for the linear approximation $\mathbf{x}_l$.

In the case of the partitioned plane estimator there are two useful condition indicators. The condition number of H should be small, as we require that H be singular in order to yield a useful solution. We are also interested in a measure of uniqueness of the null direction of H. Since the singular values of a matrix represent gains of the system along particular basis directions, we expect the gain in the (unique) null direction to be significantly smaller than that in any orthogonal direction. We therefore define the detectability of the null direction as the ratio of the two smallest singular values σ_{12}/σ_{11}, where the singular values are ordered from largest to smallest. When detectability is close to 1.0, the unique null direction is not well defined and we can expect that the linear orientation estimate will be poor. Figure 4.8 shows that the null direction of H is discriminated by about three orders of magnitude after four views in the case of moderately noisy data, which is sufficient to provide a reliable initial estimate of the rotational unknowns.

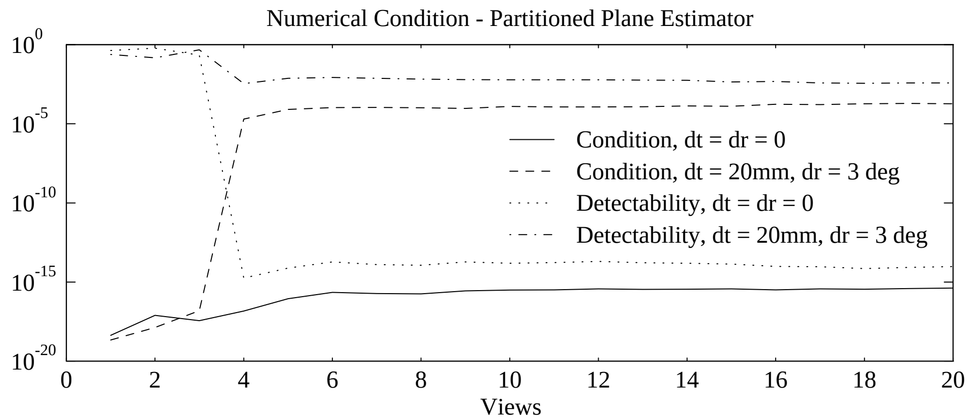

Figure 4.8. Condition number and detectability (see text) of the rotation part of the partitioned plane estimator. With at least four views the null space of the linear system is well defined by a single direction corresponding to the smallest singular value.

3.2. Residual errors. The RMS residual error provided by an estimator is useful to us in two ways. We noted in Chapter 3 that we expect the residual of a constrained estimator to be somewhat higher than that of the corresponding unconstrained linear approximation, and the difference between these values to decrease as more views are acquired. Residuals that do not behave in this fashion suggest that the projection model used in the estimator is inappropriate, which may result from inaccurate intrinsic calibration of the camera or manipulator, or from variation of quantities that are assumed constant.

In addition to providing diagnostic information, the residuals also yield an estimate of the mean projection error that can be expected when we use the estimated camera transform to compute the world frame representations of camera frame observations. The error metric of the point feature formulation has pure dimensions of length, and the RMS residual represents the expected error in the world frame coordinates of a point when that point is projected by the hand-eye transform and the manipulator kinematic model. This provides an independent performance measure for the combined camera/manipulator system over a specific part of the workspace. Residuals of the partitioned plane estimator are interpreted in a similar fashion. The utility of direct plane formulation residuals is not as clear, since these are not dimensionally pure.

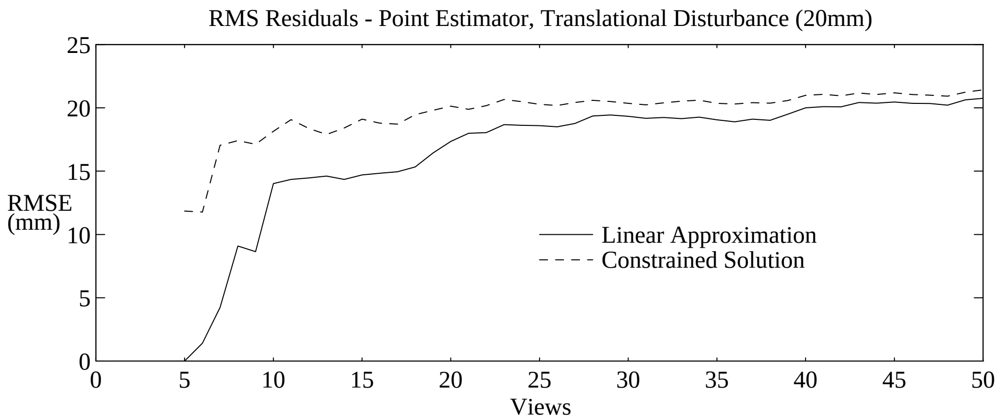

FIGURE 4.9. Simulated RMS residual errors for the point formulation, using a semi-random trajectory. A normally distributed translational disturbance ($\sigma_t = 20mm$) is applied to the end effector at each view. Measurements of the rotational component of end effector pose are noiseless.

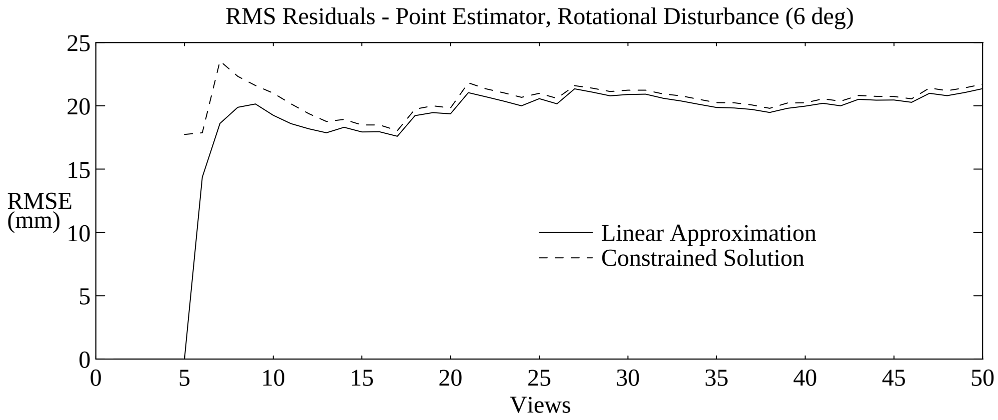

FIGURE 4.10. Simulated RMS residual errors as in Figure 4.9, with a rotational disturbance ($\sigma_r = 6°$) applied to the end effector. As in the case of translational disturbances the linear and constrained residuals converge as more views are obtained, indicating that the constraint model is consistent with the data.

