

# Iterative Algorithm for Optimal Fiducials Under Weak Perspective Projection

Alfred M. Bruckstein,<sup>1</sup> Robert J. Holt,<sup>2</sup> Arun N. Netravali<sup>2</sup>

<sup>1</sup> Department of Computer Science, Technion—Israel Institute of Technology, Haifa 32000, Israel

<sup>2</sup> Bell Laboratories, Lucent Technologies, Murray Hill, NJ 07974

Received 9 November 2007; accepted 28 November 2008

**ABSTRACT:** In previous work, we designed space fiducials with the aim of making camera pose determination as noise-insensitive as possible. These fiducials turned out to be sets of points that formed concentric regular polyhedra. Here, we apply an idea of Dementhon and Davis and test and analyze an iterative linear algorithm in conjunction with our optimal fiducials to increase the accuracy of the computed camera pose. We also analyze under what circumstances this iterative algorithm is guaranteed to converge to the correct solution. Comprehensive computer simulations illustrate the behavior of the algorithm and the degree of improvement in pose determination in case of convergence. © 2009 Wiley Periodicals, Inc. *Int J Imaging Syst Technol*, 19, 27–36, 2009; Published online in Wiley InterScience (www.interscience.wiley.com). DOI 10.1002/ima.20175

**Key words:** iterative algorithm; fiducials; weak perspective; computer vision

## I. INTRODUCTION

In Bruckstein et al. (1999), we investigated the problem of deciding where a given fixed number of points in space should be located so that the pose of a camera viewing them from unknown locations can be analyzed with the greatest accuracy. Under the assumption that image points were obtained by weak perspective projection, we found that the optimal point configurations formed concentric regular polyhedra. In the process of drawing this conclusion we used a straightforward matrix inversion based algorithm to calculate the camera pose.

In this article we consider an iterative pose recovery algorithm of Dementhon and Davis (1995) and adapt it to our setting of optimal fiducials to improve pose recovery performance in regions where the weak perspective projection is a poor approximation to the true perspective. This happens when the object is near the camera; in particular when the distance from the object to the camera is less than 20 times the diameter of the object. This iterative algo-

gorithm starts with the weak perspective assumption for pose recovery and then successively improves the estimates of pose by using the current estimate of the 3D structure to shift the position of feature points in the image plane towards what would be their “correct” weak perspective projection. We note that the algorithm of Dementhon and Davis, (1995) was further analyzed and improved by Horaud et al. (1997), incorporating paraperspective approximation in the iteration process. Also, the Dementhon and Davis algorithm was modified in a different manner by Chang and Tsai, (2002) in developing a technique for determining facial pose and expression.

In this article, we first analyze under what circumstances the algorithm is guaranteed to converge monotonically to the correct solution, and then present simulation results showing that the iterative pose recovery process works very well, even beyond the monotone convergence region.

## II. FORMULATION

For purposes of fiducial design for efficient pose recovery we assume that we can designate where we place several feature points  $\mathbf{P}_i = [X_i Y_i Z_i]^T$  in the environment. Hence, we assume that the coordinates of the  $\mathbf{P}_i$  are known in the world coordinate system. These points may be observed from a camera in any position, and thus the camera coordinate system is related to the world coordinate system by an arbitrary rotation  $\mathbf{R}$  and translation  $\mathbf{T}$ . A point  $\mathbf{P}'_i = [X'_i Y'_i Z'_i]^T$  in the camera system corresponding to  $\mathbf{P}_i$  in the world system thereby satisfies

$$\mathbf{P}'_i = \mathbf{R}\mathbf{P}_i + \mathbf{T}, \quad (1)$$

with

$$\mathbf{R} = \begin{bmatrix} r_1 & r_2 & r_3 \\ r_4 & r_5 & r_6 \\ r_7 & r_8 & r_9 \end{bmatrix} \quad \text{and} \quad \mathbf{T} = \begin{bmatrix} t_1 \\ t_2 \\ t_3 \end{bmatrix}.$$

We let  $\mathbf{p}'_i = [x'_i y'_i]^T$  denote the corresponding image point in the image plane of the camera. Under true perspective viewing with a camera having a focal length  $f$ , the projection equations are

Correspondence to: Robert J. Holt; e-mail: rholt@qcc.cuny.edu or rjholt@alcatel-lucent.com.

Present address of Robert J. Holt: Department of Mathematics and Computer Science, Queensborough, City University of New York, Bayside, NY 11364, USA.

