Fast and Globally Convergent Pose Estimation from Video Images

Chien-Ping Lu, *Member*, *IEEE*,
Gregory D. Hager, *Member*, *IEEE Computer Society*, and Eric Mjolsness, *Member*, *IEEE*

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.

Index Terms—Pose estimation, absolute orientation, optimization,weak-perspective camera models, numerical optimization.

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 noncollinear 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 has 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 al. [28]. They 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 nonlinearity 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

- *C.-P. Lu is with iBEAM Broadcasting, Corp., 645 Almor Ave., Suite 100, Sunnyvale, CA 94086. E-mail: cplu@ibeam.com.*
- *G.D. Hager is with the Department of Computer Science, The Johns Hopkins University, 3400 N. Charles St., Baltimore, MD 21218. E-mail: hager@cs.jhu.edu.*
- *E. Mjolsness is with the Jet Propulsion Laboratory, MS 126-346, 4800 Oak Grove Dr., Pasadena CA 91109-8099. E-mail: eric.d.mjolsness@jpl.nasa.gov.*

Manuscript received 18 Feb. 1998; revised 24 Feb. 2000; accepted 20 Mar. 2000.
Recommended for acceptance by K. Bowyer.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number 107625.

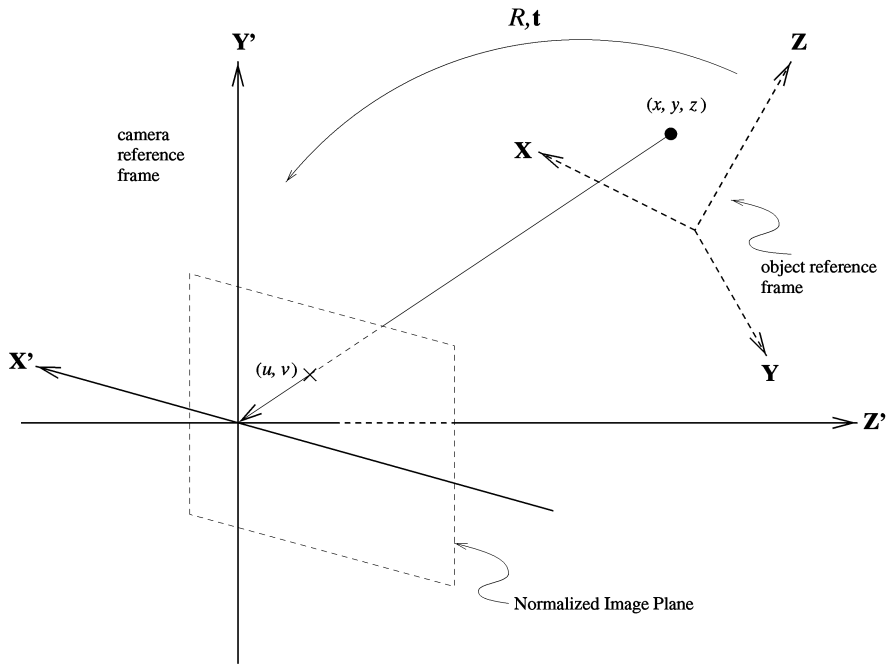

Fig. 1. The reference frames in the pose estimation problem.

algorithm that 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 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 five 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.

1.1 Outline of the Article

The remainder of this article is organized as follows: Section 2 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 noncollinear 3D coordinates of reference points $\mathbf{p}_i = (x_i, y_i, z_i)^t, i = 1, \ldots, n, n \geq 3$ expressed in an object-centered reference frame, the corresponding camera-space coordinates $\mathbf{q}_i = (x_i', y_i', z_i')^t$, are related by a rigid transformation as:

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

where

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

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 $\mathbf{p}_i$ 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 $\mathbf{v}_i = (u_i, v_i, 1)^t$ be the projection of $\mathbf{p}_i$ on the normalized image plane. Under the idealized pinhole imaging model, $\mathbf{v}_i$, $\mathbf{q}_i$ and the center of projection are collinear. This fact is expressed by the following equation:

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

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

or

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.

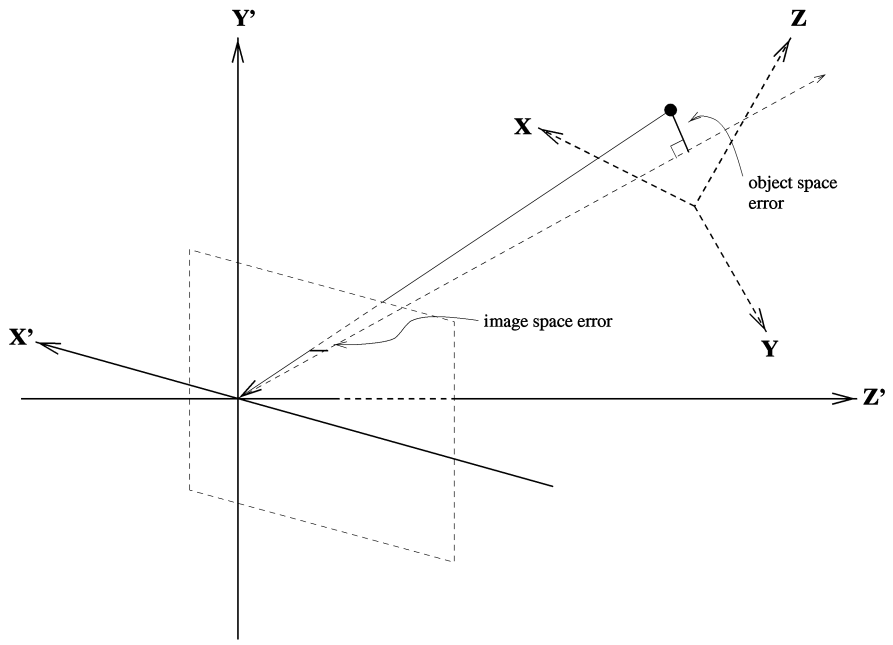

Fig. 2. Object-space and image-space collinearity errors.

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

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

$$R\mathbf{p}_i + \mathbf{t} = V_i(R\mathbf{p}_i + \mathbf{t}), \tag{5}$$

where

$$V_i = \frac{\mathbf{v}_i \mathbf{v}_i^t}{\mathbf{v}_i^t \mathbf{v}_i} \tag{6}$$

is the line-of-sight projection matrix that, when applied to a scene point, projects the point orthogonally to the line of sight defined by the image point $\mathbf{v}_i$. Since V_i is a projection operator, it satisfies the following properties:

$$\|\mathbf{x}\| \geq \|V_i\mathbf{x}\|, \quad \mathbf{x} \in \mathcal{R}^3, \tag{7a}$$

$$V_i^t = V_i, \tag{7b}$$

$$V_i^2 = V_i V_i^t = V_i. \tag{7c}$$

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 $(R, \mathbf{t})$ that minimizes some form of accumulation of the errors (for example, summation of squared errors) of either of the collinearity equations (see Fig. 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:

$$\sum_{i=1}^{n}\left[\left(\hat{u}_i - \frac{\mathbf{r}_1^t\mathbf{p}_i + t_x}{\mathbf{r}_3^t\mathbf{p}_i + t_z}\right)^2 + \left(\hat{v}_i - \frac{\mathbf{r}_2^t\mathbf{p}_i + t_y}{\mathbf{r}_3^t\mathbf{p}_i + t_z}\right)^2\right], \tag{8}$$

given observed image points $\hat{\mathbf{v}}_i = (\hat{u}_i, \hat{v}_i, 1)^t$, which are usually modeled as theoretical image points perturbed by Gaussian noise. The rotation matrix, R, is usually parameterized using Euler angles. Note that this is a minimization over image-space collinearity.