Figure 4.9 shows a typical simulation sequence of residual errors for the point formulation using a semi-random trajectory. The end effector is subject to a translational disturbance with a standard deviation of $20mm$. The constrained residual is initially much larger than the linear one, but these rapidly converge to yield a stable estimate of the true disturbance magnitude. In the case of rotational disturbances (Figure 4.10), convergence to a fixed value occurs similarly. Note that rotational disturbances map into point measurement errors according to $\delta t = r\delta\theta$, where r is the distance from the end effector to the target point and which is varying randomly in our simulation.

3.3. True parameter errors. The performance of an estimation scheme is best measured by its ability to recover the true values of a set of unknowns in the presence of measurement noise. In simulation the actual parameter values are known, so we can determine parameter errors with respect to ground truth. This information is valuable in evaluating performance of the method, and also in interpreting the results of real data experiments where ground truth is unknown.

True parameter errors for the point feature estimator are shown in Figure 4.11. Parameter errors are initially of roughly the same order as the disturbance magnitudes, and decrease rapidly over the first fifteen views. This underscores the value

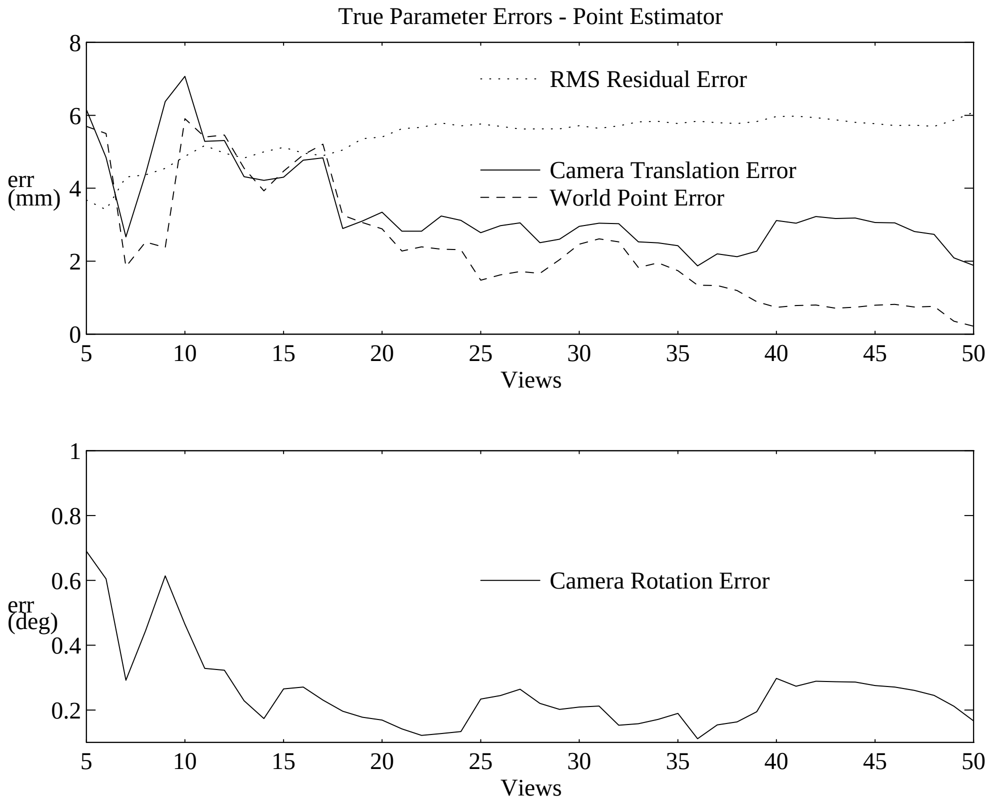

Figure 4.11. Magnitudes of the true parameter errors of the point feature estimator, with $\sigma_r = 1°$, $\sigma_t = 5mm$. We also show the RMS residual error, which indicates the combined effects of rotational and translational disturbances on world frame point projections for a typical viewpoint.

of redundant measurements, since schemes based only on *sufficient* data to constrain the parameters would yield the same results as our five-view estimate. As more data is collected the incremental improvements diminish, and in this simulation there is little practical benefit in collecting more than thirty views. The point past which additional views bring insignificant improvement is determined by the disturbance amplitude, as well as by the perceived cost of collecting views.