Present address of Arun N. Netravali: OmniCapital LLP, Westfield, NJ 07090, USA.

$$x'_i = \frac{fX'_i}{Z'_i} = \frac{f}{[r_7 \ r_8 \ r_9] \cdot \mathbf{P}_i + t_3} ([r_1 \ r_2 \ r_3] \cdot \mathbf{P}_i + t_1)$$

$$y'_i = \frac{fY'_i}{Z'_i} = \frac{f}{[r_7 \ r_8 \ r_9] \cdot \mathbf{P}_i + t_3} ([r_4 \ r_5 \ r_6] \cdot \mathbf{P}_i + t_2),$$

and these equations can be rewritten as follows (see Dementhon and Davis, (1995); Horaud et al., (1997)):

$$(1 + e_i)x'_i = \frac{f}{t_3} X'_i$$

$$(1 + e_i)y'_i = \frac{f}{t_3} Y'_i,$$

where

$$e_i \triangleq \frac{1}{t_3} [r_7 \ r_8 \ r_9] \cdot \mathbf{P}_i.$$

If  $t_3$  displaces the configuration of points far away from the camera, as we assume,  $e_i$  is very small. Setting  $e_i$  to zero, we obtain the weak perspective projection equations. Denote the image points obtained by the weak perspective projection as  $\mathbf{p}''_i = [x''_i \ y''_i]^T$ . Then under the weak perspective projection assumption, which holds when the configuration of points is viewed from a sufficiently large distance, we have

$$x''_i = sX'_i, \quad y''_i = sY'_i \quad (2)$$

for some positive constant  $s$  ( $s = ft_3$ ). Alternatively, we can write

$$\mathbf{p}''_i = \mathbf{S}\mathbf{P}'_i \triangleq \begin{bmatrix} s & 0 & 0 \\ 0 & s & 0 \end{bmatrix} \mathbf{P}'_i. \quad (3)$$

The weak perspective image points can be expressed directly in terms of the 3D feature points by combining Eqs. (1) and (3) as

$$\begin{bmatrix} x''_i \\ y''_i \end{bmatrix} = \begin{bmatrix} s & 0 & 0 \\ 0 & s & 0 \end{bmatrix} \begin{bmatrix} r_1 & r_2 & r_3 & t_1 \\ r_4 & r_5 & r_6 & t_2 \\ r_7 & r_8 & r_9 & t_3 \end{bmatrix} \begin{bmatrix} X_i \\ Y_i \\ Z_i \\ 1 \end{bmatrix}$$

or

$$\mathbf{p}''_i = \mathbf{S}[\mathbf{R}|\mathbf{T}] \begin{bmatrix} \mathbf{P}_i \\ \mathbf{1} \end{bmatrix}. \quad (4)$$

If we now denote by  $\mathbf{X}$  the row vector  $[X_1 \dots X_N]$ , and similarly for  $\mathbf{Y}$ ,  $\mathbf{Z}$ ,  $\mathbf{x}'$ ,  $\mathbf{y}'$ , and let  $\mathbf{1} = [1 \dots 1]$ , a vector of  $N$  "1"s, where  $N$  is the number of known points, then Eq. (4) can be readily augmented to

$$\begin{bmatrix} \mathbf{x}'' \\ \mathbf{y}'' \end{bmatrix} = \mathbf{S}[\mathbf{R}|\mathbf{T}] \begin{bmatrix} \mathbf{X} \\ \mathbf{Y} \\ \mathbf{Z} \\ \mathbf{1} \end{bmatrix}. \quad (5)$$

This relation clearly shows that for various rotations and translations, the  $N$ -vectors  $\mathbf{x}'$ ,  $\mathbf{y}'$  all live in a subspace of the  $N$ -dimensional space spanned by the four vectors  $\mathbf{X}$ ,  $\mathbf{Y}$ ,  $\mathbf{Z}$ ,  $\mathbf{1}$ . This observation is the key to the developments of Tomasi and Kanade (1992) and Ullman and Basri (1991), which assume weak perspective observation of a rigid configuration of points in space from different points of view, or from moving cameras. We shall however concentrate in the sequel on the problem of recovering a single rotation  $\mathbf{R}$  and translation  $\mathbf{T}$ , when the vectors  $\mathbf{X}$ ,  $\mathbf{Y}$ ,  $\mathbf{Z}$  are known a priori, from given, single snapshots of the fiducial providing  $(\mathbf{x}', \mathbf{y}')$ .

### III. DETERMINING WEAK PERSPECTIVE IMAGES

In this section, we describe and slightly modify the iterative algorithm of Dementhon and Davis, (1995) that determines where the image points would be if they actually were obtained under weak perspective instead of true perspective projection. Had we the actual weak perspective projections of the points  $X_i$ ,  $Y_i$ , and  $Z_i$ , we would recover the pose as described below (Bruckstein et al. (1999); Dementhon and Davis (1995)).