Two commonly used optimization 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 percent 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], [34] and augmented reality demand pose estimation algorithms be not only accurate, but also be robust to corrupted data and be computationally efficient. Hence, there 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 $\mathbf{q}_i$ 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

$$\mathbf{q}_i = R\mathbf{p}_i + \mathbf{t}. \tag{9}$$

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

$$\min_{R,\mathbf{t}} \sum_{i=1}^{n} \|R\mathbf{p}_i + \mathbf{t} - \mathbf{q}_i\|^2, \quad \text{subject to } R^tR = I. \tag{10}$$

Such a constrained least-squares problem [35] can be solved in closed form using quaternions [36], [37] or singular value decomposition (SVD) [27], [38], [36], [37].

The SVD solution proceeds as follows: Let $\{\mathbf{p}_i\}$ and $\{\mathbf{q}_i\}$ denote lists of corresponding vectors related by (1) and define

$$\bar{\mathbf{p}} \stackrel{\text{def}}{=} \frac{1}{n}\sum_{i=1}^{n}\mathbf{p}_i, \quad \bar{\mathbf{q}} \stackrel{\text{def}}{=} \frac{1}{n}\sum_{i=1}^{n}\mathbf{q}_i, \tag{11}$$

that is, $\bar{\mathbf{p}}$ and $\bar{\mathbf{q}}$ are the centroid of $\{\mathbf{p}_i\}$ and $\{\mathbf{q}_i\}$, respectively. Define

$$\mathbf{p}'_i = \mathbf{p}_i - \bar{\mathbf{p}}, \quad \mathbf{q}'_i = \mathbf{q}_i - \bar{\mathbf{q}}, \tag{12}$$

and

$$M = \sum_{i=1}^{n} \mathbf{q}'_i \mathbf{p}'^t_i. \tag{13}$$

In other words, $\frac{1}{n}M$ is the sample cross-covariance matrix between $\{\mathbf{p}_i\}$ and $\{\mathbf{q}_i\}$. It can be shown that [27] if R^*, $\mathbf{t}^*$ minimize (10), then they satisfy

$$R^* = \arg\max_R \operatorname{tr}(R^tM) \tag{14}$$

$$\mathbf{t}^* = \bar{\mathbf{q}} - R^*\bar{\mathbf{p}}. \tag{15}$$

Let (U, Σ, V) be a SVD of M, that is, $U^tMV = \Sigma$. Then, the solution to (10) is

$$R^* = VU^t. \tag{16}$$

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 (13). 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 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 (see Fig. 2):

$$\mathbf{e}_i = (I - \hat{V}_i)(R\mathbf{p}_i + \mathbf{t}), \tag{17}$$

where $\hat{V}_i$ is the observed line-of-sight projection matrix defined as:

$$\hat{V}_i = \frac{\hat{\mathbf{v}}_i\hat{\mathbf{v}}_i^t}{\hat{\mathbf{v}}_i^t\hat{\mathbf{v}}_i}. \tag{18}$$

We then seek to minimize the sum of the squared error

$$E(R,\mathbf{t}) = \sum_{i=1}^{n}\|\mathbf{e}_i\|^2 = \sum_{i=1}^{n}\|(I - \hat{V}_i)(R\mathbf{p}_i + \mathbf{t})\|^2 \tag{19}$$

over R and $\mathbf{t}$. Note that all the information contained in the set of the observed image points $\{\mathbf{v}_i\}$ is now completely encoded in the set of projection matrices $\{\hat{V}_i\}$. Since this objective function is quadratic in $\mathbf{t}$, given a fixed rotation R, the optimal value for $\mathbf{t}$ can be computed in closed form as:

$$\mathbf{t}(R) = \frac{1}{n}\left(I - \frac{1}{n}\sum_j \hat{V}_j\right)^{-1}\sum_j (\hat{V}_j - I)R\mathbf{p}_j. \tag{20}$$

For (20) to be well-defined, $I - \frac{1}{n}\sum_{i=1}^{n} \hat{V}_i$ must be positive definite, which can be verified as follows:

For any 3-vector $\mathbf{x} \in \mathcal{R}^3$, it can be shown that

$$\begin{aligned}&\mathbf{x}^t(I - \frac{1}{n}\sum_{i=1}^{n} \hat{V}_i)\mathbf{x}\\ =&\frac{1}{n}\sum_{i=1}^{n}(\|\mathbf{x}\|^2 - \mathbf{x}^t\hat{V}_i\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n}(\|\mathbf{x}\|^2 - \mathbf{x}^t\hat{V}_i^t\hat{V}_i\mathbf{x}) \qquad (21)\\ =&\frac{1}{n}\sum_{i=1}^{n}(\|\mathbf{x}\|^2 - \|\hat{V}_i\mathbf{x}\|^2) > 0.\end{aligned}$$

While $\|\mathbf{x}\|^2 - \|\hat{V}_i\mathbf{x}\|^2$ can be individually greater than or equal to zero, they cannot be all equal to zero unless all scene points are projected to the same image point. Therefore, (21) is generally strictly greater than zero and, thus, the positive definiteness of $\hat{V}_i$ is asserted.

Given the optimal translation as a function of R and defining

$$\mathbf{q}_i(R) \stackrel{\text{def}}{=} \hat{V}_i(R\mathbf{p}_i + \mathbf{t}(R)) \quad \text{and} \quad \bar{\mathbf{q}}(R) \stackrel{\text{def}}{=} \frac{1}{n}\sum_{i=1}^{n} \mathbf{q}_i(R), \quad (22)$$

(19) can be rewritten as:

$$E(R) = \sum_{i=1}^{n} \|R\mathbf{p}_i + \mathbf{t}(R) - \mathbf{q}_i(R)\|^2. \quad (23)$$

This equation now bears a close resemblance to the absolute orientation problem (compare with (10)). Unfortunately, in this case, we cannot solve for R in closed form as the sample cross-covariance matrix between $\{\mathbf{p}_i\}$ and $\{\mathbf{q}_i(R)\}$, that is,

$$M(R) = \sum_{i=1}^{n} \mathbf{q}_i'(R)\mathbf{p}_i'^t \quad \text{where} \quad \mathbf{p}_i' = \mathbf{p}_i - \bar{\mathbf{p}}, \quad \mathbf{q}_i'(R) = \mathbf{q}_i(R) - \bar{\mathbf{q}}(R), \quad (24)$$

is dependent on R itself.

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

$$R^{(k+1)} = \arg\min_R \sum_{i=1}^{n} \|R\mathbf{p}_i + \mathbf{t} - \hat{V}_i\mathbf{q}_i^{(k)}\|^2 \quad (25)$$

$$= \arg\max_R \operatorname{tr}\left(R^t M(R^{(k)})\right), \quad (26)$$

where the set of $\hat{V}_i\mathbf{q}_i^{(k)}$ is treated as a hypothesis of the set of the scene points $\mathbf{q}_i$ in (10). In this form, the solution for $R^{(k+1)}$ is given by (16). We then compute the next estimate of translation, using (20), as:

$$\mathbf{t}^{(k+1)} = \mathbf{t}\left(R^{(k+1)}\right) \quad (27)$$

and repeat the process. A *solution* R^* to the pose estimation problem using the orthogonal iteration algorithm is defined to be a fixed point to (25), that is, R^* satisfies

$$R^* = \arg\min_R \sum_{i=1}^{n} \|R\mathbf{p}_i + \mathbf{t} - \hat{V}_i(R^*\mathbf{p}_i + \mathbf{t}(R^*))\|^2. \quad (28)$$

Note that a *solution* does not necessarily correspond to the *correct* true pose.

