Online Computation of Exterior Orientation with Application to Hand-Eye Calibration

Chien-Ping Lu
Eric Mjolsness
Gregory D. Hager

Research Report YALEU/DCS/RR-1046
August 1994

YALE UNIVERSITY
DEPARTMENT OF COMPUTER SCIENCE

Online Computation of Exterior Orientation with Application to Hand-Eye Calibration

Chien-Ping Lu, Eric Mjolsness and Gregory D. Hager
Department of Computer Science, Yale University
New Haven, CT 06520-8285

August 16, 1994

Abstract

Computation of the relative position and orientation between a camera and an observed object is a central problem in many vision-based robotics applications. Although many solution methods have been proposed, several problems remain. Nonlinear methods require considerable computation time and a good initial estimate, while linear methods, which do not need an initial estimate, are sensitive to noise and outliers. In this paper, we review a number of existing methods and present a new iterative algorithm that is globally convergent and is as computationally efficient as linear methods. Experiments on simulated and real data indicate that the new method is less sensitive to noise and outliers than other commonly used methods. We discuss the use of this method in the context of several vision-based applications.

Keywords camera calibration, exterior orientation, model-based object recognition, pose estimation

1 Introduction

Given a set of 3-D reference points in an object coordinate frame, and their 2-D perspective projections, the process of determining the rigid transformation that relates the object coordinate frame to that of the camera is referred to as *exterior orientation* or *hand-eye calibration* [21] when the goal is to locate the camera in the object coordinate frame, and as *object pose estimation* [17] or *object localization* [10] when the goal is to locate the object in the camera coordinate frame. The rigid transformation is called the *object pose* and its inverse the *camera pose*. Either object pose or camera pose can be called the camera *extrinsic parameters* or the *exterior orientation* depending on which coordinate frame is being referred to.

Exterior orientation plays a central role in many classical vision problems. For example, a popular paradigm in object recognition is to hypothesize a set of matches between a stored model and observed data, and then to perform an object pose calculation to verify the consistency of the match [26, 10]. Exterior orientation is an important part of camera calibration. The camera calibration problem is to determine the relationship between 3-D coordinates and their projections in a camera image. This relationship is comprised of the camera pose composed with a mapping that describes the physical attributes of the camera-lens system and the digitizing hardware [27, 31]. Because the physical attributes of the imaging system are independent of the camera pose, the parameters describing them are referred to as the camera *intrinsic parameters.*

Most reported exterior orientation methods perform well when the input data are accurate and well-behaved. Extremely accurate results can be achieved with precise 3-D models under controlled conditions. For this reason, camera calibration systems often use specially designed calibration patterns that are metrically accurate and have high contrast to enhance the performance of feature extraction.

In a completely static environment where the pose estimation needs to be done only once, the time and expense needed for highly accurate estimation can often be justified. However, many applications demand fast online pose estimation. For example, a recent paper [14] describes an application where an operator registers a geometric model with an image by pointing out specific model features in an image. Following this registration operation, features of the model are tracked and the pose of the model is updated in real time. A similar scheme could be used, for example, to compute a "movie" of a moving object to be rendered graphically. This sort of capability may be needed to support applications such as *Enhanced Reality* [7] where real-time graphical rendering of a model based on visual data is needed.

Recent progress in visual servoing has led to systems which make use of online calibration [4, 12, 13, 11, 20, 32]. It is done by tracking visual features of a manipulator as it performs a set of motions. Data is acquired in the form of 3-D positions computed using the robot inverse kinematics and 2-D image positions computed by feature tracking. Both of these sources suffer from errors. Robot inverse kinematics are notoriously imprecise, particularly on smaller, flexible robots. Feature tracking suffers from noise and localization bias. In addition to statistical errors, errors such as mechanical backlash must be tolerated. It can also be expected that the 3-D to 2-D correspondences will occasionally be incorrect due to operator error, mistracking, or similar problem.

This paper presents a new exterior orientation algorithm that is well-suited to problems requiring fast pose estimation or exterior orientation computation from noisy data. This algorithm, unlike most existing methods, minimizes 3-D feature position error (object space error) rather than 2-D image error. The objective function, though it is nonlinear, is optimized in full by efficient closed-form coordinate-wise optimizations. The new algorithm has been been compared to a number of previously existing methods including one linear algorithm that uses the perspective transformation matrix formulation [1, 33, 8], another that uses a radial alignment constraint [27, 25], and a nonlinear algorithm that uses classical nonlinear optimization. All methods have been tested on synthetic data with varying noise, percentages of outliers, and numbers of reference points. The methods have also been compared experimentally in the context of hand-eye calibration for a robot arm.

The remainder of this article is organized as follows. The next section introduces the pose estimation problem in more detail, relates it to camera calibration, and describes several existing methods for exterior orientation. Section 3 describes the new algorithm. Section 4 compares the algorithms on experimental and real data. Section 5 discusses further extensions of the method and outlines a set of interesting problems that remain to be addressed.

2 The Problem and Some Solution Methods

The mapping from 3-D points to 2-D image coordinates can be formalized as follows. Given a set of 3-D coordinates of reference points $\mathbf{X}_i = (x_i, y_i, z_i)^t, i = 1, \ldots N$ in an object coordinate frame, and the corresponding coordinates (the scene points) $\mathbf{Y}_i = (x_i', y_i', z_i')^t$ in a camera coordinate frame, the two frames can be related by a rigid transformation as

$$\mathbf{Y}_i = R\mathbf{X}_i + \mathbf{T}, \tag{1}$$

where

$$R = \begin{pmatrix} \mathbf{r}_1^t \\ \mathbf{r}_2^t \\ \mathbf{r}_3^t \end{pmatrix} \quad \text{and} \quad \mathbf{T} = \begin{pmatrix} t_1 \\ t_2 \\ t_3 \end{pmatrix} \tag{2}$$

are a rotation matrix and a translation vector, respectively.