The computation based on Eq. (5) is somewhat simplified if we consider "normalized coordinates," where each coordinate of the  $\mathbf{P}_i$  is replaced by its difference with the average value of that coordinate for all the points. For example, we define  $\hat{X}_i = X_i - (\sum_{i=1}^N X_i)/N$ , and let  $\hat{\mathbf{X}}$  be the row vector containing all the  $\hat{X}_i$ . Similarly, we define  $\hat{\mathbf{Y}}$ ,  $\hat{\mathbf{Z}}$ ,  $\hat{\mathbf{x}}''$ , and  $\hat{\mathbf{y}}''$ . In the present situation, because we are going to analyze in detail an optimal fiducial, which is a regular tetrahedron, we can, in the object coordinate system, readily choose to have its center at the origin, and its vertices at  $(a,a,a)$ ,  $(a,-a,-a)$ ,  $(-a,a,-a)$ , and  $(-a,-a,a)$ . Therefore, here we have  $\hat{\mathbf{X}} = \mathbf{X}$ ,  $\hat{\mathbf{Y}} = \mathbf{Y}$ ,  $\hat{\mathbf{Z}} = \mathbf{Z}$  (but not  $\hat{\mathbf{x}}'' = \mathbf{x}$ ,  $\hat{\mathbf{y}}'' = \mathbf{y}$ ).

The pose recovery is based on solving Eq. (5). Defining

$$\begin{aligned} \boldsymbol{\lambda} &= [\lambda_1 \ \lambda_2 \ \lambda_3] = [s \ 0 \ 0]\mathbf{R} = [sr_1 \ sr_2 \ sr_3], \\ \boldsymbol{\gamma} &= [\gamma_1 \ \gamma_2 \ \gamma_3] = [0 \ s \ 0]\mathbf{R} = [sr_4 \ sr_5 \ sr_6], \end{aligned}$$

and

$$\mathbf{C} = \begin{bmatrix} \mathbf{X} \\ \mathbf{Y} \\ \mathbf{Z} \end{bmatrix},$$

Equation (5) can be rewritten as follows:

$$\begin{bmatrix} \hat{\mathbf{x}}'' \\ \hat{\mathbf{y}}'' \end{bmatrix} = \begin{bmatrix} \boldsymbol{\lambda} \\ \boldsymbol{\gamma} \end{bmatrix} \mathbf{C}$$

From this, the vectors  $\boldsymbol{\lambda}$  and  $\boldsymbol{\gamma}$  are obtained using the transpose of the pseudo-inverse of  $\mathbf{C}$ , namely  $(\mathbf{C}\mathbf{C}^T)^{-1} \mathbf{C}$ , as follows:

$$\tilde{\boldsymbol{\lambda}}^T = (\mathbf{C}\mathbf{C}^T)^{-1} \mathbf{C}(\hat{\mathbf{x}}'')^T \quad \text{and} \quad \tilde{\boldsymbol{\gamma}}^T = (\mathbf{C}\mathbf{C}^T)^{-1} \mathbf{C}(\hat{\mathbf{y}}'')^T,$$

where  $\tilde{\boldsymbol{\lambda}}$  and  $\tilde{\boldsymbol{\gamma}}$  denote the computed values of  $\boldsymbol{\lambda}$  and  $\boldsymbol{\gamma}$ . Now, we compute the scale  $s$  of the projection as  $[(\|\tilde{\boldsymbol{\lambda}}\|^2 + \|\tilde{\boldsymbol{\gamma}}\|^2)/2]^{1/2}$  followed by Bruckstein et al. (1999) instead of  $(\|\tilde{\boldsymbol{\lambda}}\| + \|\tilde{\boldsymbol{\gamma}}\|)/2$  as in Dementhon and Davis (1995). The former saves one square root computation per iteration, and we found that it is more accurate in some cases.

Because we don't have the actual weak perspective projections, we will seek quantities  $e_i$  that will move the true perspective projections into the weak perspective projection positions, according to  $\mathbf{p}_i'' = (1 + e_i) \mathbf{p}_i'$ . It is here that the iterative Dementhon and Davis procedure begins. For these correction variables  $e_i$ , the initial estimates  $e_i^{(0)}$  are all zero. The iterative algorithm then proceeds into the following loop. Based on the recovered pose via  $\tilde{\lambda}$  and  $\tilde{\gamma}$ , new  $e_i^{(n)}$  are computed as  $[1/(sf)][(\tilde{\lambda} \times \tilde{\gamma}) \cdot \mathbf{P}_i]$ . This works because the exact  $\lambda$  and  $\gamma$  satisfy

$$\begin{aligned} \frac{1}{sf}(\lambda \times \gamma) &= \frac{1}{sf}(s[r_1 \ r_2 \ r_3] \times s[r_4 \ r_5 \ r_6]) \\ &= \frac{s}{f}[r_7 \ r_8 \ r_9] = \frac{1}{t_3}[r_7 \ r_8 \ r_9]. \end{aligned}$$

