Fast and Globally Convergent Pose Estimation From Video Images

Chien-Ping Lu[*] Gregory D. Hager[†] Eric Mjolsness[‡§]

February 18, 1998

Abstract

Determining the rigid transformation relating 2D images to known 3D geometry is a classical problem in photogrammetry and computer vision. Heretofore, the best methods for solving the problem have relied on iterative optimization methods which cannot be proven to converge and/or which do not effectively account for the orthonormal structure of rotation matrices.

We show that the pose estimation problem can be formulated as that of minimizing an error metric based on collinearity in object (as opposed to image) space. Using object space collinearity error, we derive an iterative algorithm which directly computes orthogonal rotation matrices and which is globally convergent. Experimentally, we show that the method is computationally efficient, that it is no less accurate than the best currently employed optimization methods, and that it outperforms all tested methods in robustness to outliers.

[*]Chien-Ping Lu, Silicon Graphics Inc. cplu@engr.sgi.com

[†]Greg Hager, Department of Computer Science, Yale University. hager@cs.yale.edu.

[‡]Eric Mjolsness, Jet Propulsion Laboratory, NASA. eric.d.mjolsness@jpl.nasa.gov

[§]The authors would like to thank Prof. R. Haralick for his useful comments and ideas on this research.

1 Introduction

Determining the rigid transformation that relates images to known geometry, the *pose estimation problem*, is one of the central problems in photogrammetry, robotics, computer graphics, and computer vision. In robotics, pose estimation is commonly used in hand-eye coordination systems [1]. In computer graphics, it plays a central role in tasks that combine computer-generated objects with photographic scenes — e.g. landmark tracking for determining head pose in augmented reality [2, 3, 4, 5] or interactive manipulation of objects. In computer vision, pose estimation is central to many approaches to object recognition [6].

The information available for solving the pose estimation problem is usually given in the form of a set of point correspondences, each composed of a 3D reference point expressed in object coordinates and its 2D projection expressed in image coordinates. For three or four points, exact solutions can be computed: a fourth- or fifth-degree polynomial system can be formulated using geometrical invariants of the observed points, and the problem can be solved by finding roots of the polynomial system [7, 8, 9, 10, 11, 12]. However, the resulting methods can only be applied to a limited number of points and are thus sensitive to additive noise and possible outliers.

For more than four points, closed form solutions do not exist. The classical approach used in photogrammetry is to formulate pose estimation as a nonlinear least squares problem, and to solve it by nonlinear optimization algorithms, most typically, the Gauss-Newton method [13, 14, 15]. In the vision literature, the work by Lowe and its variants [16, 17] is an example of applying the Gauss-Newton method to the pose estimation problem. As with most nonlinear optimizations, these methods rely on a good initial guess to converge to the correct solution. There is no guarantee that the algorithm will eventually converge or that it will converge to the correct solution.

A class of approximate methods for pose estimation have been developed by relaxing the orthogonality constraint on rotation matrices and/or by simplifying the perspective camera model [18, 19, 20, 21, 22, 23, 24, 25]. In iterative reduced perspective methods [23, 26], an approximate solution computed using a simplified camera model is iteratively refined

to approach a full perspective solution. In these methods the rotation matrix is computed in two steps: first a linear (unconstrained) solution is computed, and then this solution is fit to the "closest" orthogonal matrix. It has been shown that this two-step approach for computing rotation is not the same as finding the best orthogonal matrix [27]. Again, with such methods there is no guarantee that they will eventually converge to the correct solution when applied iteratively.

The developments in this article were originally motivated by the work of Haralick *et el.*[28]. He introduced a pose estimation algorithm which simultaneously computes both object pose and the depths of the observed points. The algorithm seems to be globally convergent although a complete proof was not given. What makes this algorithm attractive is that the non-linearity due to perspective is eliminated by the introduction of the depth variables. However, this algorithm has not received much attention, probably due its slow local convergence rate (hundreds of iterations) as indicated in [28] and found by ourselves.

In our approach, we reformulate the pose estimation problem as that of minimizing an *object-space* collinearity error. From this new objective function, we derive an algorithm which operates by successively improving an estimate of the rotation portion of the pose, and then estimates an associated translation. The intermediate rotation estimates are always the best "orthogonal" solution for each iteration. The orthogonality constraint is enforced by using singular value decomposition, not from a specific parameterization of rotations, e.g. Euler angles. We further prove that the proposed algorithm is globally convergent. Empirical results suggest that the algorithm is also extremely efficient and usually converges in 5 to 10 iterations from very general geometrical configurations. In addition, the same experiments suggest that our method outperforms the Levenberg-Marquardt methods, one of the most reliable optimization methods currently in use, in terms of both accuracy against noise and robustness against outliers.

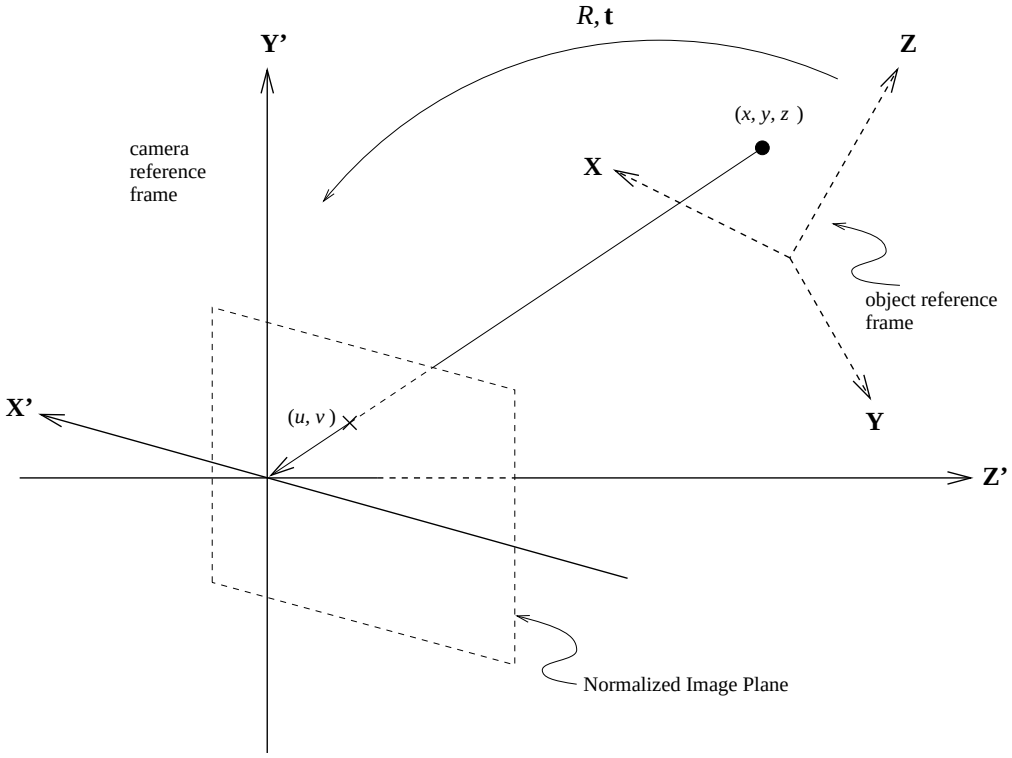

Figure 1: The reference frames in the pose estimation problem.