3.3 Global Convergence

We now wish to show that the orthogonal iteration algorithm will converge to an optimum of (25) for any set of observed points and any starting point $R^{(0)}$. Our proof, which is based on the Global Convergence Theorem [39, chapter 6], requires the following definitions:

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

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

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

Note that continuous point-to-point mappings are special closed point-to-set mappings.

Definition 3.2. *A set* $\mathbf{S}$ *is said to be* closed *if* $\mathbf{x}_k \to \mathbf{x}$ *with* $\mathbf{x}_k \in \mathbf{S}$ *implies* $\mathbf{x} \in \mathbf{S}$. $\mathbf{S}$ *is said to be* compact *if it is both closed and bounded.*

Define $\mathbf{OI} : \mathcal{SO}(3) \mapsto \mathcal{SO}(3)$ to be the mapping that generates $R^{(k+1)}$ from $R^{(k)}$, that is, $R^{(k+1)} = \mathbf{OI}(R^{(k)})$. According to the Global Convergence Theorem [39], 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 mappings:

$\mathbf{F} : \mathcal{SO}(3) \mapsto \mathcal{R}^{3\times 3}$ is a point-to-point mapping that represents the computation of $M^{(k)} = M(R^{(k)})$ in (24).

$\mathbf{SVD} : \mathcal{R}^{3\times 3} \mapsto \mathcal{SO}(3) \times \mathcal{GL}(3) \times \mathcal{SO}(3)$ is a point-to-set mapping that represents the calculation of the SVD of $M^{(k)}$.

$\mathbf{G} : \mathcal{SO}(3) \times \mathcal{GL}(3) \times \mathcal{SO}(3) \mapsto \mathcal{SO}(3)$ is a point-to-point mapping that represents the computation of $R^{(k+1)}$ from the SVD of $M(R^{(k)})$ using (16),

where $\mathcal{SO}(3)$ is the set of 3×3 orthogonal matrices and $\mathcal{GL}(3)$ is the set of 3×3 diagonal matrices.

The first and the last mappings, **F** and **G**, are continuous and, hence, are closed. In Appendix A, it is shown that **SVD** is also a closed mapping. Therefore, it follows that **OI** is closed using the fact that the composition of closed mappings is also closed [39].

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

Finally, we seek to prove the third criteria. The sum of squared error of the estimate $R^{(k+1)}$ can be related to that of $R^{(k)}$ as follows:

$$
\begin{aligned}
&E\left(R^{(k+1)}\right)\\
&=\sum_{i=1}^{n}\|\mathbf{q}_i^{(k+1)}-\hat{V}_i\mathbf{q}^{(k+1)}\|^2\\
&=\sum_{i=1}^{n}\|\mathbf{q}_i^{(k+1)}-\hat{V}_i\mathbf{q}_i^{(k)}+\hat{V}_i\mathbf{q}_i^{(k)}-\hat{V}_i\mathbf{q}_i^{(k+1)}\|^2\\
&=\sum_{i=1}^{n}\|\mathbf{q}_i^{(k+1)}-\hat{V}_i\mathbf{q}_i^{(k)}\|^2\\
&\quad+\sum_{i=1}^{n}\left(\mathbf{q}_i^{(k)}-\mathbf{q}_i^{(k+1)}\right)^t\hat{V}_i^t\\
&\quad\left(2\left(\mathbf{q}_i^{(k+1)}-\hat{V}_i\mathbf{q}_i^{(k)}\right)+\hat{V}_i\mathbf{q}_i^{(k)}-\hat{V}_i\mathbf{q}_i^{(k+1)}\right)\\
&=\sum_{i=1}^{n}\|\mathbf{q}_i^{(k+1)}-\hat{V}_i\mathbf{q}_i^{(k)}\|^2\\
&\quad+\sum_{i=1}^{n}\left(\mathbf{q}_i^{(k)}-\mathbf{q}_i^{(k+1)}\right)^t\hat{V}_i^t\left(2\mathbf{q}_i^{(k+1)}-\hat{V}_i\left(\mathbf{q}_i^{(k)}+\mathbf{q}_i^{(k+1)}\right)\right).
\end{aligned}
\tag{29}
$$

Applying the fact that $\hat{V}_i=\hat{V}_i^t\hat{V}_i$ to the second term in the righthand side of the last equation in (29), we have

$$
\begin{aligned}
&\sum_{i=1}^{n}\left(2\left(\hat{V}_i\mathbf{q}_i^{(k)}\right)^t\hat{V}_i\mathbf{q}_i^{(k+1)}-2\|\hat{V}_i\mathbf{q}_i^{(k+1)}\|^2\right.\\
&\quad\left.-\|\hat{V}_i\mathbf{q}_i^{(k)}\|^2+\|\hat{V}_i\mathbf{q}_i^{(k+1)}\|^2\right)=-\sum_{i=1}^{n}\|\hat{V}_i\mathbf{q}_i^{(k+1)}-\hat{V}_i\mathbf{q}_i^{(k)}\|^2.
\end{aligned}
\tag{30}
$$

But, according to (25) and (27),

$$
\sum_{i=1}^{n}\|\mathbf{q}_i^{(k+1)}-\hat{V}_i\mathbf{q}_i^{(k)}\|^2\le\sum_{i=1}^{n}\|\mathbf{q}_i^{(k)}-\hat{V}_i\mathbf{q}_i^{(k)}\|^2=E\left(R^{(k)}\right)
\tag{31}
$$

and we obtain

$$
E\left(R^{(k+1)}\right)\le E\left(R^{(k)}\right)-\sum_{i=1}^{n}\|\hat{V}_i\mathbf{q}_i^{(k+1)}-\hat{V}_i\mathbf{q}_i^{(k)}\|^2.
\tag{32}
$$

Assume that $R^{(k)}$ is not a fixed point of **OI**, which implies $R^{(k+1)}\neq R^{(k)}$ and $\mathbf{q}_i^{(k+1)}\neq\mathbf{q}_i^{(k)}$. If $\sum_{i=1}^{n}\|\hat{V}_i\mathbf{q}_i^{(k+1)}-\hat{V}_i\mathbf{q}_i^{(k)}\|^2$ is equal to zero, then $\hat{V}_i\mathbf{q}_i^{(k+1)}=\hat{V}_i\mathbf{q}_i^{(k)}$. But since the optimal solution to the absolution orientation problem is unique, according to (25), we must have $R^{(k+1)}=R^{(k)}$, which contradicts our assumption that $R^{(k)}$ is not a fixed point. Therefore, $\sum_{i=1}^{n}\|\hat{V}_i\mathbf{q}_i^{(k+1)}-\hat{V}_i\mathbf{q}_i^{(k)}\|^2$ cannot be zero. Combined with (32), we have

$$
E\left(R^{(k+1)}\right)<E\left(R^{(k)}\right),
\tag{33}
$$

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

Now, we can claim that the orthogonal iteration algorithm is globally convergent, that is, a solution, or a fixed point, will eventually be reached from arbitrary starting point. Although global convergence does not guarantee that the true pose will always be recovered, it does suggest that the true pose can be reached from very a broad range of initial guesses. Based on the experiments in Section 4.1, we have empirically observed that the only constraint on $R^{(0)}$ for **OI** to recover the true pose is that it does not result in translation with negative z component, i.e., it does not place the object behind the camera.