Let  $\mathbf{p}_i^{(n)} = (x_i^{(n)}, y_i^{(n)})$  denote the  $n$ th approximation to the weak perspective projection of the  $i$ th image point, and  $e_i^{(n)}$  the corresponding relative error, that is, the difference between the true perspective and weak perspective image point locations. Also let  $\hat{\mathbf{p}}_i^{(n)} = (\hat{x}_i^{(n)}, \hat{y}_i^{(n)})$  denote the corresponding normalized coordinates. The image-point vectors are updated as follows:

$$\begin{bmatrix} \hat{\mathbf{x}}^{(n)} \end{bmatrix}^T = \begin{bmatrix} \hat{x}_1^{(n-1)}[1 + e_1^{(n-1)}] \\ \hat{x}_2^{(n-1)}[1 + e_2^{(n-1)}] \\ \hat{x}_3^{(n-1)}[1 + e_3^{(n-1)}] \\ \hat{x}_4^{(n-1)}[1 + e_4^{(n-1)}] \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} \hat{\mathbf{y}}^{(n)} \end{bmatrix}^T = \begin{bmatrix} \hat{y}_1^{(n-1)}[1 + e_1^{(n-1)}] \\ \hat{y}_2^{(n-1)}[1 + e_2^{(n-1)}] \\ \hat{y}_3^{(n-1)}[1 + e_3^{(n-1)}] \\ \hat{y}_4^{(n-1)}[1 + e_4^{(n-1)}] \end{bmatrix}.$$

Note that, we use the origin as the reference point instead of one of the feature points as was done in Dementhon and Davis, (1995). Next we compute the updated vectors  $\hat{\lambda}^T = (\mathbf{C}\mathbf{C}^T)^{-1}\mathbf{C}[\hat{\mathbf{x}}^{(n)}]^T$  and  $\hat{\gamma}^T = (\mathbf{C}\mathbf{C}^T)^{-1}\mathbf{C}[\hat{\mathbf{y}}^{(n)}]^T$ , and  $s$  is updated as before. Then we refine the  $e_i$  estimates based on the updated quantities  $\lambda$ ,  $\gamma$ , and  $s$ .

Initially, the image point coordinates are rounded to the nearest pixel. We continue the iteration process until the sum of the absolute differences of the computed image point coordinates in successive iterations is less than one-tenth of a pixel, and only then round the final results to the nearest pixel. We found that when we tried rounding the image points coordinates at each loop in the iteration as in Dementhon and Davis (1995), on several occasions we reached an infinite loop, with one of the image point coordinates oscillating back and forth between two consecutive integers whereas all the others remained constant.

Typically, the computed vector quantities  $\tilde{\lambda}$  and  $\tilde{\gamma}$  will not be orthogonal and of equal length, especially with noisy data. Consequently, at the end of the iteration process, we adjust  $\tilde{\lambda}$  and  $\tilde{\gamma}$  to the closest orthonormal pair  $\hat{\lambda}$  and  $\hat{\gamma}$ . Formally, we solve the minimization problem:

$$\begin{aligned} \text{Minimize} \quad & \|\hat{\lambda} - \tilde{\lambda}\|^2 + \|\hat{\gamma} - \tilde{\gamma}\|^2 = \sum_{i=1}^3 [(\hat{\lambda}_i - \tilde{\lambda}_i)^2 + (\hat{\gamma}_i - \tilde{\gamma}_i)^2] \\ \text{subject to} \quad & \hat{\lambda}_1 \hat{\gamma}_1 + \hat{\lambda}_2 \hat{\gamma}_2 + \hat{\lambda}_3 \hat{\gamma}_3 = 0 \\ & \hat{\lambda}_1^2 + \hat{\lambda}_2^2 + \hat{\lambda}_3^2 = \hat{\gamma}_1^2 + \hat{\gamma}_2^2 + \hat{\gamma}_3^2. \end{aligned}$$