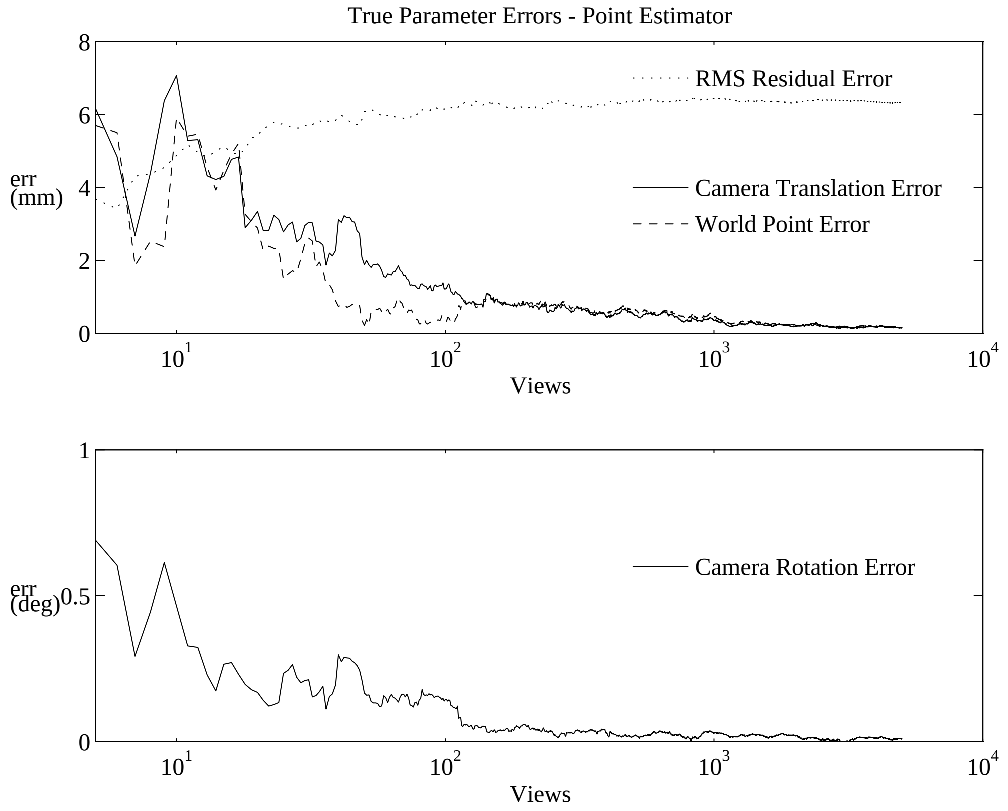

FIGURE 4.12. Evolution of the true parameter errors of Figure 4.11 over 5000 viewpoints. Parameter errors decay very slowly, in a roughly exponential manner. Note the logarithmic scale on the horizontal axis.

Despite the diminishing returns apparent in Figure 4.11, we expect the estimated values to converge to the true values in the limit of increasing viewpoints. Figure 4.12 shows that estimation errors do indeed drop significantly as data are added, although at an ever decreasing rate. After 5000 views, true parameter errors for the camera transform are less than $0.02°$ and $0.1mm$.

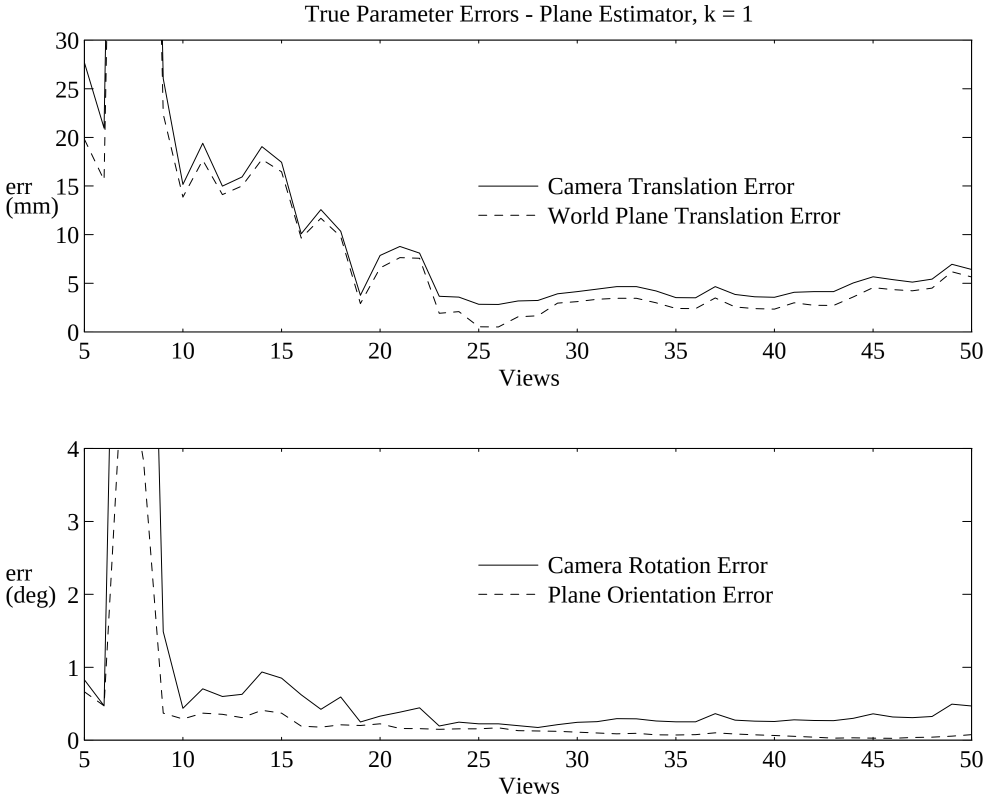

FIGURE 4.13. True parameter errors for the direct plane formulation, in moderate noise conditions ($\sigma_r = 1°, \sigma_t = 5mm$), and with length scaling factor $\kappa = 1.0$. The linear approximation is very poor with less than ten views, and convergence of the constraint enforcement iteration is slow. As more data becomes available the linear solution improves, convergence is more rapid, and ground truth parameter errors become comparable to the sensor noise level.

True parameter errors for the direct plane formulation with length scaling factor $\kappa = 1$ appear in Figure 4.13. With fewer than ten views the linear estimate is unreliable, and the constrained optimization typically requires ten to twenty iterations to achieve a rather poor solution. The situation improves rapidly, and after 25 views a stable and fairly accurate estimate of the unknowns is obtained. The value of κ strongly influences both quality of the linear approximation and accuracy of the estimated parameters. Figure 4.14 shows results over the same data set with $\kappa = 0.01$, which is a value suggested by the observation that there are approximately two orders of magnitude between typical distance measurements (expressed in millimeters) and components of unit normals (order of 1.0). This yields reasonable solutions with as