1.1 Outline of the article

The remainder of this article is organized as follows. The next section describes the formulation of the pose estimation problem more formally and briefly reviews some of the classical iterative methods used to solve it. Section 3 introduces the orthogonal iteration algorithm and proves its global convergence. The link between weak perspective and the proposed method is also presented. In Section 4, detailed performance analyses using large scale simulations are given to compare our method to existing methods. Finally, Section 5 concludes by suggesting some directions in which the method could be extended. An appendix contains technical arguments for two results needed for discussions within the article.

2 Problem Formulation

2.1 Camera model

The mapping from 3D reference points to 2D image coordinates can be formalized as follows. Given a set of 3D coordinates of reference points $\boldsymbol{p}_i = (x_i, y_i, z_i)^t, i = 1, \ldots, n, n \geq 3$ expressed in an object-centered reference frame, the corresponding camera-space coordinates $\boldsymbol{q}_i = (x_i', y_i', z_i')^t$, are related by a rigid transformation as

$$\boldsymbol{q}_i = R\boldsymbol{p}_i + \boldsymbol{t}, \tag{1}$$

where

$$R = \begin{pmatrix} \boldsymbol{r}_1^t \\ \boldsymbol{r}_2^t \\ \boldsymbol{r}_3^t \end{pmatrix} \in \mathbf{SO(3)} \qquad \text{and} \qquad \boldsymbol{t} = \begin{pmatrix} t_x \\ t_y \\ t_z \end{pmatrix} \in \mathbf{R}^3 \tag{2}$$

are a rotation matrix and a translation vector, respectively.

The camera reference frame is chosen so that the center of projection of the camera is at the origin, and the optical axis points in the positive z direction. The reference points $\boldsymbol{p}_i$ are projected to the plane with $z' = 1$, referred to as the *normalized image plane*, in the camera reference frame[1]. Let the image point $\boldsymbol{v}_i = (u_i, v_i, 1)^t$ be the projection of $\boldsymbol{p}_i$ on the normalized image plane. Under the idealized pinhole imaging model, $\boldsymbol{v}_i$, $\boldsymbol{q}_i$ and the center of projection are collinear. This fact is expressed by the following equation:

$$u_i = \frac{\boldsymbol{r}_1^t \boldsymbol{p}_i + t_x}{\boldsymbol{r}_3^t \boldsymbol{p}_i + t_z} \tag{3a}$$

$$v_i = \frac{\boldsymbol{r}_2^t \boldsymbol{p}_i + t_y}{\boldsymbol{r}_3^t \boldsymbol{p}_i + t_z} \tag{3b}$$

or

$$\boldsymbol{v}_i = \frac{1}{\boldsymbol{r}_3^t \boldsymbol{p}_i + t_z}(R\boldsymbol{p}_i + \boldsymbol{t}), \tag{4}$$

[1]We assume throughout this article that the camera internal calibration (including both lens distortion and the mapping from metric to pixel coordinates) is known.

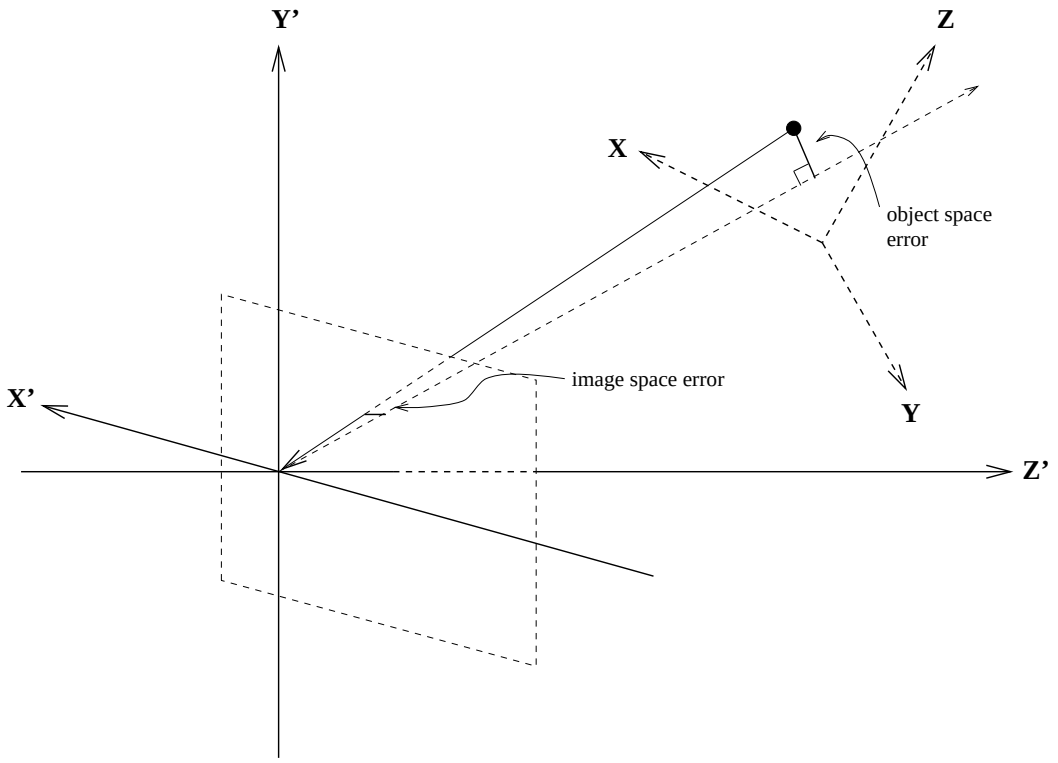

Figure 2: Object-space and image-space collinearity errors.

which is known as the *collinearity equation* in the photogrammetry literature. However, another way of thinking of collinearity is that the orthogonal projection of $\boldsymbol{q}_i$ on $\boldsymbol{v}_i$ should be equal to $\boldsymbol{q}_i$ itself. This fact is expressed by the following equation:

$$R\boldsymbol{p}_i + \boldsymbol{t} = F_i(R\boldsymbol{p}_i + \boldsymbol{t}) \tag{5}$$

where

$$F_i = \frac{\boldsymbol{v}_i \boldsymbol{v}_i^t}{\boldsymbol{v}_i^t \boldsymbol{v}_i} \tag{6}$$

Note that $\|F_i \boldsymbol{x}\| \leq \|\boldsymbol{x}\|, \boldsymbol{x} \in \mathbf{R}^3$ and, since F_i is a projection operator, $F_i^t = F_i$ and $F_i^2 = F_i$.