In Bruckstein et al. (1999), it was shown that the solution to this minimization is given by

$$\begin{aligned} \hat{\lambda} &= \left[ \frac{\|\tilde{\lambda}\| + \|\tilde{\gamma}\|}{2\|\tilde{\lambda}\|} + \frac{\|\tilde{\gamma}\|(\tilde{\lambda} \cdot \tilde{\gamma})^2}{2\|\tilde{\lambda}\|d_2} \right] \tilde{\lambda} - \frac{\tilde{\lambda} \cdot \tilde{\gamma}}{2d_1} \tilde{\gamma} \\ \hat{\gamma} &= -\frac{\tilde{\lambda} \cdot \tilde{\gamma}}{2d_1} \tilde{\lambda} + \left[ \frac{\|\tilde{\lambda}\| + \|\tilde{\gamma}\|}{2\|\tilde{\gamma}\|} + \frac{\|\tilde{\lambda}\|(\tilde{\lambda} \cdot \tilde{\gamma})^2}{2\|\tilde{\gamma}\|d_2} \right] \tilde{\gamma} \end{aligned}$$

where

$$\begin{aligned} d_1 &= \sqrt{\|\tilde{\lambda}\|^2 \|\tilde{\gamma}\|^2 - (\tilde{\lambda} \cdot \tilde{\gamma})^2} \\ d_2 &= \|\tilde{\lambda}\|^2 \|\tilde{\gamma}\|^2 + \|\tilde{\lambda}\| \|\tilde{\gamma}\| d_1 - (\tilde{\lambda} \cdot \tilde{\gamma})^2. \end{aligned}$$

From this we obtain the scale factor, which is the length of  $\hat{\lambda}$  and  $\hat{\gamma}$ ,

$$s = \frac{\sqrt{\|\tilde{\lambda}\|^2 + \|\tilde{\gamma}\|^2 + 2\sqrt{\|\tilde{\lambda}\|^2 \|\tilde{\gamma}\|^2 - (\tilde{\lambda} \cdot \tilde{\gamma})^2}}}{2}.$$

Thus, we obtain the first two rows of the rotation  $\mathbf{R}$  by dividing  $\hat{\lambda}$  and  $\hat{\gamma}$  by  $s$ , and the third row by taking the cross product of the first two. The translation is then

$$\mathbf{T} = \frac{1}{s} \left[ \frac{1}{N} \sum_{i=1}^N x_i'' \quad \frac{1}{N} \sum_{i=1}^N y_i'' \quad f \right]^T,$$

using the last computed values of  $\mathbf{x}''$  and  $\mathbf{y}''$ . These expressions are those on which the numerical results obtained in Section 5 are obtained.

#### IV. CONVERGENCE OF ITERATIVE METHOD FOR OPTIMAL FIDUCIALS

In this section, we analyze conditions under which the iterative algorithm is guaranteed to converge to the correct solution. That is, we start with a set of points in the image plane that are true perspective projections of the vertices of a regular tetrahedron, and show that the points obtained after successive iterations of the loop converge to the positions they would be if they were obtained by weak perspective projection. We allow arbitrary rotations, unlike in Dementhon and Davis (1995) where the authors only consider rotations about the optic axis.

Let the vertices of a regular tetrahedron (an optimal fiducial according to Bruckstein et al. (1999)) in the world coordinate system be  $(a, a, a)$ ,  $(a, -a, -a)$ ,  $(-a, a, -a)$ , and  $(-a, -a, a)$ , where  $a$  is a positive real number. The rotation and translation are as in Eq. (1), with  $t_3 = D$ , where  $D$  is the distance from the center of the tetrahedron to the focal point of the camera. The ratios  $a/D$ ,  $t_1/D$ , and  $t_2/D$  will be the quantities which analyze whether or not the algorithm will converge.

The image points under true perspective projection are given by Eq. (1) as

$$\begin{aligned} \mathbf{p}'_1 &= \left( \frac{f[(r_1 + r_2 + r_3)a + t_1]}{(r_7 + r_8 + r_9)a + D}, \frac{f[(r_4 + r_5 + r_6)a + t_2]}{(r_7 + r_8 + r_9)a + D} \right) \\ \mathbf{p}'_2 &= \left( \frac{f[(r_1 - r_2 - r_3)a + t_1]}{(r_7 - r_8 - r_9)a + D}, \frac{f[(r_4 - r_5 - r_6)a + t_2]}{(r_7 - r_8 - r_9)a + D} \right) \\ \mathbf{p}'_3 &= \left( \frac{f[(-r_1 + r_2 - r_3)a + t_1]}{(-r_7 + r_8 - r_9)a + D}, \frac{f[(-r_4 + r_5 - r_6)a + t_2]}{(-r_7 + r_8 - r_9)a + D} \right) \\ \mathbf{p}'_4 &= \left( \frac{f[(-r_1 - r_2 + r_3)a + t_1]}{(-r_7 - r_8 + r_9)a + D}, \frac{f[(-r_4 - r_5 + r_6)a + t_2]}{(-r_7 - r_8 + r_9)a + D} \right). \end{aligned}$$