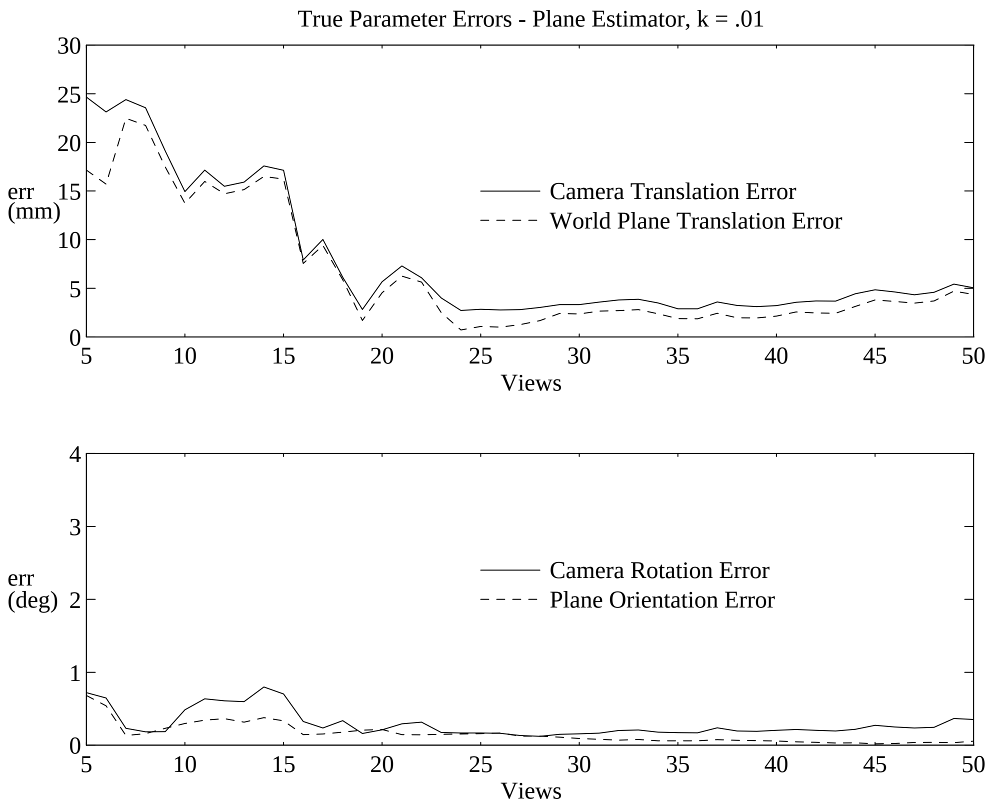

Figure 4.14. True parameter errors for the direct plane formulation, with the data set of Figure 4.13 and $\kappa = 0.01$. Performance of the estimator is significantly improved, particularly for small numbers of viewpoints.

few as five views, as well as reduced ground truth errors overall. The need to select a suitable κ based on typical dimensions of the problem at hand is a weakness of this formulation as the criterion for optimality of κ is unclear. Despite this limitation, parameter estimates with tolerances comparable to sensor noise can be obtained at the expense of collecting more views, with only a very coarse tuning of κ.

The partitioned plane estimator improves significantly on the direct formulation by eliminating the problem of choosing κ, as well as reducing overall ground truth parameter errors. Results of this method for the same data set as Figures 4.13 and 4.14 are shown in Figure 4.15. The linear estimates seed the iteration well, and convergence is typically obtained in less than four iterations. Here we use Method 1 to estimate the unknown translational components. Estimated parameters are reasonable for small numbers of views, and with twenty viewpoints the true values are recovered

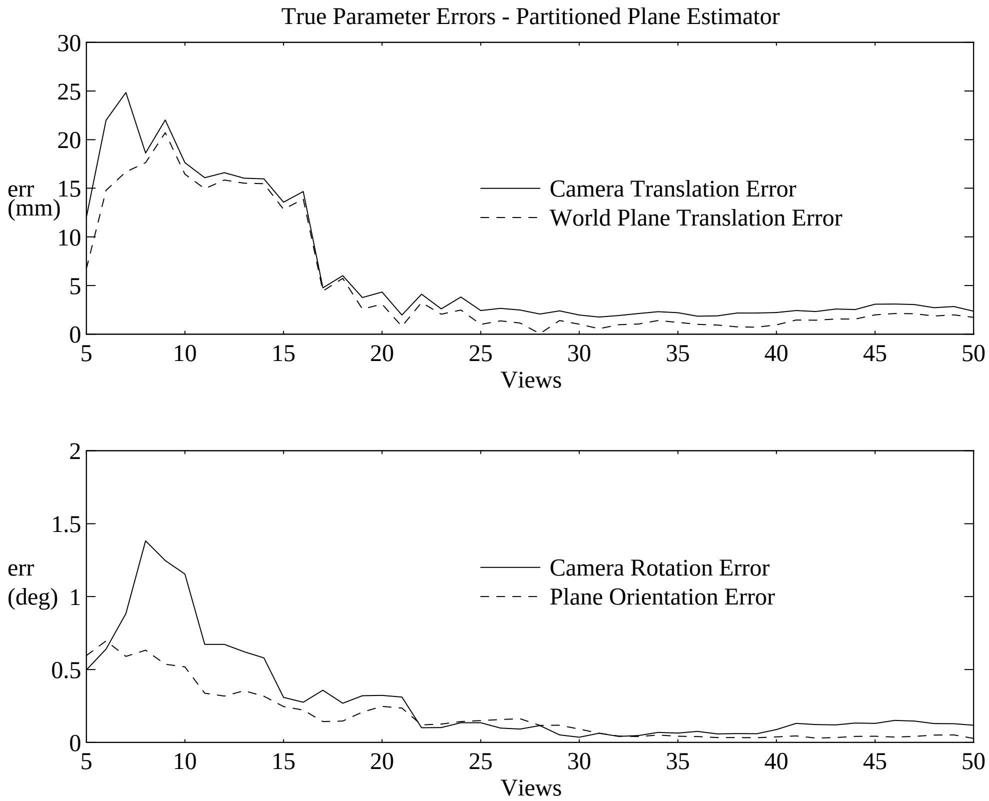

Figure 4.15. True parameter errors for the partitioned plane formulation, with the data set of Figure 4.13. Rotational parameter errors decay quickly, but propagation into the translation estimates causes relatively large errors there. As the rotation estimate is refined, translation parameter errors decrease accordingly.