We first assume that 3-D reference points, $\mathbf{X}_i$, are projected to the plane with equation $z = 1$, referred to as the *normalized image plane*, in the camera coordinate frame. The resulting projections $(u_i, v_i)^t$ are called the *normalized image coordinates.* Under the idealized pinhole imaging model, the image vector $\mathbf{y}_i = (u_i, v_i, 1)^t$, the scene point $\mathbf{Y}_i$, and the center of projection are collinear. The projection equation can be written as

$$\mathbf{y}_i = \frac{1}{\mathbf{r}_3^t\mathbf{X}_i + t_3}(R\mathbf{X}_i + \mathbf{T}), \tag{3}$$

which is known as the *collinearity equation* in photogrammetry literature.

The imaging geometry of real cameras is somewhat more complex than the pinhole model. This causes various kinds of optical distortions [9]. If only radial distortion is considered, the normalized image coordinates $(u_i, v_i)^t$ can be corrected from the distorted (uncorrected) ones $(\hat{u}_i, \hat{v}_i)^t$ by

$$u_i = \hat{u}_i(1 + \kappa r^2) \tag{4}$$

$$v_i = \hat{v}_i(1 + \kappa r^2), \tag{5}$$

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

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

$$\hat{u} = (m - m_0)/s_u \tag{6}$$

$$\hat{v} = (n - n_0)/s_v, \tag{7}$$

where (m_0, n_0) is the image center in sensor coordinates, and (s_u, s_v) are the horizontal and vertical scale factors, respectively.

The parameters s_u, s_v, m_0, n_0 and κ are referred to as the *camera intrinsic parameters.* In the discussion which follows, we assume that the camera intrinsic parameters are known, and normalized image coordinates can be computed from sensor coordinates accordingly.

We note that if the 3-D camera frame coordinates $\{\mathbf{Y}_i\}$ have been determined by some means, the process of determining R and $\mathbf{T}$ from $\{\mathbf{X}_i\}$ and $\{\mathbf{Y}_i\}$ is called *absolute orientation.* It can be generalized to include an unknown scaling factor, in which case (1) takes the form

$$\mathbf{Y}_i = sR\mathbf{X}_i + \mathbf{T}. \tag{8}$$

The least-squares minimization corresponding to equation (8) is known to have an exact closed-form solution [24, 2, 22, 28, 29].

2.1 Classical Methods

In the classical photogrammetry approach, the exterior orientation problem is formulated as that of minimizing the sum of square errors of the collinearity equation

$$\sum_i w_i \|\mathbf{y}_i - \frac{1}{\mathbf{r}_3^t\mathbf{X}_i + t_3}(R\mathbf{X}_i + \mathbf{T})\|^2. \tag{9}$$

This objective function is nonlinear and can only be solved by iterative nonlinear methods. Most methods for minimizing this objective function operate by iteratively linearizing the collinearity equation around the current approximate solution and solving the linearized system for the next approximate solution. These methods usually require a good starting point. The Gauss-Newton method has been applied to model-based object recognition [26]. It is reported in [18] that for the Gauss-Newton method to work, the initial approximate solutions have to be within 10% of scale for the translation and within 15^o for each of the three rotation angles. A detailed treatment on classical methods is available in [19].

2.2 Linear Methods

There are several "linear" approaches to the exterior orientation problem. In general, linear algebraic methods solve for the 9 parameters (or part of them) in the 3-by-3 transformation matrix linearly by ignoring the orthonormality constraint. The solution can then be improved by finding the orthonormal matrix that best fits the 3-by-3 matrix.

The Perspective Transformation Matrix (PTM) Method By ignoring the orthonormality constraint, the 3-by-3 rotation matrix $R = (r_{ij})$ and the translation vector $\mathbf{T}$ can be solved in closed form up to a scale factor. This method is adapted from the original PTM method [1, 33, 8] which uses the 4-by-3 perspective transformation matrix to encode the image center and image scales in addition to R and $\mathbf{T}$.

The collinearity equation (3) can be rewritten as

$$(\mathbf{r}_3^t\mathbf{X}_i + t_3)u_i = \mathbf{r}_1^t\mathbf{X}_i + t_1 \tag{10}$$

$$(\mathbf{r}_3^t\mathbf{X}_i + t_3)v_i = \mathbf{r}_2^t\mathbf{X}_i + t_2. \tag{11}$$

Dividing by t_3 on both sides, equations (10) and (11) become

$$((\mathbf{r}_3')^t\mathbf{X}_i + 1)u_i = (\mathbf{r}_1')^t\mathbf{X}_i + t_1' \tag{12}$$

$$((\mathbf{r}_3')^t\mathbf{X}_i + 1)v_i = (\mathbf{r}_2')^t\mathbf{X}_i + t_2', \tag{13}$$

where $\mathbf{r}_1' = t_3^{-1}\mathbf{r}_1, \mathbf{r}_2' = t_3^{-1}\mathbf{r}_2, \mathbf{r}_3' = t_3^{-1}\mathbf{r}_3, t_1' = t_3^{-1}t_1$, and $t_2' = t_3^{-1}t_2$. Each pair consisting of an image vector $\mathbf{y}_i = (u_i, v_i, 1)^\mathbf{t}$ and a reference point $\mathbf{X}_i$ contributes two linear equations ((12) and (13)) for solving for the 11 parameters $R' = (\mathbf{r}_1', \mathbf{r}_2', \mathbf{r}_3')^\mathbf{t}, t_1'$ and t_2'. A least-squares solution is found by singular value decomposition of the resulting system.

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

$$\det(R') = \det(t_3^{-1}R) = t_3^{-1}. \tag{14}$$

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

Now with known t_3 and R, the remaining unknowns t_1 and t_2 can be calculated by solving the overdetermined system of equations (10) and (11) for each pair of image vector and reference point. To improve the accuracy, t_3 can be recalculated together with t_1 and t_2.

The Radial Alignment Constraint (RAC) Method Tsai introduced a two-step method for camera calibration [27, 25]. The first stage makes use of the radial alignment constraint which can be formulated in our notation as

$$\frac{\hat{u}_i}{\hat{v}_i} = \frac{u_i}{v_i} = \frac{\mathbf{r}_1^\mathbf{t}\mathbf{X}_i + t_1}{\mathbf{r}_2^\mathbf{t}\mathbf{X}_i + t_2}. \tag{15}$$

Essentially, the radial alignment constraint says that the vector defined by the orthographic projection of a 3-D reference point is parallel to that of the corresponding distorted image coordinates under radial distortion. Using this formula, the horizontal scale factor and all of the camera pose parameters except for t_3 can be computed using linear techniques. The second stage computes the remaining parameters by applying nonlinear optimization to the collinearity equation (3) using the values computed in the first stage.