3.4 Initialization and Weak Perspective Approximation

The **OI** algorithm can be initiated as follows: Given an initial guess $R^{(0)}$ of R, compute $\mathbf{t}^{(0)}$. The initial pose $(R^{(0)},\mathbf{t}^{(0)})$ is then used to establish a set of hypothesized scene points $\hat{V}_i(R^{(0)}\mathbf{p}_i+\mathbf{t}^{(0)})$, which are used to start the first absolute orientation iteration. Although the orthogonal iteration algorithm is globally convergent, it does not guarantee that it will efficiently or eventually converge to the correct solution. Instead of choosing $R^{(0)}$, we can treat $\mathbf{v}_i$ themselves as the first hypothesized scene points. This leads to an absolute orientation problem between the set of 3D reference points $\mathbf{p}_i$ and the set of image points $\mathbf{v}_i$ considered as coplanar 3D points. This initial absolute orientation problem is related to weak perspective approximation.

3.4.1 Weak-Perspective Model

Weak-perspective is an approximation to the perspective camera model described in Section 2.1. Under the weak perspective model, we have the following relation for each reference point $\mathbf{p}_i$

$$
u_i\approx\frac{1}{s}(\mathbf{r}_1^t\mathbf{p}_i+t_x)
\tag{34a}
$$

$$
v_i\approx\frac{1}{s}(\mathbf{r}_2^t\mathbf{p}_i+t_y),
\tag{34b}
$$

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 when $\bar{\mathbf{p}}$, the center of the reference points, is also the origin of the object space. 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

$$
\sum_{i=1}^{n}\left(\mathbf{r}_3^t\mathbf{p}_i+t_z-s\right)^2.
\tag{35}
$$

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

$$
\sum_{i=1}^{n}\left[\left(\mathbf{r}_1^t\mathbf{p}_i+t_x-s\hat{u}_i\right)^2+\left(\mathbf{r}_2^t\mathbf{p}_i+t_y-s\hat{v}_i\right)^2\right].
\tag{36}
$$

Combining (35) and (36), and weighting them equally, we have the following least-squares objective function:

$$
\sum_{i=1}^{n}\|R\mathbf{p}_i+\mathbf{t}-s\hat{\mathbf{v}}_i\|^2.
\tag{37}
$$

This is the same objective function as for absolute orientation, (10), 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 $\mathbf{t}$ must be determined simultaneously.

By considering the following modified objective function [36], [27]

$$\min_{R,\mathbf{t},s} \sum_{i=1}^{n} \| \frac{1}{\sqrt{s}} R\mathbf{p}'_i - \sqrt{s}\mathbf{q}'_i \|^2, \tag{38}$$

the solution for s can be found to be

$$s = \sqrt{\frac{\sum_{i=1}^{n} \|\mathbf{p}'_i\|^2}{\sum_{i=1}^{n} \|\mathbf{q}'_i\|^2}}. \tag{39}$$

The rotation matrix of the pose is independent of s, yet it reduces the overall least-squares objective function. After R and s are determined, $\mathbf{t}$ can be computed as:

$$\mathbf{t} = s\bar{\mathbf{v}} - R\bar{\mathbf{p}}, \tag{40}$$

where $\bar{\mathbf{v}} = \frac{1}{n}\sum_{i=1}^{n} \hat{\mathbf{v}}_i$. Note that if the origin of the object space is placed at $\bar{\mathbf{p}}$, i.e., $\bar{\mathbf{p}} = \mathbf{0}$, then $s = t_z$.

3.4.2 Initial Absolute Orientation Solution

With the **OI** algorithm, the initial rotation will be the same as those computed using the aforementioned weak-perspective algorithm, however, the translation is different in that it is computed using (20) as a result of optimizing (19). Let us rewrite (20) here

$$\mathbf{t}(R) = \frac{1}{n}\left(I - \frac{1}{n}\sum_j \hat{V}_j \right)^{-1} \sum_j \left(\hat{V}_j - I\right) R\mathbf{p}_j. \tag{41}$$

Comparing (40) and (41), we find that the former is approximated by the latter if the following conditions hold:

$$\left(I - \frac{1}{n}\sum_j \hat{V}_j \right)^{-1} \approx I \tag{42}$$

$$\frac{1}{n}\sum_j \hat{V}_j R\mathbf{p}_j \approx s\bar{\mathbf{v}} \quad \text{for some } s > 0. \tag{43}$$

The first condition states that the scene points are located close to the optical axis and the second condition states that the scene points are distributed like a plane parallel to the image plane. These two conditions closely resemble the conditions under which weak-perspective approximation is valid.

In summary, we have reformulated the pose estimation problem under the weak-perspective model as the problem of fitting the set of the reference points to a planar projection of the image points. Using the image points themselves as the hypothesized scene points in the initial absolute orientation iteration results in a pose solution better than the unmodified weak-perspective solution. This pose solution serves, therefore, as a good initial guess for the subsequent iterative refinement.

3.5 Depth-Dependent Noise

The global convergence of the **OI** algorithm is attained at the expense of being biased when the observed image points are perturbed by homogeneous Gaussian noise. 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.

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

$$\sum_{i=1}^{n} (R\mathbf{p}_i + \mathbf{t})^t (I - \hat{V}_i)\mathbf{\Lambda}_i^{-1}(I - \hat{V}_i)(R\mathbf{p}_i + \mathbf{t}), \tag{44}$$

where $\mathbf{\Lambda}_i$ is the covariance matrix associated with $\hat{V}_i$ due to noise in image point $\mathbf{v}_i$. 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:

$$\mathbf{\Lambda}_i^{(k)} \approx \left(d_i^{(k-1)} \right)^2 aI, \tag{45}$$

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

$$\sum_{i=1}^{n} \frac{1}{\left(d_i^{(k)} \right)^2} \| R\mathbf{p}_i + \mathbf{t} - \hat{V}_i \left(R^{(k)}\mathbf{p}_i + \mathbf{t}^{(k)} \right) \|^2. \tag{46}$$

Such weighting schemes were used in [40], [41] and can be easily incorporated into the algorithm developed above.

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. According our experiments in Section 4.1, the bias is noticeable only when the ratio between the size of the object in the direction of optical axis and distance to camera is smaller than 3.5, in which cases unbiased methods like Levenberg-Marquardt may be preferable.

4 Experiments

We have developed implementations of **OI** in both C++ and in Matlab. The code for the latter can be found from http://www.cs.jhu.edu/~hager. These implementations have been tested in both simulation and on real data and have also

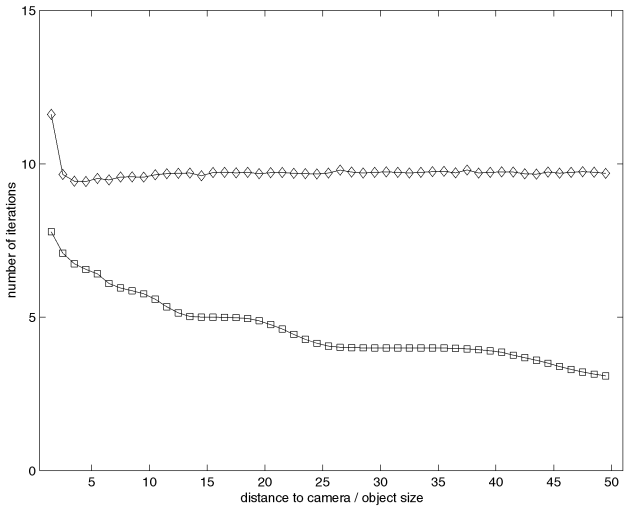

Fig. 3. Number of iterations as a function of distance to camera. The results for **OI** initialized with weak-perspective pose are plotted as squares (□) and the results for **OI** randomly initialized are plotted as diamonds (◇). Each point represents results averaged over 1,000 uniformly distributed rotations.