In the remainder of this article, we refer to (4) as the *image space* collinearity equation, and (5) as the *object space* collinearity equation. The pose estimation problem is to develop an algorithm for finding the rigid transform $(R, \boldsymbol{t})$ that minimizes some form of accumulation of the errors (for example, summation of squared errors) of either of the collinearity equations (see Figure 2).

2.2 Classical iterative methods

As noted in the introduction, the most widely used and most accurate approaches to the pose estimation problem use iterative optimization methods. In classical photogrammetry, the pose estimation problem is usually formulated as the problem of optimizing the following objective function:

$$\sum_{i=1}^{n}\left[\left(u_i - \frac{\boldsymbol{r}_1^t \boldsymbol{p}_i + t_x}{\boldsymbol{r}_3^t \boldsymbol{p}_i + t_z}\right)^2 + \left(v_i - \frac{\boldsymbol{r}_2^t \boldsymbol{p}_i + t_y}{\boldsymbol{r}_3^t \boldsymbol{p}_i + t_z}\right)^2\right], \tag{7}$$

where the rotation matrix, R, is usually parameterized using Euler angles. Note that this is a minimization over image-space collinearity.

Two commonly used minimization algorithms are the Gauss-Newton method and Levenberg-Marquardt method. The Gauss-Newton method is a classical technique for solving nonlinear least-squares problems. It operates by iteratively linearizing the collinearity equation around the current approximate solution by first-order Taylor series expansion, and then solving the linearized system for the next approximate solution. The Gauss-Newton method relies on a good local linearization. If the initial approximate solution is good enough, it should converge very quickly to the correct solution. However, when the current solution is far from the correct one and/or the linear system is ill-conditioned, it may converge slowly or even fail to converge altogether. It has been empirically observed [29] that for the Gauss-Newton method to work, the initial approximate solutions have to be within 10% of scale for translation and within 15^o for each of the three rotation angles.

The Levenberg-Marquardt method can be regarded as an interpolation of steepest descent and the Gauss-Newton method. When the current solution is far from the correct one, the algorithm behaves like a steepest descent method: slow but guaranteed to converge. When the current solution is close to the correct solution, it becomes a Gauss-Newton method. It has become a standard technique for nonlinear least squares problems, and has been widely adopted in computer vision [30, 31] and computer graphics [3].

2.3 Why another iterative algorithm?

Classical optimization techniques are currently the only choice when observed data is noisy and a high accuracy solution to the pose estimation problem is desired. However, since these algorithms are designed for solving general optimization problems, the specific structure of the pose estimation problem is not fully exploited. Furthermore, the commonly used Euler angle parameterization of rotation obscures the algebraic structure of the problem. The analysis for both global and local convergence is only valid when the intermediate result is close to the solution. At the same time, recent developments in vision-based robotics [32, 33] and augmented reality demand pose estimation algorithms to be not only accurate, but also robust to corrupted data and computationally efficient. Hence, this is a need for algorithms that are as accurate as classical optimization methods, yet are also globally convergent and fast enough for real-time applications.

3 The Orthogonal Iteration Algorithm

In this section we develop our new pose estimation algorithm, subsequently referred to as the *orthogonal iteration* (OI) algorithm. The method of attack is to first define pose estimation using an appropriate object space error function, and then to show that this function can be rewritten in a way which admits an iteration based on the solution to the 3D-3D pose estimation or *absolute orientation* problem. Since the algorithm depends heavily on the solution to absolute orientation, we first review the absolute orientation problem and its solution before presenting our algorithm and proving its convergence.

3.1 Optimal absolute orientation solution

The absolute orientation problem can be posed as follows: Suppose the 3D camera-space coordinates $\boldsymbol{q}_i$ could be reconstructed physically (for example, by range sensing) or computationally (for example, by stereo matching or structure-from-motion). Then for each observed

point, we have

$$\boldsymbol{q}_i = R\boldsymbol{p}_i + \boldsymbol{t}. \tag{8}$$

Computing absolute orientation is the process of determining R and $\boldsymbol{t}$ from corresponding pairs $\boldsymbol{q}_i$ and $\boldsymbol{p}_i$. With three or more non-collinear reference points, R and $\boldsymbol{t}$ can be obtained as a solution to the following least squares problem

$$\min_{R,\boldsymbol{t}} \sum_{i=1}^{n} \|R\boldsymbol{p}_i + \boldsymbol{t} - \boldsymbol{q}_i\|^2, \quad \text{subject to } R^tR = I. \tag{9}$$

Such a constrained least squares problem [34] can be solved in closed form using quaternions [35, 36], or singular value decomposition (SVD) [27, 37, 35, 36].

The SVD solution proceeds as follows. Let $\{\boldsymbol{p_i}\}$ and $\{\boldsymbol{q_i}\}$ denote lists of corresponding vectors related by (1) and define

$$\bar{\boldsymbol{p}} \stackrel{\text{def}}{=} \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{p}_i, \quad \bar{\boldsymbol{q}} \stackrel{\text{def}}{=} \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{q}_i, \tag{10}$$

that is, $\bar{\boldsymbol{p}}$ and $\bar{\boldsymbol{q}}$ are the centroids of $\{\boldsymbol{p}_i\}$ and $\{\boldsymbol{q_i}\}$, respectively. Define

$$\boldsymbol{p}'_i = \boldsymbol{p}_i - \bar{\boldsymbol{p}}, \quad \boldsymbol{q}'_i = \boldsymbol{q}_i - \bar{\boldsymbol{q}}, \tag{11}$$

and

$$M = \sum_{i=1}^{n} \boldsymbol{q}'_i \boldsymbol{p}'^t_i. \tag{12}$$

In other words, $\frac{1}{n}M$ is the sample cross-covariance matrix between $\{\boldsymbol{p}_i\}$ and $\{\boldsymbol{q}_i\}$. It can be shown that, if R^*, $\boldsymbol{t}^*$ minimize (9), then they satisfy

$$R^* = \arg\max_R \operatorname{tr}(R^tM) \tag{13}$$

$$\boldsymbol{t}^* = \bar{\boldsymbol{q}} - R^*\bar{\boldsymbol{p}}. \tag{14}$$

Let (U, Σ, V) be a SVD of M, that is $U^tMV = \Sigma$. Then the solution to (9) is

$$R^* = VU^t \tag{15}$$

Note that the optimal translation is entirely determined by the optimal rotation, and all information for finding the best rotation is contained in M as defined in (12). Hence, only the position of the 3D points relative to their centroids is relevant in the determination of the optimal rotation matrix. It is also important to note that, although the SVD of a matrix is not unique, the optimal rotation is as shown in Appendix A.