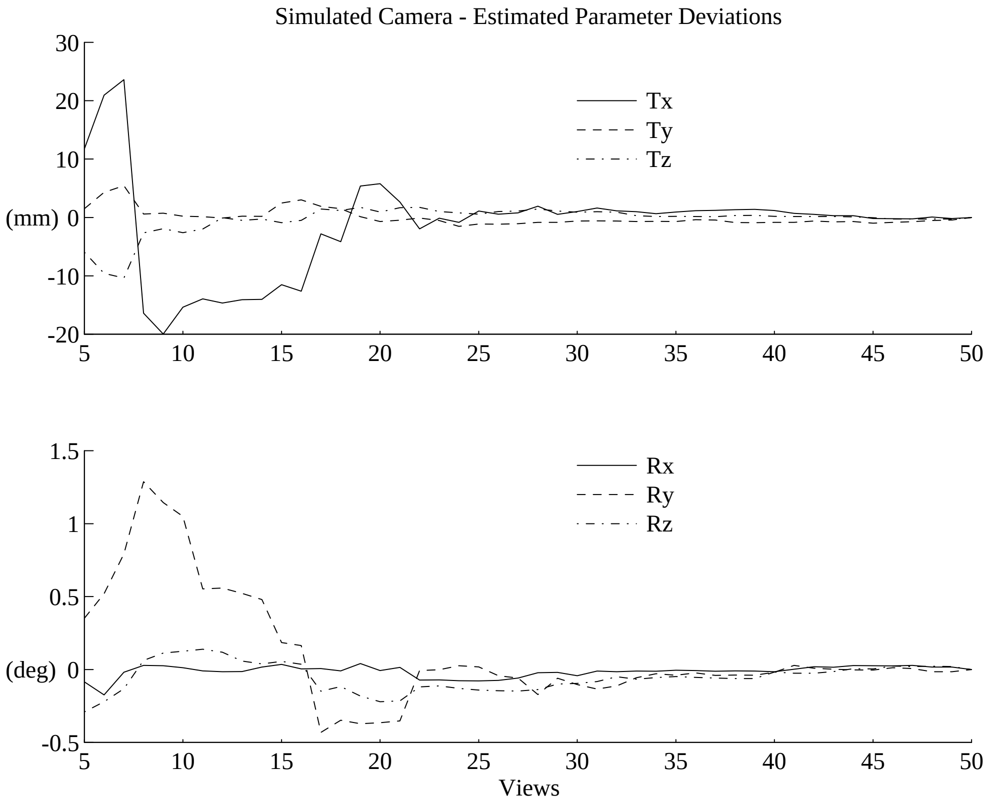

Figure 4.16. Time series behavior of the estimated camera parameters, with the data set of Figure 4.13. All parameters are shown as deviations about their stable values, which we take to be the values obtained with fifty viewpoints. Rotational parameters are expressed as $z - y - x$ Euler angles.

to a tolerance well within sensor noise. Translational parameter errors are large with respect to the point feature formulation for a given number of views (compare with Figure 4.11), but fall rapidly enough to yield useful results economically. The recovered parameters appear in Table 4.2. We show time series behavior of the recovered camera parameters in Figure 4.16, as this reflects stability of the estimate without reference to the ground truth parameters.

Parameter	Value	Units
camera orientation R_{CR} (Euler angles)	-82.93, -1.81, -90.92	deg
camera translation $\mathbf{t}_{CR}$	45.29, 35.95, 232.19	mm
feature plane normal $\mathbf{u}_W$	-0.1077, 0.2153, -0.9706	-
feature plane offset d_w	-1076.6	mm

Table 4.2. Recovered simulation parameters, $\sigma_r = 1°$, $\sigma_t = 5mm$, 50 views

The fact that translational parameter recovery is slower than the point feature case results from two issues. Translations in the partitioned plane estimator are solved subject to a rotation estimate that minimizes only orientation errors, and therefore propagates a bias to the translational estimate. In addition, plane features provide only one length-dimensioned datum per view as opposed to three per point measurement, so we expect the plane formulation to take somewhat longer to acquire equivalent translational information.

We have run numerous simulation experiments using Method 2 to estimate the translational parameters. In the case of noiseless data the unknowns are recovered correctly, and in the case of translational disturbances of the end effector the results are comparable to those of Method 1. However, Method 2 exhibits a very high sensitivity to rotational disturbances, and in the case of large perturbations generates very poor estimates. We do not completely understand why this formulation fails to generate reliable estimates, although it seems a likely consequence of the fact that we discard information by considering *only* the point on the line of sight in the translational error metric. Since Method 2 has failed to fulfill our expectations, the issue of finding an elegant solution to the plane metric problem remains for further study.

4. Vision workcell calibration results

Figure 4.17. Active vision workcell at the McGill Centre for Intelligent Machines. A scanning laser range finding camera is affixed to the end effector of a PUMA 560 robot. A table directly beneath the robot base provides a suitable surface for scene objects, and also serves as the reference plane in our calibration process.

The robot vision workcell in our laboratory is based on a PUMA 560 industrial robot and the NRC/McGill scanning laser range-finding camera. The base of the PUMA is attached to the ceiling directly above the workspace table, and the camera is rigidly fixed to the manipulator end-effector (Figure 4.17). With this configuration we can conveniently explore objects on or above the table surface in a useful working volume of several cubic meters.

We use the partitioned plane formulation to recover the hand-eye transformation of our mobile camera. The tabletop itself is used as the calibration feature. We select a series of views of the table surface using an interactive robot control interface, ensuring to avoid degenerate viewpoint combinations as described in Section 1. At each view position we acquire a 64×64 pixel range image of a part of the tabletop, and fit a plane to each image by linear least-squares. We also record at each position the robot base to end-effector transformation T_{RW}, which is calculated from known kinematics of the PUMA by the RCCL [17] robot control system. We thus obtain a stream of $(T_{RWi}, \mathbf{q}_i)$ pairs which are provided as input to the partitioned plane estimator.