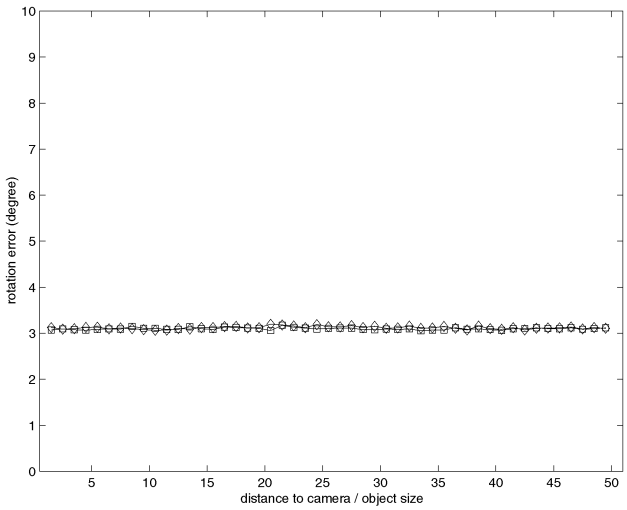

Fig. 4. Rotation error as a function of distance to camera. The results for **OI** initialized with weak-perspective pose are plotted as squares (□) and the results for **OI** randomly initialized are plotted as diamonds (◇). Each point represents results averaged over 1,000 uniformly distributed rotations.

been compared with implementations of other pose estimation algorithms. The results of these experiments are detailed below.

4.1 Dependence on Object Location and Initial Guesses

In this section, we evaluate the proposed algorithm as a function of distances to camera and to optical axis, respectively. The purpose is to study the following three aspects of the algorithm:

1. The performance of **OI** when it is initialized with a weak-perspective pose as a function of how sufficient the weak-perspective model approximates the true perspective camera model. The validity of the weak-perspective model can be characterized by the following two parameters:
   - distance to camera and
   - distance to optical axis.
2. The effect of the bias described in Section 3.5 when the distance between the object and the camera is relatively small compared to the size of the object in the direction of optical axis.
3. The performance of **OI** when it is randomly initialized compared to that of **OI** when it is initialized with a weak-perspective pose.

The set of 3D reference points are defined as the eight corners of the box defined by $[-5,5] \times [-5,5] \times [-5,5]$ in the object space. The translation vector is varied with increasing distance to camera and with increasing distance to optical axis. Uniformly distributed random 3D rotation is generated for each translation [42]. The set of reference points are then transformed by the selected rotation and translation.

Finally, the resulting 3D points are projected onto the normalized image plane to produce image points. Gaussian noise with signal-to-noise ratio (SNR) = 70 dB is added to both coordinates of the image points to generate the perturbed image points. The variance σ of the noise is related to signal-to-noise ratio (SNR) by $\mathrm{SNR} = -20\log(\sigma t_z/10)$ dB.

The following two tests are conducted on the generated input data:

D1. The translation vector is generated by fixing $t_x = 5$ and $t_y = 5$, and varies $t_z/10$ from 1.5 to 50 by a step of 1. The purpose is to measure how well the proposed algorithm performs when the target object is approaching the camera.

D2. The translation vector is generated by fixing $t_x = 5$ and $t_z = 200$ and varies $t_y/10$ from 1.5 to 50 by a step of 1. The purpose is to measure how well the proposed algorithm performs when the target object is moving away from the optical axis.

The distances are expressed relative to the object size. For each translation, the average rotation error and translation error are computed over 1,000 uniformly distributed rotation matrices.

4.1.1 Results and Discussions

Depth-dependent noise. From Fig. 4 and Fig. 5, we can see that, as the target object moves closer to the camera, the translation error increases due to the bias of **OI** toward points farther away from camera. However, the effect is significant only when the ratio between the distance to camera and the object size in z direction is smaller than 3.5. Beyond this, the average translation error remains almost constant as the object moves away from the camera when image points are perturbed by noise with the same SNR.

It is interesting to see that the average rotation error is almost not affected by the bias. It seems that the biasing effects introduced by each $\hat{V}_i$ are canceled by each other during the computation of rotation, while their influences remain significant in the computation of translation using (20).

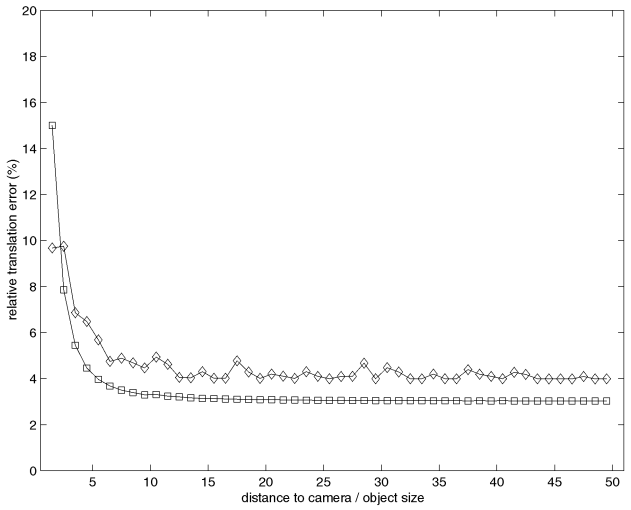

Fig. 5. Translation error as a function of distance to camera. The results for **OI** initialized with weak-perspective pose are plotted as squares (□), and the results for **OI** randomly initialized are plotted as diamonds (◇). Each point represents result averaged over 1,000 uniformly distributed rotations.

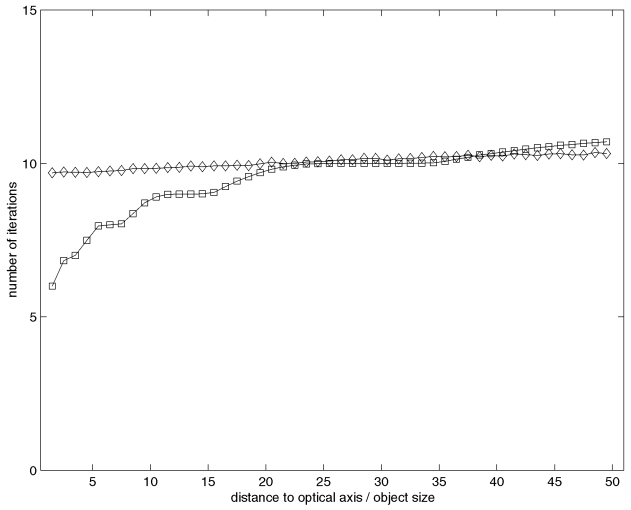

Fig. 6. Number of iterations as a function of distance to optical axis. The results for **OI** initialized with weak-perspective pose are plotted as squares (□) and the results for **OI** randomly initialized are plotted as diamonds (◇). Each point represents results averaged over 1,000 uniformly distributed rotations.

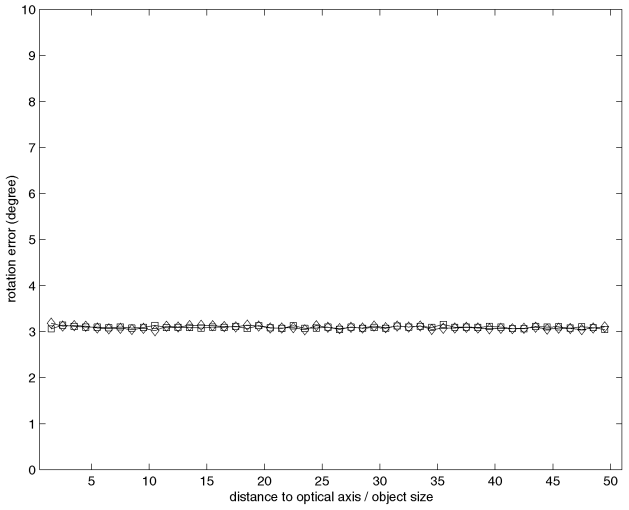

Fig. 7. Rotation error as a function of distance to optical axis. The results for **OI** initialized with weak-perspective pose are plotted as squares (□) and the results for **OI** randomly initialized are plotted as diamonds (◇). Each point represents results averaged over 1,000 uniformly distributed rotations.