3.2 The Orthogonal Iteration Algorithm

We now turn to the development of the Orthogonal Iteration Algorithm. The starting point for the algorithm is to state the pose estimation problem using the following *object-space* collinearity error vector:

$$\boldsymbol{e}_i = (I - F_i)(R\boldsymbol{p}_i + \boldsymbol{t}), \tag{16}$$

where F_i is as given in (6). We then seek to minimize

$$E(R, \boldsymbol{t}) = \sum_{i=1}^{n} \|\boldsymbol{e}_i\|^2 = \sum_{i=1}^{n} \|(I - F_i)(R\boldsymbol{p}_i + \boldsymbol{t})\|^2 \tag{17}$$

over R and $\boldsymbol{t}$. Since this objective function is quadratic in $\boldsymbol{t}$, given a fixed rotation R the optimal value for $\boldsymbol{t}$ can be computed in closed form as

$$\boldsymbol{t}(R) = (I - \frac{1}{n}\sum_j F_j)^{-1} \sum_j (F_j - I)R\boldsymbol{p}_j. \tag{18}$$

Note that $I - \frac{1}{n}\sum_{i=1}^{n} F_i$ is positive definite since for any 3-vector $\boldsymbol{x}$

$$\begin{aligned}
&\boldsymbol{x}^t(I - \frac{1}{n}\sum_{i=1}^{n} F_i)\boldsymbol{x} \\
=&\frac{1}{n}\sum_{i=1}^{n}(\|\boldsymbol{x}\|^2 - \boldsymbol{x}^t F_i \boldsymbol{x}) \\
=&\frac{1}{n}\sum_{i=1}^{n}(\|\boldsymbol{x}\|^2 - \boldsymbol{x}^t F_i^t F_i \boldsymbol{x}) \\
=&\frac{1}{n}\sum_{i=1}^{n}(\|\boldsymbol{x}\|^2 - \|F_i\boldsymbol{x}\|^2) \\
>&0
\end{aligned} \tag{19}$$

and hence (18) is well-defined.

Given the optimal translation as a function of R and defining

$$\boldsymbol{q}_i(R) \stackrel{\text{def}}{=} F_i(R\boldsymbol{p}_i + \boldsymbol{t}(R)) \quad \text{and} \quad \bar{\boldsymbol{q}}(R) \stackrel{\text{def}}{=} \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{q}_i(R), \tag{20}$$

(17) can be rewritten as

$$E(R) = \sum_{i=1}^{n} \|R\boldsymbol{p}_i + \boldsymbol{t}(R) - \boldsymbol{q}_i(R)\|^2. \tag{21}$$

This equation now bears a close resemblance to the absolute orientation problem (compare with (9)). Unfortunately, in this case we cannot solve for R in closed form as the sample cross-covariance matrix between $\{\boldsymbol{p}_i\}$ and $\{\boldsymbol{q}_i(R)\}$, that is

$$M(R) = \sum_{i=1}^{n} \boldsymbol{q}'_i(R)\boldsymbol{p}'^t_i \quad \text{where} \quad \boldsymbol{p}'_i = \boldsymbol{p}_i - \bar{\boldsymbol{p}}, \quad \boldsymbol{q}'_i(R) = \boldsymbol{q}_i(R) - \bar{\boldsymbol{q}}(R), \tag{22}$$

is dependent on R itself.

However, R can be computed iteratively as follows. First, assume that the kth estimate of R is $R^{(k)}$, $\boldsymbol{t}^{(k)} = \boldsymbol{t}(R^{(k)})$ and $\boldsymbol{q}_i^{(k)} = R^{(k)}\boldsymbol{p}_i + \boldsymbol{t}^{(k)}$. The next estimate, $R^{(k+1)}$ is determined by solving the following absolute orientation problem:

$$R^{(k+1)} = \arg\min_R \sum_{i=1}^{n} \|R\boldsymbol{p}_i + \boldsymbol{t}^{(k)} - F_i\boldsymbol{q}_i^{(k)}\|^2 \tag{23}$$

$$= \arg\max_R \operatorname{tr}(R^t M(R^{(k)})). \tag{24}$$

In this form, the solution for $R^{(k+1)}$ is given by (15). We then compute the next estimate of translation using (18) as

$$\boldsymbol{t}^{(k+1)} = \boldsymbol{t}(R^{(k+1)}). \tag{25}$$

and repeat the process. From (23), we see that the fixed point, R^*, of these iterations will satisfy

$$R^* = \arg\min_R \sum_{i=1}^{n} \|R\boldsymbol{p}_i + \boldsymbol{t}(R) - F_i(R^*\boldsymbol{p}_i + \boldsymbol{t}(R^*))\|^2. \tag{26}$$

3.3 Global Convergence

We now wish to show that the orthogonal iteration algorithm will converge to an optimum of (23) for any set of observed points and any starting point $R^(0)$. Our proof, which follows the development in [38, Chap. 6], first requires the following definition:

Definition 3.1 A point-to-set mapping $\mathbf{A}$ from X to Y is said to be *closed* at $\boldsymbol{x} \in X$ if the assumptions

1. $\boldsymbol{x}_k \to \boldsymbol{x},\ \boldsymbol{x}_k \in X$
2. $\boldsymbol{y}_k \to \boldsymbol{y},\ \boldsymbol{y}_k \in \mathbf{A}(\boldsymbol{x}_k)$

imply

3. $\boldsymbol{y} \in \mathbf{A}(\boldsymbol{x})$

The point-to-set map $\mathbf{A}$ is said to be *closed on* X if it is closed at each point of X.

The following two facts about closed functions are useful:

- A continuous point-to-point mapping is a closed mapping.
- The composition of two closed mappings is a closed mapping.

Define $\mathbf{OI} : \mathbf{SO}(3) \to \mathbf{SO}(3)$ to be the mapping the generates $R^{(k+1)}$ from $R^{(k)}$, that is, $R^{(k+1)} = \mathbf{OI}(R^{(k)})$. According to the Global Convergence Theorem [38], to prove the global convergence of the orthogonal iteration algorithm we need to show that

1. **OI** is closed.
2. All $\{R^{(k)}\}$ generated by **OI** are contained in a compact set.
3. **OI** strictly decreases the objective function unless a solution is reached.