The image points under the weak perspective projection, with the scale factor  $s$  equaling  $f/D$ , are given by Eq. (2) as

$$\begin{aligned} \mathbf{p}_1'' &= \left( \frac{f[(r_1 + r_2 + r_3)a + t_1]}{D}, \frac{f[(r_4 + r_5 + r_6)a + t_2]}{D} \right) \\ \mathbf{p}_2'' &= \left( \frac{f[(r_1 - r_2 - r_3)a + t_1]}{D}, \frac{f[(r_4 - r_5 - r_6)a + t_2]}{D} \right) \\ \mathbf{p}_3'' &= \left( \frac{f[(-r_1 + r_2 - r_3)a + t_1]}{D}, \frac{f[(-r_4 + r_5 - r_6)a + t_2]}{D} \right) \\ \mathbf{p}_4'' &= \left( \frac{f[(-r_1 - r_2 + r_3)a + t_1]}{D}, \frac{f[(-r_4 - r_5 + r_6)a + t_2]}{D} \right) \end{aligned}$$

The idea behind the method in Dementhon and Davis, (1995) is to find quantities  $e_i$  such that  $\mathbf{p}_i'' = (1 + e_i) \mathbf{p}_i'$ . The exact values for the  $e_i$  are

$$\begin{aligned} e_1 &= \frac{(r_7 + r_8 + r_9)a}{D} \\ e_2 &= \frac{(r_7 - r_8 - r_9)a}{D} \\ e_3 &= \frac{(-r_7 + r_8 - r_9)a}{D} \\ e_4 &= \frac{(-r_7 - r_8 + r_9)a}{D}. \end{aligned}$$

There are two main steps in determining the rate of convergence of the  $e_i$  to the correct values. First, we will show that after one iteration, the values of the  $e_i$  differ from the correct values by less than a constant  $K$ , whose value is  $\sim 0.316$ , when  $a \leq D/4$  and  $a + T \leq D/2$ , where  $T$  is the maximum allowed translation in either the  $x$  or  $y$  direction. Second, if at the start of an iteration the sum of the squared errors in the  $e_i$  differ from the correct values by at most  $K^2 C^{2n}$ , then at the end of the iteration the sum of the squares of the errors for the new values of the  $e_i$  differ from the correct values by at most  $K^2 C^{2n+2}$ . For the above delineated region for the parameter values, we show that  $C \leq 0.9$ , and hence the geometric convergence to the true values is ensured.

We will analyze the error  $e_1$  as we proceed through iterations of the algorithm. The other  $e_i$  behave the same way, with the only differences being in the signs of some of the coefficients of the rotation components. After one loop of the algorithm, we find that

$$\begin{aligned} e_1^{(1)} &= \frac{\sqrt{2} a \text{numer}_1^{(1)}}{\sqrt{\text{denom}^{(1)}}} & e_2^{(1)} &= \frac{\sqrt{2} a \text{numer}_2^{(1)}}{\sqrt{\text{denom}^{(1)}}} \\ e_3^{(1)} &= \frac{\sqrt{2} a \text{numer}_3^{(1)}}{\sqrt{\text{denom}^{(1)}}} & e_4^{(1)} &= \frac{\sqrt{2} a \text{numer}_4^{(1)}}{\sqrt{\text{denom}^{(1)}}} \end{aligned} \quad (6)$$

---


$$\begin{bmatrix} \cos \theta \cos \phi & -\sin \theta \cos \psi + \cos \theta \sin \phi \sin \psi & \sin \theta \sin \psi + \cos \theta \sin \phi \cos \psi \\ \sin \theta \cos \phi & \cos \theta \cos \psi + \sin \theta \sin \phi \sin \psi & -\cos \theta \sin \psi + \sin \theta \sin \phi \cos \psi \\ -\sin \phi & \cos \phi \sin \psi & \cos \phi \cos \psi \end{bmatrix},$$


---

and find the maximum value of Eq. (7) when  $a$ ,  $t_1$ , and  $t_2$  are at extreme values, and  $\theta$ ,  $\phi$ , and  $\psi$  range over all possible angles. Fortunately, the search space is reduced by making the observations that Eq. (7) is unchanged when  $(t_1, t_2, \theta, \phi, \psi)$  is replaced by  $(t_1, t_2, \theta, \pi + \phi, \pi - \psi)$ ,  $(t_1, t_2, \theta, \phi, \pi + \psi)$ ,  $(t_1, t_2, \pi + \theta, -\phi, -\psi)$ ,  $(-t_1, -t_2, \pi + \theta, \phi, \psi)$ , or  $(-t_1, -t_2, \pi - \theta, \phi, \pi/2 - \psi)$ . This implies that we only need to search the region  $0 \leq t_1, t_2 \leq T$ ,  $0 \leq \theta$ ,