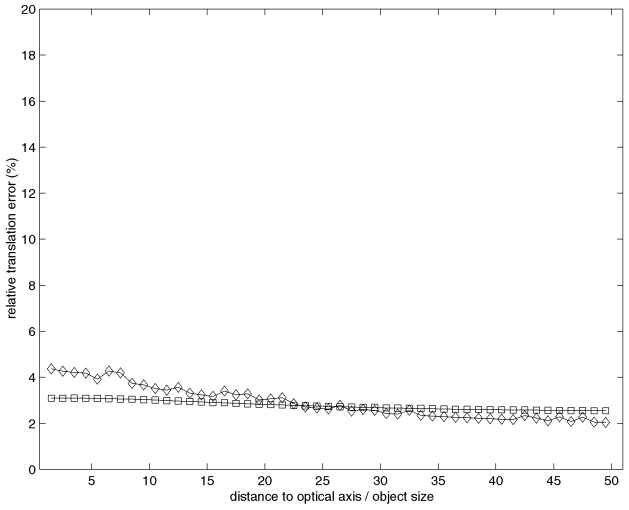

Fig. 8. Translation error as a function of distance to optical axis. The results for **OI** initialized with weak-perspective pose are plotted as squares (□) and the results for **OI** randomly initialized are plotted as diamonds (◇). Each point represents results averaged over 1,000 uniformly distributed rotations.

Weak-perspective approximation. Besides the effect of depth-dependent noise, we can see that the average rotation and translation errors do not vary significantly as the distance to camera and the distance to optical axis change (see the plots of squares (□) in Figs. 4, 5, 7, and 8). However, the number of iterations does decrease (see plots of squares (□) in Figs. 3 and 6) as the camera model is better approximated by weak-perspective model when the object moves away from the camera and/or approaches the optical axis. This means that, when the weak-perspective pose is closer to the true pose, **OI** can converge to it faster. However, even if the weak-perspective pose is not close enough, **OI** can still reach it with the same accuracy. It just takes more steps.

Initial guesses. When **OI** is initialized with randomly generated rotation, the average number of iterations taken by **OI** to converge is roughly the same for different object locations and is generally higher than when using weak-perspective initialization (see plots of diamonds (◇) in Figs. 3 and 6). From Fig. 4 and 7, we can see that the average rotation error is almost the same as with weak-perspective initialization, while average translation error varies within a range of 2 percent of the true translation (see plots of diamonds (◇) in Figs. 5 and 8).

With more than six point correspondences, one expects a unique pose solution [43] under noiseless conditions. Although in noisy cases there may be a few spurious fixed points to which **OI** converges, our experiments show that, by merely constraining initial rotation so that it does not

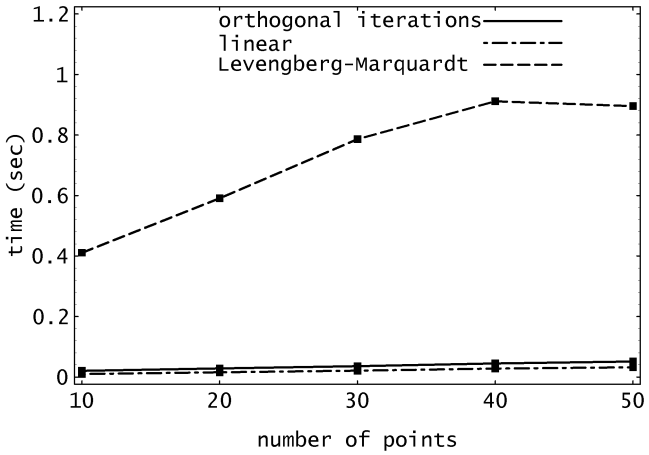

Fig. 9. Average running times used by the tested methods. Each point in the plot represents 1,000 trials.

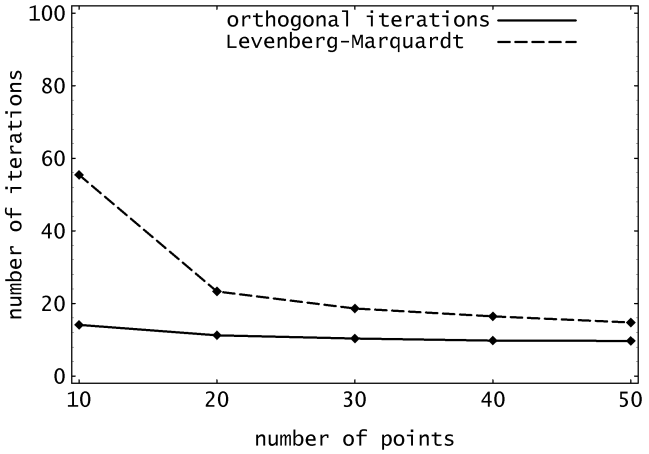

Fig. 10. Average numbers of iterations used by the tested methods. Each point in the plot represents 1,000 trials.

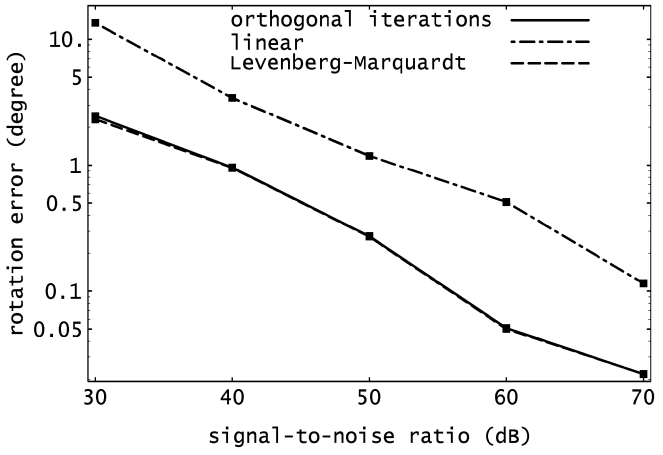

Fig. 11. Result (average rotation errors) of Experiment **C1** for comparing with the Levenberg-Marquardt method. Error is in log scale. Each point in the plot represents 1,000 trials.

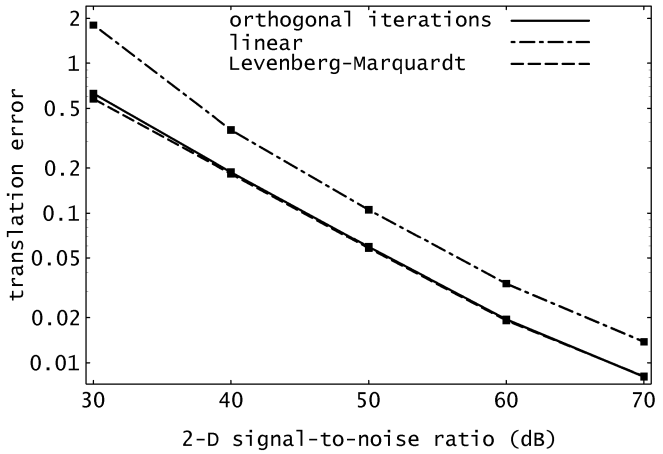

Fig. 12. Result (average translation error) of Experiment **C1** for comparing with the Levenberg-Marquardt method. Error is in log scale. Each point in the plot represents 1,000 trials.

place the object behind the camera, **OI** seems to be able to reach the true pose.