When computing only camera pose with known camera intrinsic parameters, we can use normalized image coordinates. We observe that (15) can be converted by division and cross-multiplication to

$$v_i(\mathbf{r}_1')^\mathbf{t}\mathbf{X}_i + v_i t_1' = u_i(\mathbf{r}_2')^\mathbf{t}\mathbf{X}_i + u_i, \tag{16}$$

where $\mathbf{r}_1' = t_2^{-1}\mathbf{r}_1, \mathbf{r}_2' = t_2^{-1}\mathbf{r}_2$, and $t_1' = t_2^{-1}t_1$. Each pair consisting of an image vector $\mathbf{y}_i = (u_i, v_i, 1)^\mathbf{t}$ and its corresponding reference point $\mathbf{X}_i$ contribute one linear equation (16) that can be used to solve for the 7 parameters $\mathbf{r}_1', \mathbf{r}_2'$, and t_1'. A least-squares solution is computed by singular value decomposition of the resulting system.

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

$$|t_2| = ||\mathbf{r}_1'||^{-1} \quad \text{or} \quad ||\mathbf{r}_2'||^{-1}. \tag{17}$$

The sign of t_2 also determines the signs of $\mathbf{r}_1, \mathbf{r}_2$ and t_1. It should be chosen such that u_i and v_i have the same sign as $\mathbf{r}_1^\mathbf{t}\mathbf{X}_i + t_1$ and $\mathbf{r}_2^\mathbf{t}\mathbf{X}_i + t_2$, respectively. The point $\mathbf{X}_i$ used to determine the sign of t_2 should be some reference point whose image point is far away from the image center. Using the orthonormality of the rotation matrix, $\mathbf{r}_3$ can be computed from $\mathbf{r}_1$ and $\mathbf{r}_2$.

With known t_1, t_2 and R, the remaining unknown, t_3, can be calculated by solving the overdetermined system of equations (10) and (11) for each pair of image vector and reference point. Again, to improve accuracy, t_1 and t_2 can be recalculated together with t_3.

Both the PTM and the RAC method are linear and noniterative. They are very fast and don't need initial guesses. However, it should be noted that they provide only an *approximate* closed-form solution. The orthonormality constraint on rotation matrices is not fully considered in the solution process. Consequently, in the presence of noise, the 3-by-3 matrix is not exactly orthonormal, and the accuracy of the final result is relatively poor even when it is further improved by finding the closest orthonormal matrix. For our experiments we have included an orthonormalization step described in Appendix A in PTM and RAC.

2.3 Two-Step Methods

A two-step method solves the problem in two stages. In the first stage, a linear algorithm is employed to get an approximate closed-form solution. In the second stage, a nonlinear method uses the previous closed-form solution as an initial guess to search for a better result.

The basic idea behind two-step methods is that either method serves as a complement to its counterpart. Without an initial guess, linear methods always give a solution, but it may not be optimal in the presence of noise. Conversely, classical methods can reach the optimal solution if a reasonable initial guess is available. The same idea can be found in structure-from-motion [30] and camera calibration for both camera extrinsic and intrinsic parameters [27, 31].

3 The Depth Reconstruction Approach

Most existing methods for solving exterior orientation, including classical methods and linear methods like PTM and RAC, are based on the collinearity equation (3). The basic difference in our approach is that we rewrite this expression as

$$d_i \mathbf{y}_i = R\mathbf{X}_i + \mathbf{T}, \tag{18}$$

where $d_i = \mathbf{r}_3^t \mathbf{X}_i + t_3$. Treating d_i as an undetermined factor simplifies structure of the problem from a nonlinear least-squares to a constrained linear optimization of

$$\sum_i w_i \| s d_i \mathbf{y}_i - R\mathbf{X}_i - \mathbf{T} \|^2. \tag{19}$$

This comes at the expense of having to solve for many more unknowns, $(\{d_i\})$. The value $\mathbf{Y}_i = d_i \mathbf{y}_i$ will be called the *hypothesized scene point* for the reference point $\mathbf{X}_i$. Physically, it is the 3-D coordinate of the reference point in camera coordinate frame. Because the objective function in this form measures the 3-D object space error resulting from backprojection of image points into object coordinates, we refer to the new method as the "backprojection" algorithm (BPROJ). Minimizing (19) over all unknown values can be visualized as choosing a coordinate transformation so as to fit the 3-D model points in the camera coordinate frame to the bundle of the lines of sight associated with the image vectors.

The key observation of our method is to note that, for fixed values of d_i, it is possible to solve for R, $\mathbf{T}$ and s in closed form. Conversely, for fixed R, $\mathbf{T}$ and s, it is possible to solve for the values of d_i in closed form. Intuitively, s can be thought of as the principle depth of a rigidly transformed set of points. The values d_i describe the relative depths among the points. The advantage of using two depth parameters is that the first stage of optimization can compute the principle depth in addition to rotation and translation. The second stage of optimization adjusts the relative depths of the individual points and scales them to the principle depths.

Formally, the optimal R, $\mathbf{T}$ and s for fixed depths $\{\bar{d}_i\}$ are computed by minimizing (19), which is equivalent to solving an absolute orientation problem with

$$\bar{d}_i \mathbf{y}_i = s' R \mathbf{X}_i + \mathbf{T}' \quad i = 1, \ldots, N, \tag{20}$$

where $s' = 1/s$ and $\mathbf{T}' = \mathbf{T}/s$. This is equivalent to (8) which can be solved exactly in closed-form. It is interesting to note that the scaling computed by this optimization is equal to the ratio of the root-mean-square deviations of the coordinates in the two systems from their respective centroids [22]. In our notation, this gives

$$s = \sqrt{\frac{\sum_i \|\mathbf{X}_i - \hat{\mathbf{X}}\|^2}{\sum_i \|\mathbf{Y}_i - \hat{\mathbf{Y}}\|^2}}, \tag{21}$$

where

$$\mathbf{Y}_i = \bar{d}_i \mathbf{y}_i, \quad \hat{\mathbf{X}} = \frac{\sum_i w_i \mathbf{X}_i}{\sum_i w_i}, \quad \hat{\mathbf{Y}} = \frac{\sum_i w_i \mathbf{Y}_i}{\sum_i w_i}. \tag{22}$$