To verify the first condition, we note that **OI** can be considered as the composition of three operations: the computation of $M^k = M(R^k)$, the calculation of the SVD of M^k, and the computation of R^{k+1}. The first and the last of these operations are continuous, and hence

are closed. In Appendix B, it is shown that SVD is also a closed map. Therefore, it follows that **OI** is closed.

Since **OI** always generates orthogonal matrices, and the set of orthogonal matrices $\mathbf{SO}(3)$ is compact (closed and bounded), the second criteria is met.

Finally, we seek to prove the third criteria. Using the projection properties of F_i, we have

$$\begin{aligned}
E(R^{(k+1)}) &= \sum_{i=1}^{n} \|\boldsymbol{q}_i^{(k+1)} - F_i \boldsymbol{q}^{(k+1)}\|^2 \\
&= \sum_{i=1}^{n} \|\boldsymbol{q}_i^{(k+1)} - F_i \boldsymbol{q}_i^{(k)} + F_i \boldsymbol{q}_i^{(k)} - F_i \boldsymbol{q}_i^{(k+1)}\|^2 \\
&= \sum_{i=1}^{n} \|\boldsymbol{q}_i^{(k+1)} - F_i \boldsymbol{q}_i^{(k)}\|^2 + \sum_{i=1}^{n} (\boldsymbol{q}_i^{(k)} - \boldsymbol{q}_i^{(k+1)})^t F_i^t \left(2(\boldsymbol{q}_i^{(k+1)} - F_i \boldsymbol{q}_i^{(k)}) + F_i \boldsymbol{q}_i^{(k)} - F_i \boldsymbol{q}_i^{(k+1)} \right) \\
&= \sum_{i=1}^{n} \|\boldsymbol{q}_i^{(k+1)} - F_i \boldsymbol{q}_i^{(k)}\|^2 + \sum_{i=1}^{n} (\boldsymbol{q}_i^{(k)} - \boldsymbol{q}_i^{(k+1)})^t F_i^t \left(2\boldsymbol{q}_i^{(k+1)} - F_i (\boldsymbol{q}_i^{(k)} + \boldsymbol{q}_i^{(k+1)}) \right) \\
&= \sum_{i=1}^{n} \|\boldsymbol{q}_i^{(k+1)} - F_i \boldsymbol{q}_i^{(k)}\|^2 + \sum_{i=1}^{n} \left(2(F_i \boldsymbol{q}_i^{(k)})^t F_i \boldsymbol{q}_i^{(k+1)} - 2\|F_i \boldsymbol{q}_i^{(k+1)}\|^2 - \|F_i \boldsymbol{q}_i^{(k)}\|^2 + \|F_i \boldsymbol{q}_i^{(k+1)}\|^2 \right) \\
&= \sum_{i=1}^{n} \|\boldsymbol{q}_i^{(k+1)} - F_i \boldsymbol{q}_i^{(k)}\|^2 - \sum_{i=1}^{n} \|F_i \boldsymbol{q}_i^{(k+1)} - F_i \boldsymbol{q}_i^{(k)}\|^2.
\end{aligned} \tag{27}$$

But according to (23) and (25),

$$\sum_{i=1}^{n} \|\boldsymbol{q}_i^{(k+1)} - F_i \boldsymbol{q}_i^{(k)}\|^2 \leq \sum_{i=1}^{n} \|\boldsymbol{q}_i^{(k)} - F_i \boldsymbol{q}_i^{(k)}\|^2 = E(R^{(k)}), \tag{28}$$

and we obtain

$$E(R^{(k+1)}) \leq E(R^{(k)}) - \sum_{i=1}^{n} \|F_i \boldsymbol{q}_i^{(k+1)} - F_i \boldsymbol{q}_i^{(k)}\|^2. \tag{29}$$

Assume that $R^{(k)}$ is not a fixed point of **OI** which implies $R^{(k+1)} \neq R^{(k)}$, and $\boldsymbol{q}_i^{(k+1)} \neq \boldsymbol{q}_i^{(k)}$. If $\sum_{i=1}^{n} \|F_i \boldsymbol{q}_i^{(k+1)} - F_i \boldsymbol{q}_i^{(k)}\|^2$ is equal to zero, then $F_i \boldsymbol{q}_i^{(k+1)} = F_i \boldsymbol{q}_i^{(k)}$. But since the optimal solution to the absolution orientation problem is unique, according to (23), we must have $R^{(k+1)} = R^{(k)}$ which contradicts our assumption that $R^{(k)}$ is not a fixed point. Therefore, $\sum_{i=1}^{n} \|F_i \boldsymbol{q}_i^{(k+1)} - F_i \boldsymbol{q}_i^{(k)}\|^2$ cannot be zero. Combined with (29), we have

$$E(R^{(k+1)}) < E(R^{(k)}) \tag{30}$$

meaning that **OI** decreases E strictly unless a solution is reached.

Now we can claim that the orthogonal iteration algorithm is globally convergent.

3.4 Initialization as weak perspective approximation

How do we get the initial estimate of R, that is $R^{(0)}$? Since the orthogonal iteration algorithm is globally convergent, it should not matter where we start our iterations, but in practice we want to start from some reasonable guess. We show that the initial pose can also be found by performing absolute orientation between the set of reference points and the set of image points considered as coplanar 3D points. We show this initial absolute orientation problem is equivalent to using weak perspective camera model.

Under the weak perspective model, we have the following relation for each reference point $\boldsymbol{p}_i$

$$u_i \approx \frac{1}{s}(\boldsymbol{r}_1^t \boldsymbol{p}_i + t_x) \tag{31a}$$

$$v_i \approx \frac{1}{s}(\boldsymbol{r}_2^t \boldsymbol{p}_i + t_y), \tag{31b}$$

where s is called *scale* or *principle depth*. Weak perspective is valid when the depths of all camera-space coordinates are roughly equal to the principle depth, and the object is close to the optical axis of the camera. Conventionally, the principle depth is chosen as the depth of the origin of the object space, that is, the z-component of the translation t_z. Here, we decouple the scale s from t_z, so it can be chosen as the one that minimizes its deviation from the depths of the camera space coordinates

$$\sum_{i=1}^{n} (\boldsymbol{r}_3^t \boldsymbol{p}_i + t_z - s)^2. \tag{32}$$

Of course, we also need to minimize the square of the image error over R, $\boldsymbol{t}$ and s

$$\sum_{i=1}^{n} \left[(\boldsymbol{r}_1^t \boldsymbol{p}_i + t_x - s u_i)^2 + (\boldsymbol{r}_2^t \boldsymbol{p}_i + t_y - s v_i)^2 \right]. \tag{33}$$

Combining (32) and (33), and weighting them equally, we have the following least squares

objective function:

$$\sum_{i=1}^{n} \|R\boldsymbol{p}_i + \boldsymbol{t} - s\boldsymbol{v}_i\|^2. \tag{34}$$

This is the same objective function as for absolute orientation, (9), but with the additional scale variable and the (implicit) constraint that all camera-space coordinates have the same depth. In this new objective function, the value of s together with R and $\boldsymbol{t}$ must be determined simultaneously.