4.2 Comparisons to Other Methods

In this section, the the proposed algorithm is compared to other methods using different test strategies with synthetically generated data. 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 are generated uniformly within a box defined by $[-5,5] \times [-5,5] \times [-5,5]$ in the object space. A uniformly distributed random 3D rotation is generated as in Section 4.1. For translation, the x and y components are uniformly selected from the interval $[5,15]$ and the z component was selected from the interval $[20,50]$. The set of reference points is then transformed by the randomly selected rotation and translation.

Following this, a fraction (= PO) of the 3D points are selected as outliers. Each of these points is replaced by another 3D point whose components are taken from a uniform distribution within a box $[-5,5] \times [-5,5] \times [-5,5]$ in the object space. The rest of the processing is the same as that in Section 4.1.

4.2.1 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 four 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 step). 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 percent in 5 percent step). 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 step of 10).

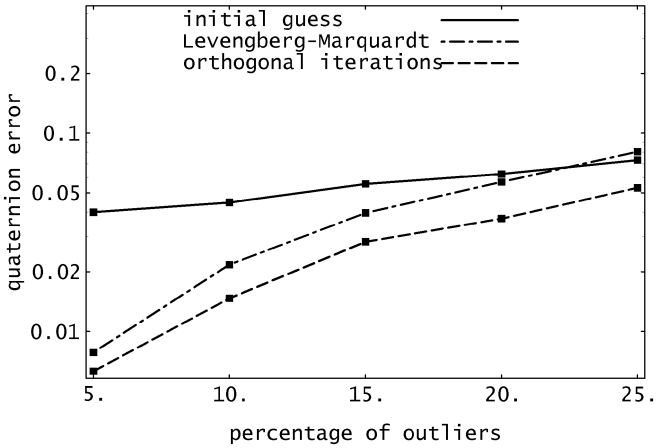

Fig. 13. Result (average rotation errors) of Experiment **C2** for comparing with the Levenberg-Marquardt method. Error is in log scale. Each point in the plot represents 1,000 trials.

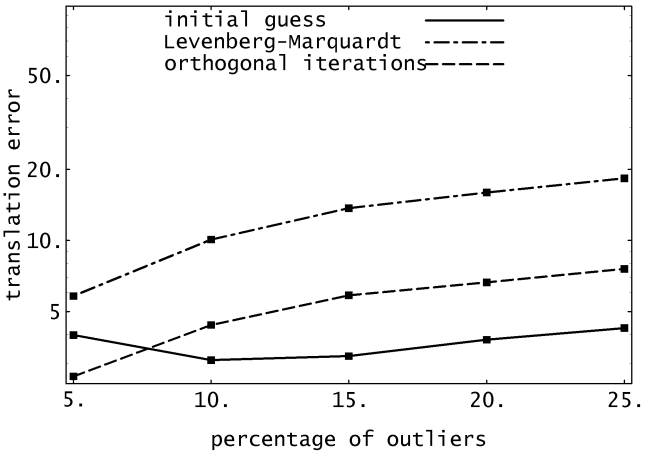

Fig. 14. Result (average translation errors) of Experiment **C2** for comparing with the Levenberg-Marquardt method. Error is in log scale. Each point in the plot represents 1,000 trials.

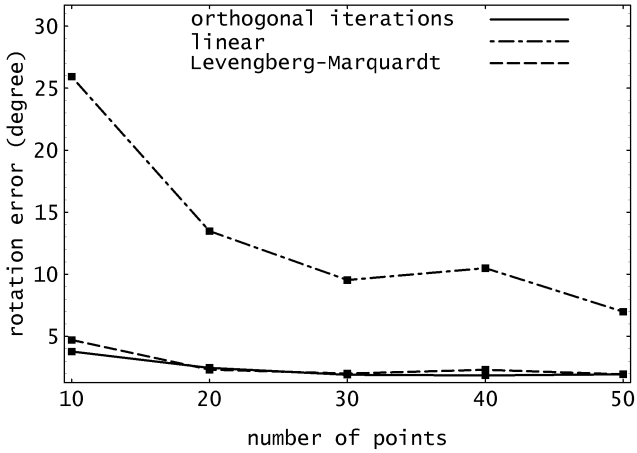

Fig. 15. Result (average rotation errors) of Experiment **C3** for comparing with the Levenberg-Marquardt method. Error is in log scale. Each point in the plot represents 1,000 trials.

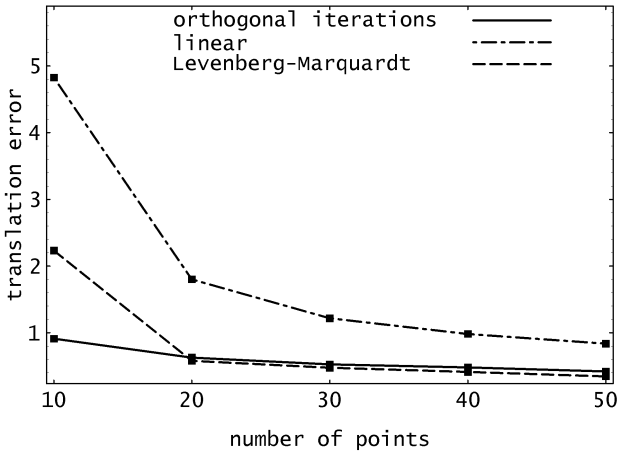

Fig. 16. Result (average translation errors) of Experiment **C3** for comparing with the Levenberg-Marquardt method. Error is in log scale. Each point in the plot represents 1,000 trials.

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 error in rotation and translation over 1,000 trials for each setting of the control parameters.

4.2.2 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. 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.

Figs. 9 and 10 show the average running times and number of iterations of the methods we tested against the number of reference points. These times are measured for $\mathrm{SNR} = 60$ dB and $\mathrm{PO} = 0$ on a Silicon Graphics IRIS Indigo with a MIPS R4400 processor. The orthogonal iteration algorithm is clearly much more efficient than LM, having about the same accuracy as LM without outliers (see Figs. 11, 12, 15, and 16). It significantly outperforms LM in the presence of outliers, as shown by Figs. 13 and 14.

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 is well-suited for any situation,

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

especially where both efficiency 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 [44], making it yet more robust to outliers. In addition, our results suggest that **OI** tends to find the correct pose solution, suggesting that there are few, if any, spurious local minima. We are currently working to determine the conditions under which the pose computed by **OI** is unique and the error of **OI** can be analytically determined.

We are currently implementing a version of the algorithm within the XVision [45] environment for use in robotic applications, as well as augmented and virtual reality. An initial implementation described in [46] has shown that, by combining efficient local tracking with efficient pose estimation, it is relatively simple to construct a real-time object tracking system which runs 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].

Appendix A

Uniqueness of the Optimal Solution to the Absolute Orientation Problem

We show that the best rotation R to (10) is unique. Let

$$M = U\Sigma V^t = \sigma_1 \hat{u}_1 \hat{v}_1^t + \sigma_2 \hat{u}_2 \hat{v}_2^t + \sigma_3 \hat{u}_3 \hat{v}_3^t \tag{47}$$

be an 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 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 an 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 $VU^t = \hat{v}_1\hat{u}_1^t + \hat{v}_2\hat{u}_2^t + \hat{v}_3\hat{u}_3^t$.

Appendix 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 $\mathcal{SO}(3)$, U and V are orthonormal matrices. Likewise, the set of diagonal matrices in $\mathcal{GL}(3)$ is a closed subgroup and, hence, Σ is a diagonal matrix. Therefore, (U, Σ, V) is an 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 an SVD of M.