Given fixed rotation $\bar{R}$, translation $\bar{\mathbf{T}}$, and scale $\bar{s}$, the optimal depths $\{d_i\}$ are computed by

$$d_i = \frac{(\bar{R}\mathbf{X}_i + \bar{\mathbf{T}})^t \mathbf{y}_i}{\bar{s}\mathbf{y}_i^t \mathbf{y}_i}. \tag{23}$$

We can imagine the operation of the algorithm by viewing the hypothesized scene points as sliding beads moving along their respective lines of sight. The algorithm works by first moving the reference points rigidly toward the bundle of the lines of sight while the positions of the sliding beads are adjusted in concert along their lines of sight by the parameter s. In the second stage, the reference points are held fixed, while the beads are individually moved toward their respective reference points.

Since both problems are solvable in closed form, the overall solution can be cheaply obtained by coordinate-wise optimization over either group of parameters iteratively with $\{d_i\}$ starting out at some initial depths. Experiments show that initializing all d_i to the same positive constant (notice that the exact value is not important because of the use of s) works very well.

An earlier depth reconstruction algorithm [6] uses similar objective function, but without the scale factor s. It suffers from tremendously slow convergence [16]. It is proved to be globally convergent with respect to initial values of $\{d_i\}$. The essence of the proof is that for arbitrary temporary depths, we can always find the optimal R and $\mathbf{T}$, and since the objective function is quadratic in d_i, the solution for d_i is guaranteed to further decrease the objective function. The proof is also applicable to our algorithm since s, together with R and $\mathbf{T}$ can be solved exactly given $\{d_i\}$ (see Appendix B for details). It turns out that the role played by s is critical to the performance of the algorithm. Both the convergence rate and the accuracy of the result are dramatically improved by solving for s in addition to the other unknowns.

4 Experiments

We have performed extensive experiments on synthetic data with varying numbers of reference points, noise, and percentages of outliers. We have also performed an experimental analysis of the method in the context of robot calibration. The results of these tests are summarized below.

4.1 Simulation

A set of 3-D points for $\{\mathbf{X}_i\}$ are generated uniformly within a box defined by x_i, y_i, z_i $\in$ $[-5, 5]$. In order to generate a 3-D rotation R, a unit quaternion is uniformly selected from a unit 4-sphere. The resulting distribution of 3-D rotations is also uniform [5]. For translation $\mathbf{T}$, t_1 and t_2 are uniformly selected from $[5, 15]$, and t_3 from $[20, 50]$. The set of 3-D coordinates in the camera coordinate frame $\mathbf{Y}_i = R\mathbf{X}_i + \mathbf{T}$ are generated according to the following control parameters: number of points N, signal-to-noise ratio (SNR), and percentage of outliers (PO).

Gaussian noise $\mathcal{N}(0, \sigma)$ is added to both coordinates of the perspective projection of each $\mathbf{Y}_i$, where the variance, σ, is related to SNR by SNR $= -20\log(\sigma/0.3)$ dB (the image size is roughly $10/35 \approx 0.3$). A fraction (= PO %) of the 3-D points are selected as outliers. Each of these points $\mathbf{Y}_i = (x'_i, y'_i, z'_i)^t$ is replaced by another 3-D point $(x^*_i, y^*_i, z^*_i)^t, z^*_i = z'_i$, where x^*_i and y^*_i are uniformly distributed within $[t_1 - 5, t_1 + 5]$ and $[t_2 - 5, t_2 + 5]$, respectively. The preprocessed 3-D points are then projected onto the normalized image plane ($z = 1$).

The following three experiments were conducted on the generated data sets:

C1 Set $N = 20$, PO $= 0$. Estimate the errors of rotation and translation against SNR (30 dB-70 dB in 10 dB step). The purpose is to measure the noise-resisting capability of the tested method.

C2 Set $N = 20$, SNR $= 60$ dB. Estimate the errors of rotation and translation against PO (5 %-25 % in 5 % step). The purpose is to see how well the tested method tolerates outliers.

C3 Set PO $= 0$, SNR $= 30$. Estimate the errors of rotation and translation against N (10 to 50 by step of 10). The purpose is to investigate how the performance can be improved by increasing the number of reference points.

To assess the performance of the methods, we measure the mean errors in rotation and translation of 1000 trials for each setting of the control parameters.

4.1.1 An Error measure for 3-D rotations

The error measure for translation is straightforward since a 3-vector has natural Euclidean norm. The error measure for rotation depends on its representation. When represented by Euler angles, there is no natural norm for 3-D rotation. However, when represented by a unit quaternion, the rotation error can be represented by quaternion error. The difference between any two unit quaternions $\boldsymbol{q}, \boldsymbol{q}'$ is

$$\boldsymbol{q} - \boldsymbol{q}' = 2(1 - \boldsymbol{q}^t\boldsymbol{q}') \tag{24}$$

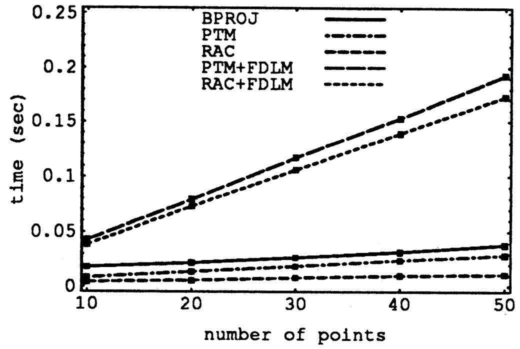

Figure 1: Average running times for the tested methods against the number of the reference points.

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

$$1 - |\boldsymbol{q}^t\boldsymbol{q}'| \in [0, 1]. \tag{25}$$

An important advantage of this error measure is that it is independent of coordinate system.