Horn presents a solution to this problem [35, 27] by considering the following modified objective function:

$$\min_{R,\boldsymbol{t},s} \sum_{i=1}^{n} \|\frac{1}{\sqrt{s}} R\boldsymbol{p}'_i - \sqrt{s}\boldsymbol{q}'_i\|^2. \tag{35}$$

In this case, the solution for s is

$$s = \sqrt{\frac{\sum_{i=1}^{n} \|\boldsymbol{p}'_i\|^2}{\sum_{i=1}^{n} \|\boldsymbol{q}'_i\|^2}}. \tag{36}$$

Furthermore, if the s is computed using (36), the rotation matrix of the pose is independent of s yet it reduces the overall least-squares objective function. This is even true when the 3D reference points are far away from the optical axis, a case where weak perspective is not expected to be a good approximation.

After R and s are determined, $\boldsymbol{t}$ can be computed as

$$\boldsymbol{t} = s\bar{\boldsymbol{v}} - R\bar{\boldsymbol{p}}, \tag{37}$$

where $\bar{\boldsymbol{v}} = \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{v}_i$. Note that if the origin of the object space is placed at $\bar{\boldsymbol{p}}$, i.e. $\bar{\boldsymbol{p}} = \boldsymbol{0}$, then $s = t_z$. The solutions for R and $\boldsymbol{t}$ computed by this method are then used as the initial values to start the iterative descent outlined in the previous section.

It is important to note that the accuracy of the final solution (after iteration) does not depend on how good the geometrical relation of the object and the camera is approximated by the weak-perspective model — this is merely a starting point.

3.5 Modifications for depth-dependent noise

Although the use of object-space collinearity error gives us a fast and globally convergent algorithm, it also introduces some potential problems. In particular, if the observed image points are perturbed by homogeneous Gaussian noise, which is likely to happen, the pose solution will implicitly more heavily weight reference points that are farther away from the camera. This is because the object-space collinearity error increases as the reference point is moved away from the camera. Note, however that this kind of bias is significant only when the object is very close to the camera, or the depth of the object is comparable to the distance between the object and the camera.

We can reduce this bias by slightly modifying the optimization algorithm. Note that the projection operator, F_i, is a function of the image vector $\boldsymbol{v}_i$. If the noise distribution were accounted for, the orthogonal iteration algorithm would involve minimizing the following objective function:

$$\sum_{i=1}^{n} (R\boldsymbol{p}_i + \boldsymbol{t})^t (I - F_i) \boldsymbol{\Lambda}_i^{-1} (I - F_i)(R\boldsymbol{p}_i + \boldsymbol{t}) \tag{38}$$

where $\boldsymbol{\Lambda}_i$ is the covariance matrix associated with F_i due to noise in image point $\boldsymbol{v}_i$. The presence of this matrix prohibits using the orthogonality of the rotation matrix to simplify the dependence of the objective function on R. An exact closed-form solution is not possible unless the orthogonality constraint on rotation is dropped, in which case the problem becomes a linear least squares problem. This linear approach faces the same problems encountered by other linear methods for pose estimation.

Although the general weighted least squares problem cannot be solved, if instead the absolute orientation problem is presented as an equally-weighted or a scalar-weighted least squares, we can still find closed-form solutions in which the orthogonality constraint is fully considered. In order to do this, we must assume that image error for each image coordinate is identical. Supposing that the error in camera-space coordinates is roughly proportional to the depth, the covariance matrix can then be approximated as

$$\boldsymbol{\Lambda}_i^{(k)} \approx (d_i^{(k-1)})^2 aI, \tag{39}$$

where a is some constant, and $d_i^{(k-1)}$ is the depth of $\boldsymbol{q}_i^{(k-1)}$. The absolute orientation problem can now be formulated as a scalar-weighted least squares

$$\sum_{i=1}^{n} \frac{1}{(d_i^{(k)})^2} \|(I - F_i)(R\boldsymbol{p}_i + \boldsymbol{t})\|^2. \tag{40}$$

Such weighting schemes were used in [39, 40] and can be easily incorporated into the algorithm developed above.

4 Performance Evaluation

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

4.1 Data generation protocol

The protocol for generating the input data used throughout this section is governed by the following control parameters: number of points N, signal-to-noise ratio (SNR) and percentage of outliers (PO).

The test data was generated as follows. A set of N 3D reference points were generated uniformly within a box defined by $[-5, 5] \times [-5, 5] \times [-5, 5]$ in the object space. A random 3D rotation was generated by selecting a random unit quaternion from a unit 4-sphere. It can be shown that the distribution of 3D rotations generated by this process is also uniform [41]. For translation, the x and y components were uniformly selected from the interval $[5, 15]$, and the z component was selected from the interval $[20, 50]$. The set of reference points were then transformed by the randomly selected rotation and translation.

Following this, a fraction (= PO) of the 3D points were selected as outliers. Each of these points was replaced by another 3D point whose components were taken from a uniform distribution within a box $[-5, 5] \times [-5, 5] \times [-5, 5]$ in the object space.

Finally, the resulting 3D points were projected onto the normalized image plane to produce image points. Gaussian noise was added to both coordinates of the image points to

generate the perturbed image points. The variance σ of the noise is related to SNR by $\text{SNR} = -20\log(\sigma/0.3)$ dB (the image size is roughly $10/35 \approx 0.3$).

4.2 Standard comparison tests

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

The following three standard tests were conducted on the generated input data:

C1 Set $N = 20$, PO $= 0$. Record the log errors of rotation and translation against SNR (30 dB-70 dB in 10 dB steps). The purpose is to measure how well the tested methods resist noise.

C2 Set $N = 20$, SNR $= 60$ dB. Record the log errors of rotation and translation against PO (5 %-25 % in 5 % steps). The purpose is to see how well the tested methods tolerate outliers.

C3 Set PO $= 0$, SNR $= 50$ dB. Record the log errors of rotation and translation against N (10 to 50 by steps of 10). The purpose is to investigate how the accuracy 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 1,000 trials for each setting of the control parameters. All the comparisons were conducted on a Silicon Graphics IRIS Indigo with a MIPS R4400 processor.