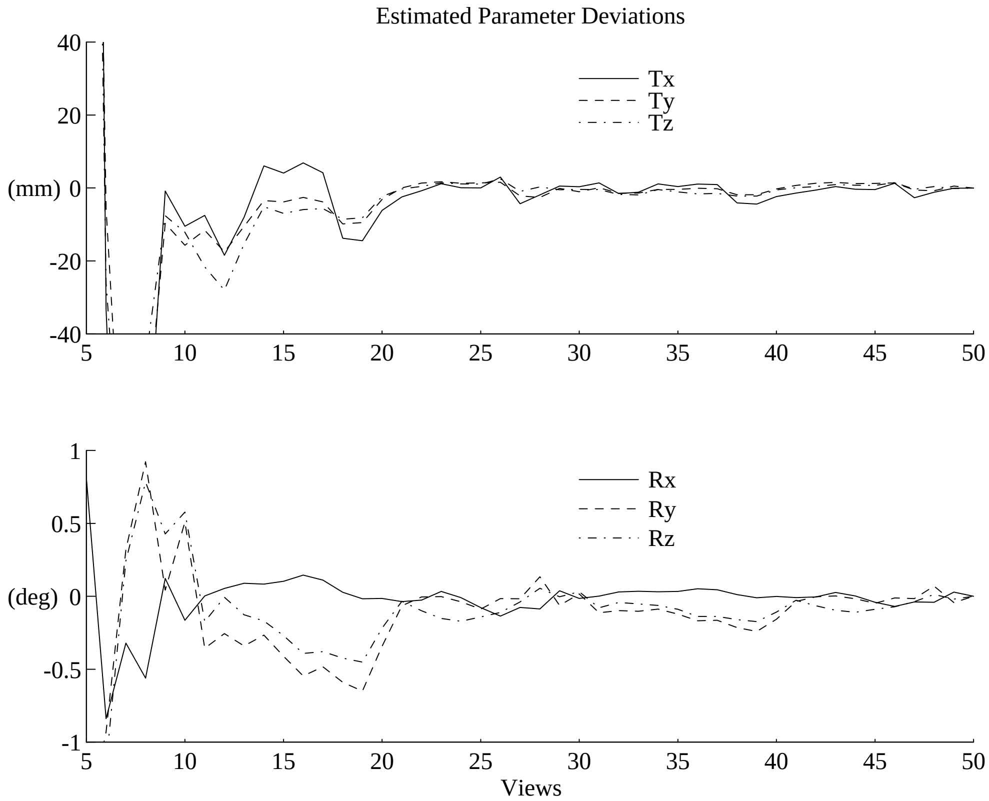

FIGURE 4.18. Deviations of the recovered camera parameters against increasing numbers of views. The rotational parameters are Euler angles.

Figure 4.18 shows deviations of the recovered parameters against increasing numbers of views, for a typical trajectory. Estimates based on fewer than ten views are poorly constrained, indicating that the data are fairly noisy. After twenty views the situation has stabilized significantly, and the recovered camera pose parameters remain confined within a band of approximately $\pm 0.2°$ and $\pm 4mm$. Rotational RMS residual errors for the linear and constrained solutions are very close and nearly constant after twenty views, at a value of $1.4°$. The recovered parameter values appear in Table 4.3.

Parameter	Value	Units
camera orientation R_{CR} (Euler angles)	-90.43, -0.39, -80.24	deg
camera translation $\mathbf{t}_{CR}$	21.1, -13.7, 213.1	mm
feature plane normal $\mathbf{u}_W$	-0.0175, 0.0252, -0.9995	-
feature plane offset d_w	-896.5	mm

Table 4.3. Recovered parameters for the active vision workcell

These figures indicate relatively high noise conditions. In particular, the RMS rotation residual error suggests that we can expect end-effector orientation errors of the order of a degree or more, which at a typical end-effector to scene distance of $500mm$ results in translational perturbations of roughly $10mm$. We are therefore not surprised that uncertainty in the camera translational parameters is of the order of a few millimeters, as this is consistent with comparable simulation results.

The RMS orientation error also serves to predict typical performance of the workcell in terms of its ability to accurately integrate data from multiple viewpoints using only the recovered hand-eye parameters and robot kinematics. Under similar operating conditions we should expect an absolute orientation error in the range of $1.5°$, which for many purposes is considered adequate. Displacement errors in view integration vary with the end-effector to target distance as $r \sin(1.5°)$, and are easily $20mm$ or more for typical distances. This is clearly a large error if we are imaging objects with dimensions of a few hundred millimeters, and illustrates the need for either (i) very high precision manipulators, or (ii) fine-grained view correspondence algorithms based on local feature matching. Figure 4.19 shows a typical integration of two views of an object using the estimated hand-eye transform. While there is a