where

$$\begin{aligned} \text{numer}_1^{(1)} &= [D + (r_7 + r_8 + r_9)a][(r_7 + r_8 + r_9)D \\ &\quad + (r_1 + r_2 + r_3)t_1 + (r_4 + r_5 + r_6)t_2 - a] \\ \text{numer}_2^{(1)} &= [D + (r_7 + r_8 + r_9)a][(r_7 - r_8 - r_9)D \\ &\quad + (r_1 - r_2 - r_3)t_1 + (r_4 - r_5 - r_6)t_2 - a] \\ \text{numer}_3^{(1)} &= [D + (-r_7 + r_8 - r_9)a][(-r_7 + r_8 - r_9)D \\ &\quad + (-r_1 + r_2 - r_3)t_1 + (-r_4 + r_5 - r_6)t_2 - a] \\ \text{numer}_4^{(1)} &= [D + (-r_7 - r_8 + r_9)a][(-r_7 - r_8 + r_9)D \\ &\quad + (-r_1 - r_2 + r_3)t_1 + (-r_4 - r_5 + r_6)t_2 - a] \end{aligned}$$

and

$$\begin{aligned} \text{denom}^{(1)} &= 2D^6 + 12r_7r_8r_9aD^5 + b^{(4)}D^4 + b^{(3)}aD^3 \\ &\quad + b^{(2)}a^2D^2 + b^{(1)}a^3D + b^{(0)}a^4, \end{aligned}$$

where the coefficients  $b^{(j)}$  are quite complicated polynomials in  $a$  and the components of  $\mathbf{R}$  and  $\mathbf{T}$ . These coefficients, along with all the others in this article, were computed through the use of the symbolic manipulation program Maple.

We wish to show that the sum of the four squared errors in the  $e_i^{(1)}$  is bounded by some constant. To this end it can be shown that

$$\begin{aligned} &\left[ e_1^{(1)} - \frac{(r_7 + r_8 + r_9)a}{D} \right]^2 + \left[ e_2^{(1)} - \frac{(r_7 - r_8 - r_9)a}{D} \right]^2 \\ &\quad + \left[ e_3^{(1)} - \frac{(-r_7 + r_8 - r_9)a}{D} \right]^2 + \left[ e_4^{(1)} - \frac{(-r_7 - r_8 + r_9)a}{D} \right]^2 \\ &= \frac{4a^2}{D^2} \left\{ \left[ 4D^6 + 36r_7r_8r_9aD^5 + (2u^{(2)} + b^{(4)})D^4 + (2u^{(1)} + b^{(3)})aD^3 \right. \right. \\ &\quad \left. \left. + (2u^{(0)} + b^{(2)})a^2D^2 + b^{(1)}a^3D + b^{(0)}a^4 \right] \right. \\ &\quad \left. - 2\sqrt{2}D[D^2 + 6r_7r_8r_9 + v^{(0)}]\sqrt{\text{denom}^{(1)}} \right\}, \quad (7) \end{aligned}$$

where the coefficients  $u^{(j)}$  and  $v^{(0)}$  are more complicated polynomials in given  $a$  and the components of  $\mathbf{R}$  and  $\mathbf{T}$ .

When  $[\text{denom}^{(1)}]^{1/2}$  is expanded into a series in descending powers of  $D$ , it becomes  $\sqrt{2}D^3 + 3\sqrt{2}r_7r_8r_9aD^2 + O(D)$ , and consequently the terms of order  $D^6$  and  $D^5$  within the braces in the last line of Eq. (7) vanish. To get a sharp bound on the magnitude of this remaining term, we express the rotation in Euler angle form as

$\phi, \psi \leq \pi$  subject to the previous conditions that  $a \leq D/4$  and  $a + T \leq D/2$ . By taking partial derivatives with respect to the several variables, the maximum was found to occur when  $t_1 = t_2 = T$ ,  $\theta = \psi = \pi/4$ , and  $\phi$  satisfies a large polynomial of degree 12 in  $\{D, a, T\}$  and degree 14 in  $\{\sin \phi, \cos \phi\}$ . This equation has to be solved numerically, and we can construct a table of the maximum values of Eq. (7) as shown in Table I.

**Table I.** Maximum values of Eq. (7)

A	T	$\phi$	max
0.25D	0.25D	0.29023	0.09989
0.25D	0.25D	0.30164	0.06291
0.25D	0.25D	0.31300	0.03528
0.25D	0.25D	0.32395	0.01580
0.25D	0.25D	0.33429	0.00401

The global maximum of the sum of the squared errors subject to the conditions that  $a \leq D/4$  and  $a + T \leq D/2$  occurs when  $a = D/4$  and  $T = D/4$ , and is found to be 0.09989. Therefore, this will be the value of  $K^2$  in the remainder of this section, and  $K = 0.31605$ .