4.2.1 Results and Discussions

The methods tested here are the orthogonal iteration algorithm, a linear method using full perspective camera model [18], and a classical method using Levenberg-Marquardt minimization. An implementation of LM (called `LMDIF`) in MINPACK [2] is used in our experiments.

[2]Visit `http://www.mcs.anl.gov/summaries/minpack93/summary.html` for information about the public-domain package MINPACK-2 that implements these methods.

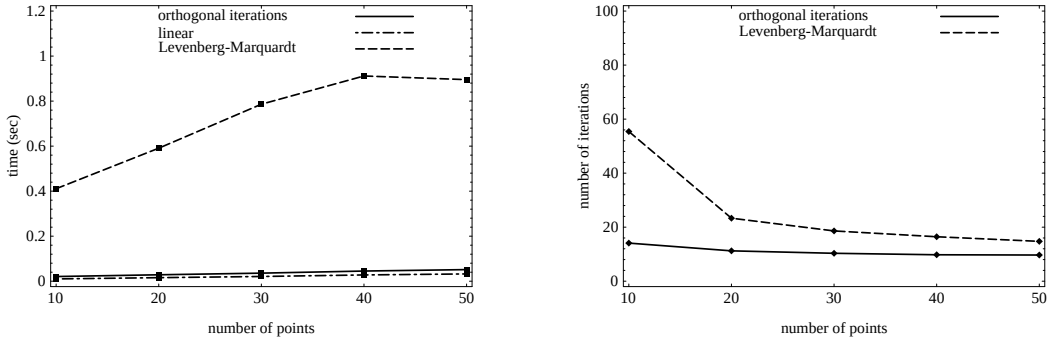

Figure 3: Running times and average numbers of iterations used by the tested methods. Each point in the plot represents 1,000 trials.

LM starts from the same initial solutions as those generated from the orthogonal iteration algorithm. The geometrical configurations are chosen in such a way that the weak-perspective approximation is poor in general. With poor initial guesses, LM behaves like a steepest descent method which exhibits a slow convergence rate. This explains why LM is slower than the proposed method with increasing SNR or PO. On the other hand, the proposed method is as fast as LM when both are initialized with appropriate values. This leads us to believe that the proposed method has quadratic-like local convergence similar to that of the Gauss-Newton method.

Figure 3 shows the average running times and number of iterations of the methods we tested against the number of reference points. These times are measured for SNR = 60 dB and PO = 0. The orthogonal iteration algorithm is clearly much more efficient than LM, having about the same accuracy as LM without outliers (see Figures 4, 6). It significantly outperforms LM in the presence of outliers as shown by Figure 5.

5 Conclusion

In this article, we have presented a fast and globally convergent pose estimation algorithm. Large-scale empirical testing has shown that this algorithm is generally more efficient and no less accurate than the classical Levenberg-Marquardt method in unconstrained geometrical conditions. Hence, the algorithm well suited for any situation especially where both efficiency

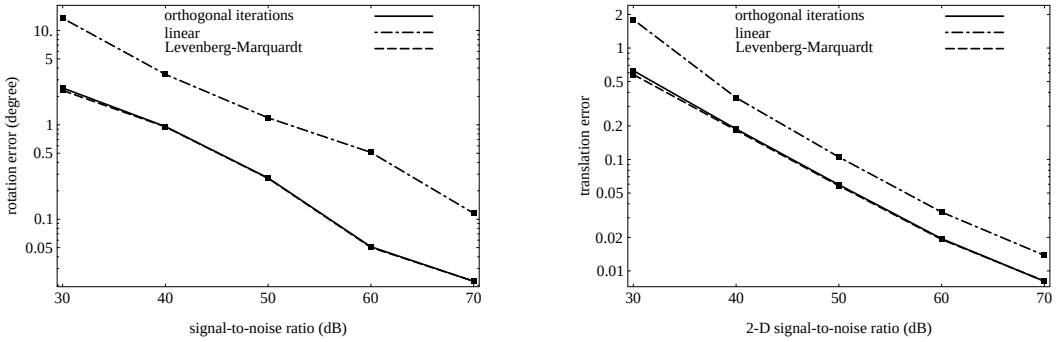

Figure 4: Result of Experiment **C1** for comparing with the Levenberg-Marquardt method. Error is in log scale. Each point in the plot represents 1,000 trials.

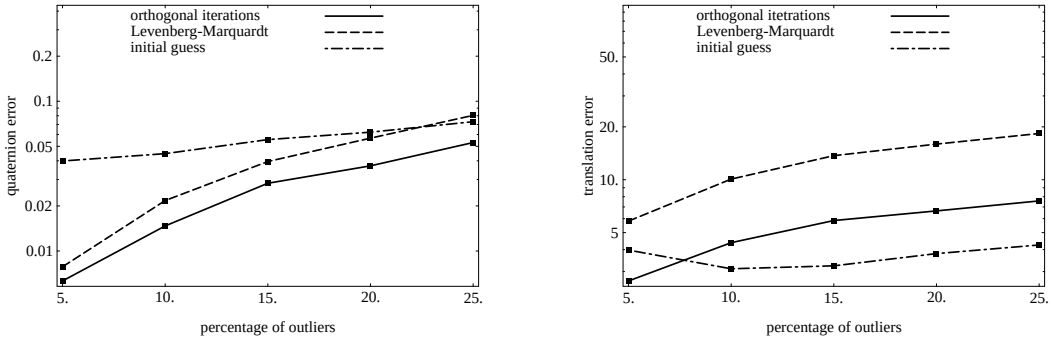

Figure 5: Result of Experiment **C2** for comparing with the Levenberg-Marquardt method. Error is in log scale. Each point in the plot represents 1,000 trials.

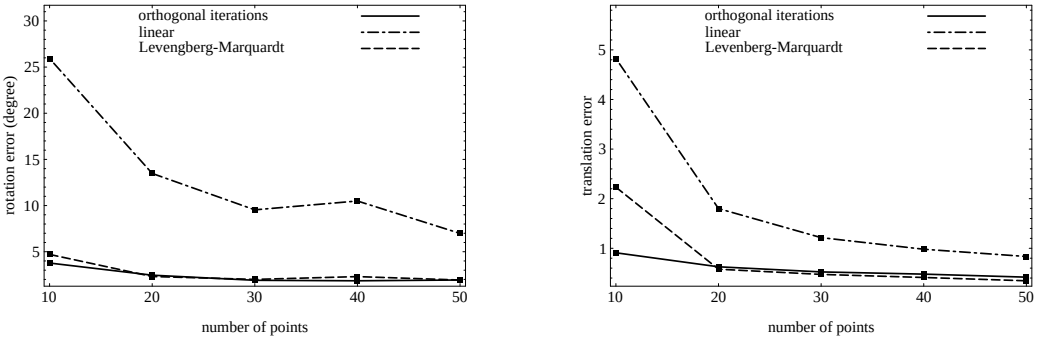

Figure 6: Result of Experiment **C3** for comparing with the Levenberg-Marquardt method. Error is in log scale. Each point in the plot represents 1,000 trials.

and accuracy are desired, and in particular when good prior initialization is not available.