4.1.2 Results and discussions

The basic methods tested are BPROJ, PTM, and RAC. The classical method tested here uses the finite difference Levenberg-Marquardt minimization (FDLM) [3]. It is not applied directly. Instead it is employed as the second stage of a two-step method using the solution provided by one of the linear methods as an initial guess. All the experiments were conducted on a Silicon Graphics IRIS Indigo with a MIPS R4400 processor. The results are plotted in logarithmic scales on both X and Y axes (SNR is logarithm of Gaussian variance). The reason that we compress the scale on large errors is that when errors exceed some threshold, the method can be thought of to have *failed*, so it is not very meaningful to try to discriminate between large errors.

Figure 1 shows the average running times of the methods we tested against the number of reference points. These times include the times for generating random data sets.

Figure 2, Figure 3 and Figure 4 compare BPROJ, PTM and RAC. BPROJ is superior to PTM and RAC when presented with noise and outliers. RAC is better than PTM in general. This is probably because in RAC only the first two rows of the 3-by-3 transformation are computed while ignoring orthonormality, whereas in PTM all nine parameters are computed without the orthonormality constraint. Note that the running time of BPROJ is about the same magnitude as that of PTM and RAC (see Figure 1). Experiments show that BPROJ converges in about 5-10 iterations, where each iteration first solves an absolution orientation problem, and then calculates the depths.

Given close initial guesses, FDLM can sometimes reach solutions better than that provided by BPROJ as shown in Figure 5 and Figure 7. However, such differences happen either at high SNR (≥ 60) or with large number of reference points (≥ 30). The magnitudes are so small that they may not be noticeable in real applications. Furthermore, FDLM is much slower than BPROJ.

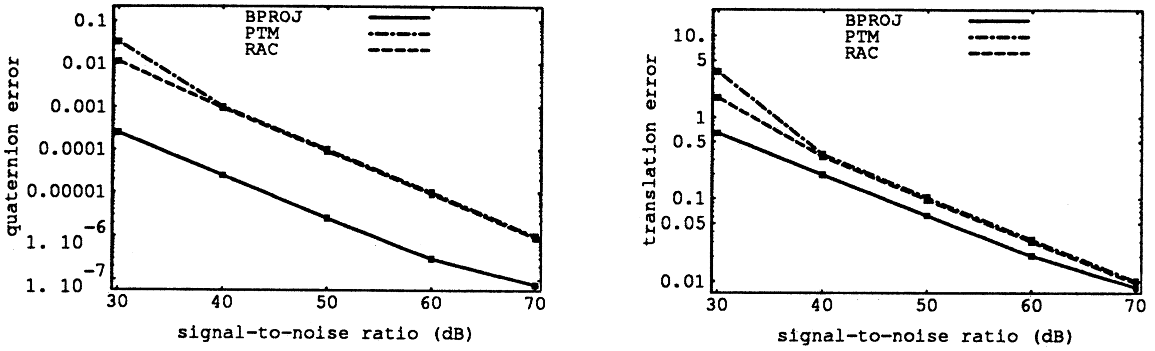

Figure 2: Result of Experiment **C1** for linear methods PTM, RAC and the backprojection algorithm (BPROJ).

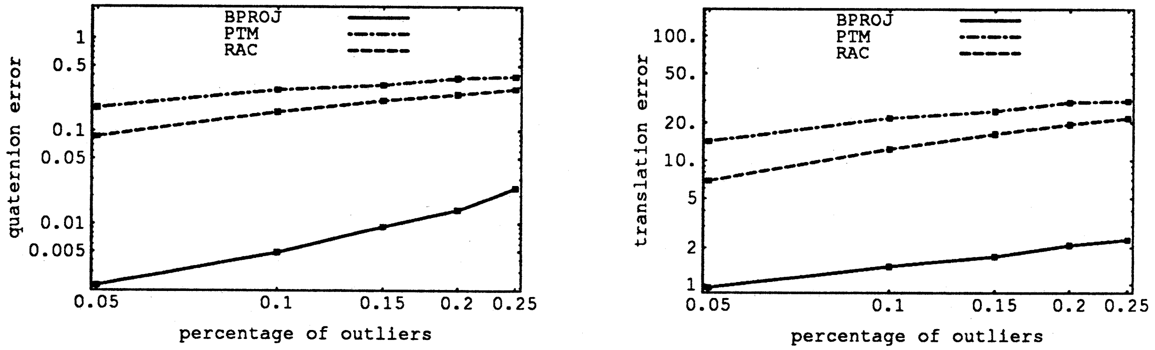

Figure 3: Result of Experiment **C2** for linear methods PTM, RAC and the backprojection algorithm (BPROJ).

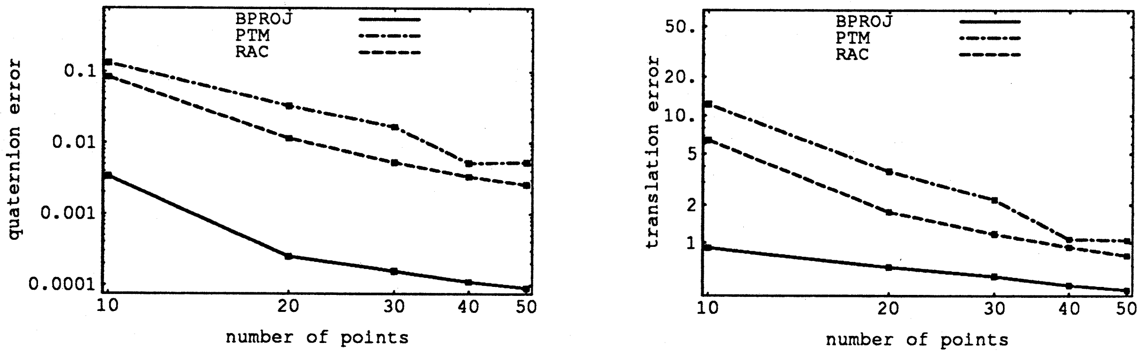

Figure 4: Result of Experiment **C3** for linear methods PTM, RAC and the backprojection algorithm (BPROJ).