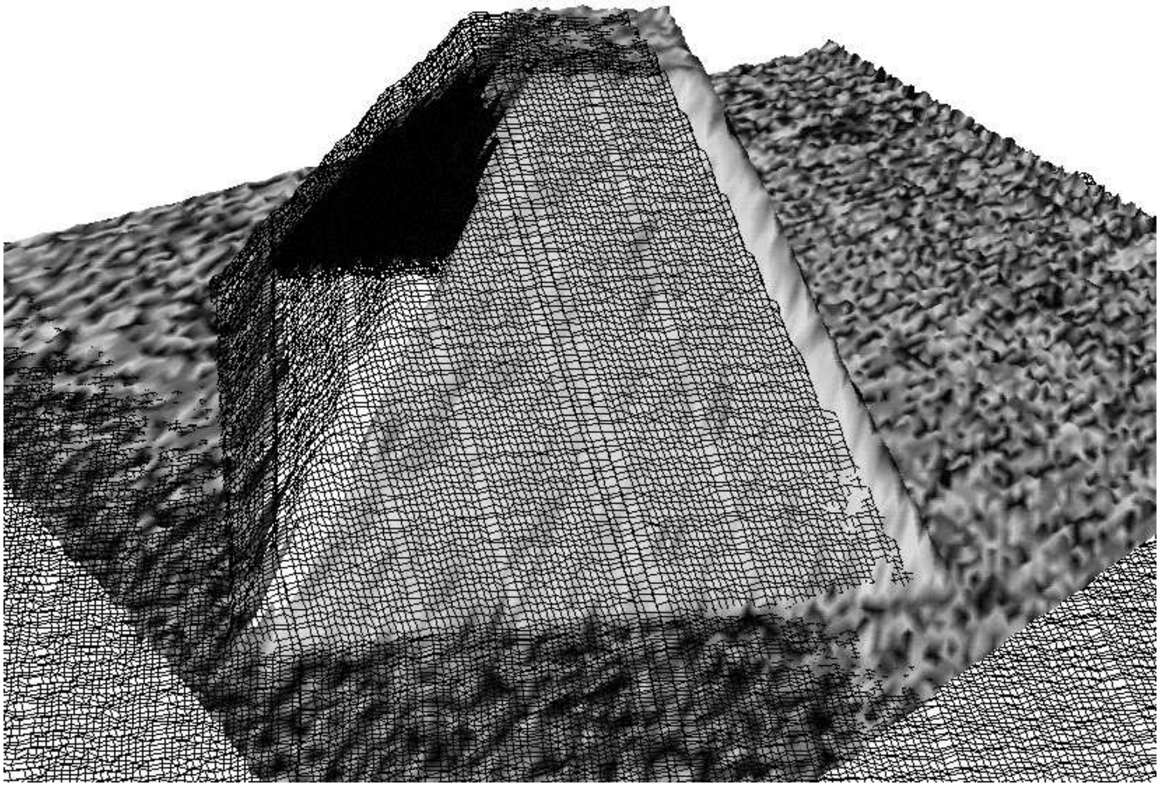

Figure 4.19. Results of a typical view integration, using the recovered camera hand-eye transform. The pyramid is $130mm$ wide at the base and $30mm$ at the top, and was imaged from two viewpoints roughly $500mm$ from the object. One view is rendered here as a shaded surface, while the other is shown as a grid. Displacement errors between the two projections of the pyramid are about $10mm$.

significant displacement error the result is as good as we can expect considering the accuracy of the manipulator, and is quite satisfactory for coarse integration.

CHAPTER 5

Conclusion

We have presented a class of solutions to the hand-eye calibration problem for range-finding cameras that is strongly motivated by practical considerations. Our approach relies on measurements of scene features that are common in typical scenes and that are easily extracted, rather than depending on results of difficult pose estimation or correspondence problems as input to our procedure. We emphasize the value of redundant measurements for suppressing noise, and visualize dynamic calibration as an ongoing refinement of the measurement model that maximizes self-consistency of the acquired data. Our solutions admit finite memory implementations, and can be applied as filter processes on infinite input data streams.

Our development has focussed primarily on a particular calibration task, but our approach is applicable to a broad range of problems. We have used these methods in our laboratory to determine the base frame transformation between two co-operating robots, where one is used to manipulate the range-finding camera, and the other performs grasping tasks within the common workspace. Our solutions for the hand-eye calibration problem are directly applicable to the two-robot problem by a simple reorganization of the input data. We are presently investigating the application of a similar method to hand-eye calibration of a range camera mounted on a mobile rolling robot. We expect this problem to require the addition of constraints reflecting the fact that this manipulator moves in the plane, but the essential principle remains unchanged. We have also explored applications in the calibration of hand-held ultrasound sensors. In this case the measurements are samples of space curves instead of surfaces, but again our approach is directly applicable.

Several issues remain for further study. The most significant and challenging of these is the matter of generating manipulator trajectories so as to optimally reduce uncertainty in the recovered parameters. This is particularly difficult because of the need to accommodate a wide range of constraints arising from physical limitations of the sensor and manipulator. Despite good results obtained with the partitioned plane formulation, we are not entirely satisfied with the translation error metric and believe that some improvement can be made here. In particular, further analysis of the failed frame-invariant formulation for translations is required.

The idea of dynamic self-calibration presented in this thesis is a general concept of which we have explored but a single example. We have demonstrated a practical means of closing the feedback loop around a visual exploration process, thereby embedding in the system a capacity for ongoing self-validation and self-adjustment. This capability is vitally important for practical autonomous mechanisms, and we will require a broad selection of computational methods in order to build artificial perception systems whose adaptability and robustness even remotely approach those of the human model.

REFERENCES

[1] J. Angeles. On the use of invariance in robotics-oriented redundant sensing. In *Proc. IEEE International Conference on Robotics and Automation*, pages 599–604, 1989.

[2] M. Armstrong. *Basic Topology*. Springer-Verlag, 1983.

[3] D. Bennett and J. Hollerbach. Autonomous calibration of single-loop closed kinematic chains formed by manipulators with passive endpoint constraints. *IEEE Transactions on Robotics and Automation*, 7(5):597–606, October 1991.

[4] S. Chaudhuri and S. Chatterjee. Performance analysis of total least squares methods in three-dimensional motion estimation. *IEEE Transactions on Robotics and Automation*, 7(5):707–714, October 1991.

[5] J. Denavit and R. Hartenberg. A kinematic notation for lower-pair mechanisms based on matrices. *ASME Journal of Applied Mechanics*, pages 215–221, June 1955.

[6] O. Faugeras and S. Maybank. Motion from point matches multiplicity of solutions. *International Journal of Computer Vision*, 4:225–246, 1990.

[7] L. Foulloy and R. Kelley. Improving the precision of a robot. In *Proc. IEEE International Conference on Robotics and Automation*, pages 62–67, Atlanta, Georgia, March 1984.

[8] G. Golub and C. Van Loan. An analysis of the total least squares problem. *SIAM J. Num. Anal.*, 17:883–893, 1980.

[9] G. Golub and C. Van Loan. *Matrix Computations*. Johns Hopkins University Press, Baltimore, Maryland, second edition, 1989.

[10] W. Hamilton. *Elements of Quaternions*. Cambridge University Press, Cambridge, 1899.

[11] J. Hollerbach. A review of kinematic calibration. In O. Khatib, J. Craig, and T. Lozano-Perez, editors, *The Robotics Review 1*, pages 207–242. MIT Press, Cambridge, Mass., 1989.

[12] B. Horn. Closed-form solution of absolute orientation using unit quaternions. *J. Opt. Soc. Am. A*, 4(4):629–642, April 1987.

[13] B. Horn, H. Hilden, and S. Negahdaripour. Closed-form solution of absolute orientation using orthonormal matrices. *J. Opt. Soc. Am. A*, 5:1127–1135, July 1988.