There are several possible extensions to this algorithm. For example, the method can be extended to handle uncertainty in the locations of the *reference points* on the object by slight modification of the objective function. The optimization could also be easily extended to perform a robust optimization step using IRLS methods [42] making it yet more robust to outliers.

We are currently implementing a version of the algorithm within the XVision [43] environment for use in robotic applications, as well as augmented and virtual reality. An initial implementation described in [44] has shown that, by combining efficient local tracking with efficient pose estimation, it is relatively simple to construct real-time object tracking system which run on typical desktop hardware. An interesting extension will be to extend the formalism to include pose estimation from lines, and to compare the efficiency and accuracy with other existing pose tracking system such as demonstrated by Lowe [30].

A Uniqueness of the Optimal Solution to the Absolute Orientation Problem

We show that the best rotation R to (9) is unique. Let

$$M = U\Sigma V^t = \sigma_1 \hat{u}_1 \hat{v}_1^t + \sigma_2 \hat{u}_2 \hat{v}_2^t + \sigma_3 \hat{u}_3 \hat{v}_3^t \tag{41}$$

be a SVD of M, where U and V are orthogonal matrices, and Σ is diagonal. The solution for R is VU^t. U, Σ and V are unique up (1) making the same permutation P of the columns of U, elements of Σ and columns of V, or (2) changing the sign of the corresponding columns of U and V, or (3) replacing columns of U and V corresponding to repeated singular values by any orthonormal basis of the span defined by the columns. This corresponds to rotating the columns by an orthogonal matrix.

For a square matrix M with a SVD $M = U\Sigma V^t$, all three changes do not affect VU^t. Let the new SVD under any of these changes be $U'\Sigma' V'^t$. For rotation, let $U' = UT, V' = VT$, then $V'U'^t = VTT^tU^t = VU^t$ since $TT^t = I$. The same reasoning can be applied to

permutation since permutation matrices are special cases of rotation matrices. Changing signs of corresponding columns of U and V will not change VU^t since $VU^t = \hat{v}_1\hat{u}_1^t + \hat{v}_2\hat{u}_2^t + \hat{v}_3\hat{u}_3^t$

B Closedness of SVD

Suppose that $M_k \to M$, that (U_k, Σ_k, V_k) is an arbitrary SVD of M_k and that $(U_k, \Sigma_k, V_k) \to (U, \Sigma, V)$. To show that SVD, viewed as a point to set mapping, is closed, we must show that (U, Σ, V) is a SVD of M.

From the closedness of $\mathbf{SO}(3)$, U and V are orthonormal matrices. Likewise, the set of diagonal matrices in $\mathbf{GL}(3)$ is a closed subgroup, and hence Σ is a diagonal matrix. Therefore, (U, Σ, V) is a SVD of some matrix $M' = U\Sigma V^t$. However, by the continuity of transposition and matrix multiplication, if $(U_k, \Sigma_k, V_k) \to (U, \Sigma, V)$ then $U_k\Sigma_k V_k^t \to U\Sigma V^t$ and hence $M_k \to M'$. Therefore, $M = M'$ and consequently (U, Σ, V) is a SVD of M.

References

[1] W. Wilson, "Visual servo control of robots using kalman filter estimates of robot pose relative to work-pieces," in *Visual Servoing* (K. Hashimoto, ed.), pp. 71–104, World Scientific, 1994.

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

[3] A. State, G. Hirota, D. CHen, W. Garrett, and M. Livingston, "Superior augmented reality registration by integrating landmark tracking and magnetic tracking," in *Proc. ACM SIGGRAPH*, pp. 429–438, 1996.

[4] R. Azuma and G. Bishop, "Improving static and dynamic registration in an optical see-through HMD," in *Proc. SIGGRAPH*, pp. 197–204, 1994.

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

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

[7] S. Ganapathy, "Decomposition of transformation matrices for robot vision," *Pattern Recognition Letters*, pp. 401–412, 1989.

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

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

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

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

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

[13] G. H. Rosenfield, "The problem of exterior orientation in photogrammetry," *Photogrammetric Engineering*, pp. 536–553, 1959.

[14] E. H. Tompson, "The projective theory of relative orientation," *Photogrammetria*, pp. 67–75, 1968.

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

[16] D. G. Lowe, "Three-dimensional object recognition from single two-dimensional image," *Artificial Intelligence*, vol. 31, pp. 355–395, 1987.

[17] H. Araujo, R. Carceroni, and C. Brown, "A fully projective formulation for Lowe's tracking algorithm," Tech. Rep. Technical Report 641, University of Rochester, 1996.

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

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

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

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

[22] 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.*, vol. 10, no. 3, pp. 713–720, 1988.

[23] D. DeMenthon and L. Davis, "Model-based object pose in 25 lines of code," *IJCV*, vol. 15, pp. 123–141, June 1995.

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

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

[26] R. Horaud, S. Christy, and F. Dornaika, "Object pose: The link between weak perspective, para perspective and full perspective," Tech. Rep. RR-2356, INRIA, Sept. 1994.

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

[28] R. M. H. et. al., "Pose estimation from corresponding point data," *IEEE Trans. Sys. Man Cyber.*, vol. 19, no. 6, pp. 1426–1446, 1989.

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

[30] D. G. Lowe, "Fitting parametrized three-dimensional models to images," *IEEE Trans. Pat. Anal. Machine Intell.*, no. 5, pp. 441–450, 1991.

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

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

[33] S. Wijesoma, D. Wolfe, and R. Richards, "Eye-to-hand coordination for vision-guided robot control applications," *Intl. J. Rob. Res.*, vol. 12, no. 1, pp. 65–78, 1993.

[34] O. Faugeras, *Three-Dimensional Computer Vision.* The MIT Press, 1993.

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

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

[37] 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.*, vol. 9, pp. 698–700, 1987.

[38] D. G. Luenberger, *Linear and Nonlinear Programming*. Reading, Massachusetts: Addison Wesley, 2 ed., 1984.

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

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

[41] D. K. Ed., *Graphics Gems III*, pp. 124–132. Academic Press, 1992.

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

[43] G. D. Hager and K. Toyama, "XVision: A portable substrate for real-time vision applications," *Computer Vision and Image Understanding*, 1998. To Appean, Jan. 1998.

[44] C.-P. Lu, *Online Pose Estimation and Model Matching*. PhD thesis, Yale University, 1995.