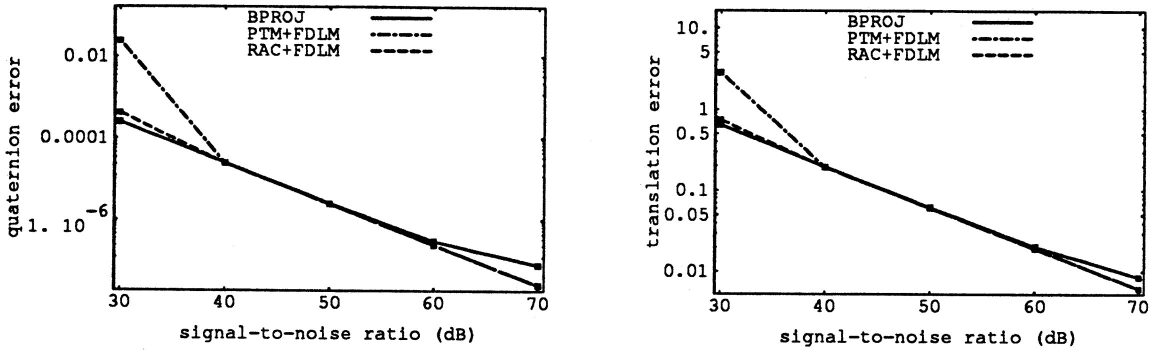

Figure 5: Result of Experiment **C1** for two-step methods and the backprojection algorithm (BPROJ).

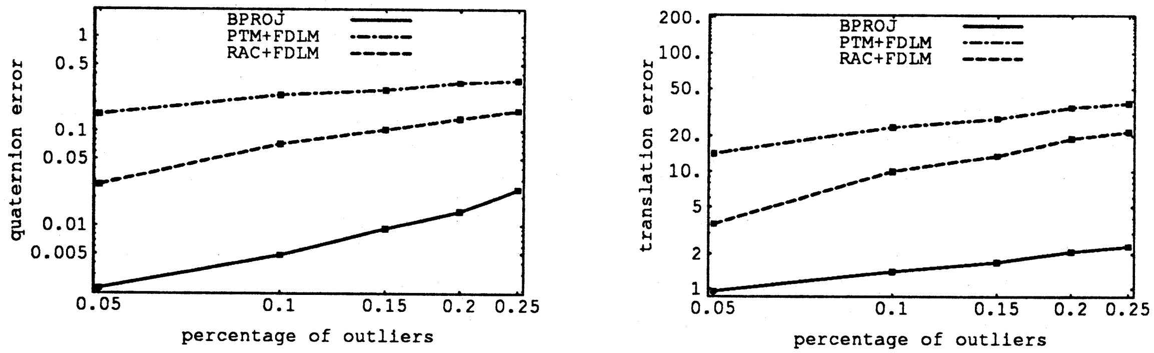

Figure 6: Result of Experiment **C2** for two-step methods and the backprojection algorithm (BPROJ).

Figure 6 shows that the capability for tolerating outliers of the two-step methods is no better than that of linear methods. When the first-stage linear method fails due to the presence of outliers, the second-stage classical method also fails because it starts with a radically wrong initial guess. BPROJ is more tolerant of outliers than the two-step methods.

4.2 Hand-Eye Calibration

Given the 3-D coordinates of the reference points and their corresponding camera projections, we compute the rotation and translation that relate the coordinate frame of a robot arm and that of a camera using BPROJ, PTM, RAC, PTM+FDLM, and RAC+FDLM.

4.2.1 Experimental setting

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

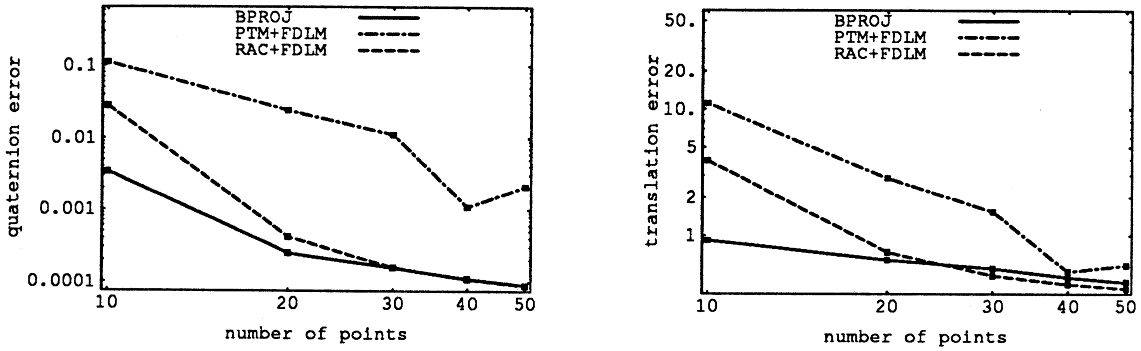

Figure 7: Result of Experiment **C3** for two-step methods and the backprojection algorithm (BPROJ).

The various methods were applied to datasets acquired by visually tracking a fixed reference point on an object attached to the robot arm.[1] Data was acquired by moving the arm to 35 positions, and at each position compiling a data pair consisting of the absolute coordinates of a feature in the robot frame (computed from the robot inverse kinematics), and the image coordinates of the feature provided by tracking. Figure 9 shows the features used to generate the data. This process was repeated 5 times to obtain 5 data sets for each camera. The physical conditions are shown in Figure 8. The Sony XC-77 (middle, bottom) was positioned nearly aligned with the robot coordinate system and was tuned to have sharp images. The Cohu (left, top) was positioned more to the side, and delivered more defocused images.

4.2.2 Results and discussion

The results of the calibration methods are compared by computing the sum of the squared image error for the data set. The image error is determined by comparing the observed 2-D projections to those obtained by projecting the reference points to the image plane using the computed calibration parameters. The results for the five trials for both cameras are plotted in Figure 10. It turned out that the results given by the two-step methods are very close to those given by BPROJ, so only the BPROJ results are plotted. Given that the two-step methods required several times as long to converge (see Figure 1), BPROJ would clearly be preferred in these circumstances.

Among the remaining methods, it is clear that BPROJ is more stable and accurate than PTM and RAC while RAC is better than PTM. The effective SNR for these data set is approximately in the range of 40dB to 60dB, so these results agree well with the simulation results. One difference between the simulations and these tests is that the errors in the reference points are significant (on the order of up to a centimeter). Despite these errors, BPROJ appears to compute an accurate transformation. Overall, the Cohu calibration error is generally lower than the Sony error. We hypothesize that this is due to the fact that the field of view covered by the data in the Cohu camera is less than half of that covered in the Sony camera. Consequently, the effect of kinematic errors is smaller by nearly a factor of two which accounts for the difference.