[14] X. Hu and N. Ahuja. Motion estimation under orthographic projection. *IEEE Transactions on Robotics and Automation*, 7(6):848–853, December 1991.

[15] T. Huang and C. Lee. Motion and structure from orthographic projections. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(5):536–540, May 1989.

[16] R. Lenz and R. Tsai. Calibrating a cartesian robot with eye-on-hand configuration independent of eye-to-hand relationship. In *Proc. IEEE Computer Vision and Pattern Recognition Conf.*, Ann Arbor, MI, June 1988.

[17] J. Lloyd and V. Hayward. *Multi-RCCL User's Guide*. McGill Centre for Intelligent Machines, Montreal, Quebec, Canada, release 4.2 edition, 1992.

[18] D. Luenberger. *Linear and Nonlinear Programming*. Addison-Wesley, Reading, Mass., second edition, 1984.

[19] H. Oswal and S. Balasurbramanian. An exact solution of absolute orientation. *Photogrammetric Engineering*, 34:1079–1083, 1968.

[20] F. Park. *The Optimal Design of Kinematic Mechanisms*. PhD thesis, Harvard University, Cambridge, MA, 1991.

[21] F. Park, A. Murray, and J. McCarthy. Designing mechanisms for workspace fit. In J. Angeles, G. Hommel, and P. Kovacs, editors, *Computational Kinematics*, pages 295–306. Kluwer Academic Publishers, 1993.

[22] R. P. Paul. *Robot Manipulators*. MIT Press, Cambridge, Mass., 1981.

[23] W. Press, B. Flannery, S. Teukolsky, and W. Vetterling. *Numeric Recipes in C.* Cambridge University Press, Cambridge, U.K., 1988.

[24] J.-M. Renders, E. Rossignol, M. Becquet, and R. Hanus. Kinematic calibration and geometrical parameter identification for robots. *IEEE Transactions on Robotics and Automation*, 7(6):721–732, December 1991.

[25] G. Schut. On exact linear equations for the computation of the rotational elements of absolute orientation. *Photogrammetria*, 16:34–37, 1960.

[26] Y. Shiu and S. Ahmad. Calibration of wrist-mounted robotic sensors by solving homogeneous transform equations of the form $AX = XB$. *IEEE Transactions on Robotics and Automation*, 5(1):16–29, February 1989.

[27] K. Shoemake. Animating rotations with quaternion curves. *Siggraph '85*, 19(3):245–254, 1985.

[28] G. Soucy. View correspondence using curvature and motion consistency. Master's thesis, Dept. of E.E., McGill Univ., 1991. To appear.

[29] K. Spring. Euler parameters and the use of quaternion algebra in the manipulation of finite rotations: A review. *Mechanism and Machine Theory*, 21(5):365–373, 1986.

[30] E. Thompson. An exact linear solution of the problem of absolute orientation. *Photogrammetria*, 15(4):163–179, 1959.

[31] M. Thompson, editor. *Manual of Photogrammetry.* American Society of Photogrammetry, Falls Church, V.A., 3 edition, 1966.

[32] R. Tsai. A versatile camera calibration technique for high accuracy 3d machine vision metrology using off-the-shelf TV cameras and lenses. *IEEE Journal of Robotics and Automation*, RA-3(4), August 1987.

[33] R. Tsai and R. Lenz. Real time versatile robotics hand/eye calibration using 3d machine vision. In *Proc. IEEE International Conference on Robotics and Automation*, pages 554–561, 1988.

[34] R. Tsai and R. Lenz. A new technique for fully autonomous and efficient 3d robotics hand/eye calibration. *IEEE Transactions on Robotics and Automation*, 5(3):345–358, June 1989.

[35] S. Ullman. *The Interpretation of Visual Motion.* MIT Press, Cambridge, Mass., 1982.

[36] P. Whaite and F. Ferrie. Autonomous exploration: Driven by uncertainty. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Oct. 1993. McGill Center for Intelligent Machines TR 93-17, Submitted.

[37] P. Whaite and F. Ferrie. Model building and autonomous exploration. In *SPIE - Intelligent Robots and Computer Vision XII: Active Vision and 3D Methods*, pages 73–85, Boston, Massachusetts, Sept.8–9 1993.

[38] P. Whaite and F. P. Ferrie. From uncertainty to visual exploration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(10):1038–1049, Oct. 1991.

[39] P. Whaite and F. P. Ferrie. Uncertain views. In *PROC. IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 3–9, Champaign, Illinois, June 15-18 1992.

[40] P. Whaite and F. P. Ferrie. Active exploration: Knowing when we're wrong. In *PROC. Fourth International Conference on Computer Vision*, pages 41–48, Berlin, Germany, May 11-14 1993. Computer Society of the IEEE, IEEE Computer Society Press.

[41] H. Zhuang and Z. Roth. Comments on "Calibration of wrist-mounted robotic sensors by solving homogenous transform equations of the form $AX = XB$". *IEEE Transactions on Robotics and Automation*, 7(6):877–878, December 1991.

Dynamically Estimating Mobile Range Camera Pose from Invariant Feature Measurements

Duncan L. Baird

M.Eng. Thesis April 1994
Dept. of Electrical Engineering Copy # 1

Dynamically Estimating Mobile Range Camera Pose from Invariant Feature Measurements

Duncan L. Baird

M.Eng. Thesis April 1994
Dept. of Electrical Engineering Copy # 2

Dynamically Estimating Mobile Range Camera Pose from Invariant Feature Measurements

Duncan L. Baird

M.Eng. Thesis April 1994
Dept. of Electrical Engineering Copy # 3

Dynamically Estimating Mobile Range Camera Pose from Invariant Feature Measurements

Duncan L. Baird

M.Eng. Thesis April 1994
Dept. of Electrical Engineering Copy # 4

Document Log:

Manuscript Version 0

Typeset by $\mathcal{A}\mathcal{M}\mathcal{S}$-LaTeX — 27 April 1994

Dynamically Estimating Mobile Range Camera Pose from Invariant Feature Measurements

Duncan L. Baird