References

[1] W. Wilson, "Visual Servo Control of Robots Using Kalman Filter Estimates of Robot Pose Relative to Work-Pieces," *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," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* 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," *Proc. ACM SIGGRAPH,* pp. 429-438, 1996.

[4] R. Azuma and G. Bishop, "Improving Static and Dynamic Registration in an Optical See-Through HMD," *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," *Proc. SIGGRAPH,* pp. 203-210, July 1992.

[6] W.E.L. Grimson, *Object Recognition by Computer.* Cambridge, Mass.: 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," *Comm. 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 Vision, Graphics, and Image Processing,* 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," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* pp. 592-598, 1991.

[11] D. DeMenthon and L.S. Davis, "Exact and Approximate Solutions of the Perspective-Three-Point Problem," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 14, no. 11, pp. 1,100-1,105, Nov. 1992.

[12] M. Dhome, M. Richetin, J. Lapresté, and G. Rives, "Determination of the Attitude of 3D Objects from a Single Perspective View," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 11, no. 12, pp. 1,265-1,278, Dec. 1989.

[13] G.H. Rosenfield, "The Problem of Exterior Orientation in Photogrammetry," *Photogrammetric Eng.,* 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, Mass.: Addison-Wesley, 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," Technical Report 641, Univ. of Rochester, 1996

[18] Y.I. Abdel-Aziz and H.M. Karara, "Direct Linear Transformation into Object Space Coordinates in Close-Range Photogrammetry," *Proc. Symp. Close-Range Photogrammetry,* 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," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* pp. 15-20, June 1986.
[21] R.Y. Tsai, "An Effecient and Accurate Camera Calibration Technique for 3D Machine Vision," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* 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 3D Machine Vision Metrology," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 10, no. 3, pp. 713-720, Mar. 1988.
[23] D. DeMenthon and L. Davis, "Model-Based Object Pose in 25 Lines of Code," *Int'l J. Computer Vision,* vol. 15, pp. 123-141, June 1995.
[24] T.D. Alter, "3D Pose from Corresponding Points under Weak-Perspective Projection," Techical Report A.I. Memo No. 1,378, MIT Artificial Intelligence Lab., 1992.
[25] D.P. Huttenlocher and S. Ullman, "Recognizing Solid Objects by Alignment with an Image," *Int'l 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," Techical Report 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. Optical Soc. Am.,* vol. 5, pp. 1,127-1,135, 1988.
[28] R.M. Harlalick et al., "Pose Estimation from Corresponding Point Data," *IEEE Trans. Systems, Man, and Cybernetics,* vol. 19, no. 6, pp. 1,426-1,446, 1989.
[29] R.M. Haralick and L.G. Shapiro, *Computer and Robot Vision,* chapter 14, p. 132. Reading, Mass.: Addison-Wesley, 1993.
[30] D.G. Lowe, "Fitting Parametrized Three-Dimensional Models to Images," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 13, no. 5, pp. 441-450, May 1991.
[31] J. Weng, N. Ahuja, and T.S. Huang, "Optimal Motion and Structure Estimation," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 15, no. 9, pp. 864-884, Sept. 1993.
[32] G.D. Hager, "Real-Time Feature Tracking and Projective Invariance as a Basis for Hand-Eye Coordination," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* pp. 533-539, IEEE CS Press, 1994.
[33] S. Wijesoma, D. Wolfe, and R. Richards, "Eye-to-Hand Coordination for Vision-Guided Robot Control Applications," *Int'l J. Robotics Research,* vol. 12, no. 1, pp. 65-78, 1993.
[34] *Robust Vision for Vision-Based Control of Motion.* M. Vincze and G. Hager eds., 1999.
[35] O. Faugeras, *Three-Dimensional Computer Vision.* MIT Press, 1993.
[36] B.K.P. Horn, "Closed-Form Solution of Absolute Orientation Uusing Unit Quaternion," *J. Optical Soc. Am.,* vol. 4, pp. 629-642, 1987.
[37] M.W. Walker, L. Shao, and R.A. Volz, "Estimating 3D Location Parameters Using Dual Number Quaternions," *CVGIP: Image Understanding,* vol. 54, no. 3, pp. 358-367, 1991.
[38] K.S. Arun, T.S. Huang, and S.D. Blostein, "Least-Squares Fitting of Two 3D Point Sets," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 9, pp. 698-700, 1987.
[39] D.G. Luenberger, *Linear and Nonlinear Programming.* second ed. Reading, Mass.: Addison Wesley, 1984.
[40] H.P. Moravec, "Obstacle Avoidance and Navigation in the Real World by a Seeing Robot Rover," PhD thesis, Stanford Univ., 1980.
[41] S.M. Kiang, R.J. Chou, and J.K. Aggarwal, "Triangulation Errors in Stereo Algorithms," *Proc. IEEE Workshop Computer Vision,* pp. 72-78, 1987.
[42] D. Kirk, *Graphics Gems III,* pp. 124-132. Academic Press, 1992.
[43] T.S. Huang and A.N. Netravali, "Motion and Structure from Feature Correspondences: A Review," *IEEE Proc.,* vol. 82, no. 2, pp. 252-268, 1994.
[44] R.H. Byrd and D.A. Pyne, "Convergence of the Iteratively Reweighted Least-Squares Algorithm for Robust Regression," Technical Report No. 313, Dept. of Math. Science, The Johns Hopkins Univ., 1992.
[45] G.D. Hager and K. Toyama, "XVision: A Portable Substrate for Real-Time Vision Applications," *Computer Vision and Image Understanding,* vol. 69, no. 1, Jan. 1998.
[46] C.-P. Lu, "Online Pose Estimation and Model Matching," PhD thesis, Yale Univ., 1995.

Chien-Ping Lu received the BS degree in electrical engineering and the MS degree in computer science from the National Taiwan University in 1985 and 1989, respectively. During 1985 and 1987, he served as an officer in the navy. He received the PhD degree in computer science from Yale University in 1995. After finishing his doctoral research, Dr. Lu joined SGI as a senior software engineer responsible for the design of imaging algorithms and architectures for SGI's high-end desktop graphics subsystems. After three years with SGI, he joined Rendition as a video and graphics architect working on MPEG-2 and 3D graphics acceleration on PC/Windows platform. He joined iBEAM in December 1999 to lead research and development on broadband interactive television technology. He is a member of the IEEE.

Gregory D. Hager received the BA degree in computer science and mathematics from Luther College in 1983 and the MS and PhD degrees in computer science from the University of Pennsylvania in 1985 and 1988, respectively. From 1988 to 1990, he was a Fulbright junior research fellow at the University of Karlsruhe and the Fraunhofer Institute IITB in Karlsruhe Germany. He then joined the Department of Computer Science at Yale University, where he remained until 1999. He is currently, a professor of computer science at The Johns Hopkins University and a faculty member of the Center for Computer Integrated Surgical Systems and Technology. His current research interests include dynamic vision, vision-based control, human-machine interaction and sensor data fusion, and sensor planning. Dr. Hager has published more than 100 articles and books in the area of vision and robotics. He is a member of the IEEE Computer Society

Eric Mjolsness earned his AB in physics and mathematics from Washington University and his PhD degree in physics and computer science from the California Institute of Technology. He is a principal computer scientist at the Jet Propulsion Laboratory of the California Institute of Technology, where he supervises the Machine Learning Systems Group. He is also a faculty associate in biology at the California Institute of Technology. His research interests include statistical pattern recognition, computer vision, large-scale nonlinear optimization, and relaxation-based neural networks, with applications to scientific problems, including gene regulation, biological development, and modeling geological processes in the solar system. He served on the faculty of Yale University from 1985 to 1994 and the University of California at San Diego from 1994 to 1997. He is a member of the IEEE and the ACM.