[1] The tracking system is more fully described in [15].

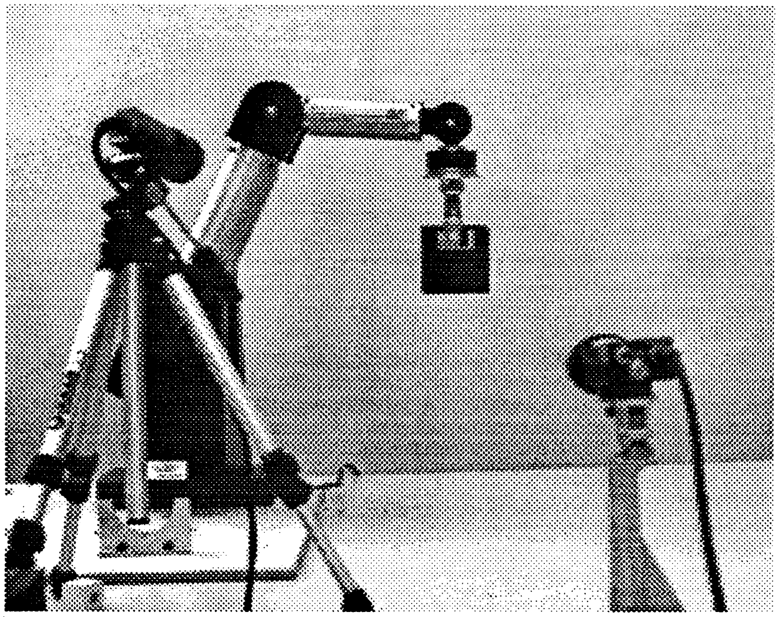

Figure 8: The experimental setup showing the positions of the two cameras relative to the robot arm.

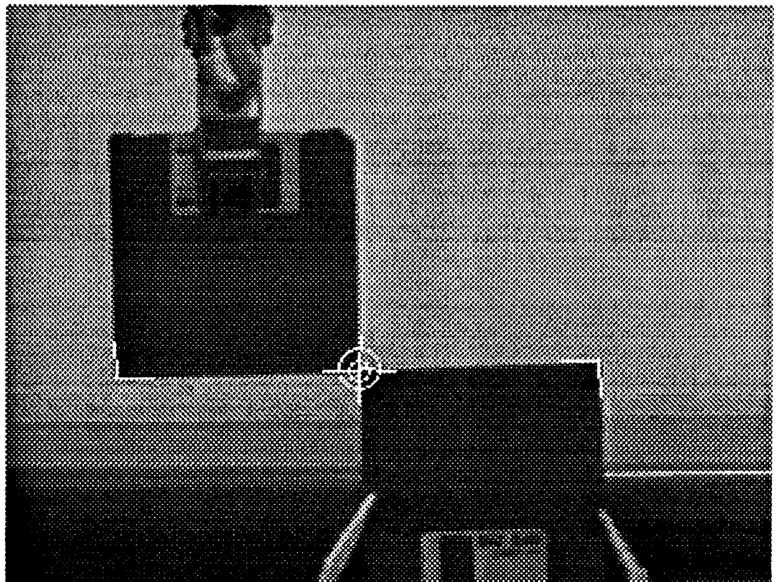

Figure 9: An image from the cameras showing the tracking used to generate image feature point data.

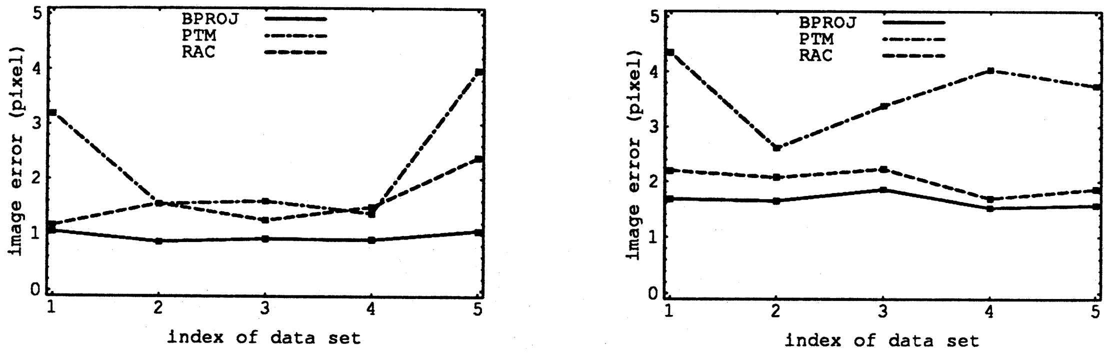

Figure 10: Results of the hand-eye calibration experiments for (*left*) Cohu and (*right*) Sony XC-77 cameras. Mean image errors in pixels are measured for each data set.

5 Discussion

Two important issues regarding online hand-eye calibration still remain unaddressed. First, although BPROJ is relatively insensitive to outliers, outliers still cause large errors just like any other least-squares method. The second problem is that there are cases where the camera intrinsic parameters change dynamically (for example, with auto-focus lenses) and therefore also need to be estimated online.

An obvious solution to the first issue is to use BPROJ as the kernal of a truly robust exterior orientation algorithm. BPROJ is a good candidate for a robust algorithm since the kernel algorithm within any robust method should itself be relatively robust. We are currently taking an aggressive approach to robustness based on on an enhancement to BPROJ that solves the exterior orientation problem when the correspondences between the 3-D reference points and their image vectors are totally unknown. We believe that this algorithm will perform well and be cheap to compute when a nearly correct set of correspondences is supplied as an initial guess.

As for the second issue, BPROJ in its current form is not well-suited for intrinsic calibration. Conversely, Tsai's method provides a nice way to estimate the intrinsic parameters efficiently and accurately given well-behaved data. Since RAC (the first stage of Tsai's method) is so sensitive to outliers, it is worthwhile to do some extra computation to remove outliers. Hence, a proposal for online intrinsic calibration is to use robust BPROJ for outlier rejection and initial exterior orientation estimation, and to use Tsai's method for estimating the camera intrinsic parameters using the "outlier-free" data. Iterating between the two methods should provide a relatively fast and effective online calibration method for both intrinsic and extrinsic camera parameters.

We believe that the coordinate optimization approach used in this work may also be generalized to solve the structure-from-motion problem [30] or relative orientation [23]. In this case, (19) can be reformulated as

$$(26) \qquad \sum_i w_i \| s d'_i \mathbf{y}_i - d_i R \mathbf{x}_i - T \|^2,$$

where $\mathbf{x}_i$ and $\mathbf{y}_i$ are two views of the same reference point $\mathbf{X}_i$ (unknown this time) using two different cameras or the same camera in tow different viewing positions. The values d_i and d'_i are the undetermined depths of $\mathbf{X}_i$ in the two camera coordinate frames, respectively. Structure from motion is solved by minimized (26) with respect to the "motion" R, $\mathbf{T}$, and the "structure" $\{d_i\}$, $\{d_i'\}$. Again, a coordinate-wise optimization may be applied. First, R and $\mathbf{T}$ are computed from two sets of hypothesized depths for each camera. The depths d_i and d_i' are reconstructed for given R and $\mathbf{T}$ by intersecting the lines of sight associated with $\mathbf{x}_i$ and $\mathbf{y}_i$, respectively, and the process is iterated.

6 Conclusions

We have presented a new algorithm for solving exterior orientation and compared it to several other commonly used methods. Both experiments with synthetic and real data have shown that it is less sensitive to noise and outliers than linear methods while at same time much faster than classical methods or two-step methods. Since it is globally convergent, we do not have to worry about initializations. Future work will be devoted to theoretical investigations on the convergence rate of the method, and on extensions to the method for solving for correspondences, and to extensions for structure-from-motion.

7 Acknowledgments

We would like to thank Anand Rangarajan and Robert Haralick for helpful discussions. This work is supported by DARPA/ONR under grant N00014-92-J-4048 and the Yale Center for Theoretical and Applied Neuroscience (CTAN).

References

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

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

[3] K. M. Brown and J. E. Dennis, *Derivative free analogues of the Levenberg-Marquardt and Guass algorithms for nonlinear least squares approximation*, Numeriche Mathematik **18** (1972), 289–297.

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

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

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

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

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

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

[10] W. E. L. Grimson, *Object Recognition by Computer*, The MIT Press, Cambridge, Massachusetts, 1990.

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

[12] ———, *Six dof visual control of relative position*, DCS RR-1038, Yale University, New Haven, CT, June 1994.

[13] G. D. Hager, W.-C. Chang, and A. S. Morse, *Robot feedback control based on stereo vision: Towards calibration-free hand-eye coordination*, Proc. IEEE Conf. Rob. Automat., IEEE Computer Society Press, May 1994, pp. 2850–2856.

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

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

[16] R. M. Haralick, March 1994, Private communication.

[17] R. M. Haralick and L. G. Shapiro, *Computer and Robot Vision*, Addison-Wesley Publishing Company, Reading, Massachusetts, 1993.

[18] ———, Computer and Robot Vision, ch. 14, p. 132, Addison-Wesley Publishing Company, Reading, Massachusetts, 1993, p. 132.

[19] ———, Computer and Robot Vision, ch. 14, pp. 131–143, Addison-Wesley Publishing Company, Reading, Massachusetts, 1993, pp. 131–143.

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

[21] B. K. P. Horn, *Robot Vision*, The MIT Press, Cambridge, Massachusetts, 1986.

[22] ———, *Closed-form solution of absolute orientation using unit quaternion*, J. Opt. Soc. Amer. **A-4** (1987), 629–642.

[23] ———, *Relative orientation*, Intl. J. Computer Vision **4** (1990), 59–78.

[24] B. K. P. Horn, H. M. Hilden, and S. Negahdaripour, *Closed-form solution of absolute orientation using orthonomal matrices*, J. Opt. Soc. Amer. **A-5** (1988), 1127–1135.

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

[26] D. G. Lowe, *Three-dimensional object recognition from single two-dimensional image*, Artificial Intelligence **31** (1987), 355–395.

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

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

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

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

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

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

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

A Fitting Orthonormal Matrices

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

$$\|M - sR\|_F, \tag{27}$$

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

$$\sum_{i=1}^{3} \|\mathbf{m}^{(i)} - sR\mathbf{e}^{(i)}\|^2, \tag{28}$$

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

B Global Convergence of BPROJ

Let $d_i^{(k)}, R^{(k)}, \mathbf{T}^{(k)}$, and $s^{(k)}$ be the estimates of $d_i, R, \mathbf{T}$, and s at the kth iteration of the optimization. Within each iteration, $R^{(k)}, \mathbf{T}^{(k)}$ and $s^{(k)}$ are computed given the kth estimates of the depths $d_i^{(k)}$. The value of the objective function at the kth iteration is denoted as $E^{(k)}$. We prove the convergence of BPROJ by showing that $E^{(k+1)} \leq E^{(k)}$ for any $d_i^{(k)}$ as follows:

Since the absolute orientation problem can be solved in exact closed form, we have

$$E^{(k+1)} = \sum_i w_i \| s^{(k+1)} d_i^{(k+1)} \mathbf{y}_i - R^{(k+1)} \mathbf{X}_i - \mathbf{T}^{(k+1)} \|^2 \tag{29}$$

$$\leq \sum_i w_i \| s^{(k)} d_i^{(k+1)} \mathbf{y}_i - R^{(k)} \mathbf{X}_i - \mathbf{T}^{(k)} \|^2. \tag{30}$$

The objective function (30) is quadratic in $d_i^{(k+1)}$ which can be solved optimally as (23). Therefore, for any $\{d_i^k\}$

$$E^{(k+1)} \leq \sum_i w_i \| s^{(k)} d_i^{(k+1)} \mathbf{y}_i - R^{(k)} \mathbf{X}_i - \mathbf{T}^{(k)} \|^2 \tag{31}$$

$$\leq \sum_i w_i \| s^{(k)} d_i^{(k)} \mathbf{y}_i - R^{(k)} \mathbf{X}_i - \mathbf{T}^{(k)} \|^2 \tag{32}$$

$$= E^{(k)}. \tag{33}$$