We now assume that at the end of the  $(n + 1)$ <sup>st</sup> iterative loop, the sum of the squared errors of the  $e_i^{(n+1)}$  is less than  $K^2 C^{2n}$ . Then the error in each  $e_i^{(n+1)}$  can be described as  $k_i^{(n+1)} C^n$  where each of the  $k_i^{(n+1)}$  is less than  $K$  in absolute value, and  $\sum_{i=1}^4 [k_i^{(n+1)}]^2 \leq K^2$ , so that

$$\begin{aligned} e_1^{(n+1)} &= \frac{(r_7 + r_8 + r_9)a}{D} + k_1^{(n+1)} C^n \\ e_2^{(n+1)} &= \frac{(r_7 - r_8 - r_9)a}{D} + k_2^{(n+1)} C^n \\ e_3^{(n+1)} &= \frac{(-r_7 + r_8 - r_9)a}{D} + k_3^{(n+1)} C^n \\ e_4^{(n+1)} &= \frac{(-r_7 - r_8 + r_9)a}{D} + k_4^{(n+1)} C^n. \end{aligned} \quad (8)$$

Then at the end of the next iteration, we find that

$$\begin{aligned} e_1^{(n+2)} &= \frac{2\sqrt{2} a \text{numer}_1^{(2)}}{\sqrt{\text{denom}^{(2)}}} & e_2^{(n+2)} &= \frac{2\sqrt{2} a \text{numer}_2^{(2)}}{\sqrt{\text{denom}^{(2)}}} \\ e_3^{(n+2)} &= \frac{2\sqrt{2} a \text{numer}_3^{(2)}}{\sqrt{\text{denom}^{(2)}}} & e_4^{(n+2)} &= \frac{2\sqrt{2} a \text{numer}_4^{(2)}}{\sqrt{\text{denom}^{(2)}}} \end{aligned} \quad (9)$$

where

$$\begin{aligned} \text{numer}_1^{(1)} &= 2B(r_7 + r_8 + r_9)a \\ &\quad + C^n D([D + (r_7 + r_8 + r_9)a][\text{numer}_1^{(3)} + C^n D \text{numer}_1^{(4)}]) \\ \text{numer}_2^{(1)} &= 2B(r_7 - r_8 - r_9)a \\ &\quad + C^n D([D + (r_7 - r_8 - r_9)a][\text{numer}_2^{(3)} + C^n D \text{numer}_2^{(4)}]) \\ \text{numer}_3^{(1)} &= 2B(-r_7 + r_8 - r_9)a \\ &\quad + C^n D([D + (-r_7 + r_8 - r_9)a][\text{numer}_3^{(3)} + C^n D \text{numer}_3^{(4)}]) \\ \text{numer}_4^{(1)} &= 2B(-r_7 - r_8 + r_9)a \\ &\quad + C^n D([D + (-r_7 - r_8 + r_9)a][\text{numer}_4^{(3)} + C^n D \text{numer}_4^{(4)}]) \end{aligned}$$

with

$$\begin{aligned} B &= [D + (r_7 + r_8 + r_9)a][D + (r_7 - r_8 - r_9)a] \\ &\quad \times [D + (-r_7 + r_8 - r_9)a][D + (-r_7 - r_8 + r_9)a] \\ \text{numer}_i^{(3)} &= c_i^{(n+1)} D^2 + c_i^{(n)} aD + c_i^{(n-1)} a^2 \\ \text{numer}_i^{(4)} &= c_i^{(1)} D + c_i^{(0)} a, \end{aligned}$$

and

$$\begin{aligned} \text{denom}^{(2)} &= 32a^2 B^2 + 8C^n aDB(d^{(n+2)} D^3 + d^{(n+1)} aD^2 + d^{(n)} a^2 D \\ &\quad + d^{(n-1)} a^3) + C^{2n} D^2 (d^{(6)} D^6 + d^{(5)} aD^5 + d^{(4)} a^2 D^4 \\ &\quad + d^{(3)} a^3 D^3 + d^{(2)} a^4 D^2 + d^{(1)} aD^5 + d^{(0)} a^6), \end{aligned}$$

where the coefficients  $c_i^{(j)}$  and  $d_i^{(j)}$  are complicated polynomials in the  $k_i$  and the components of  $\mathbf{R}$  and  $\mathbf{T}$ .

Because we show that the sum of the squared errors in the  $e_i^{(n+2)}$  is less than  $K^2 C^{2(n+2)}$ , we will show that the expression

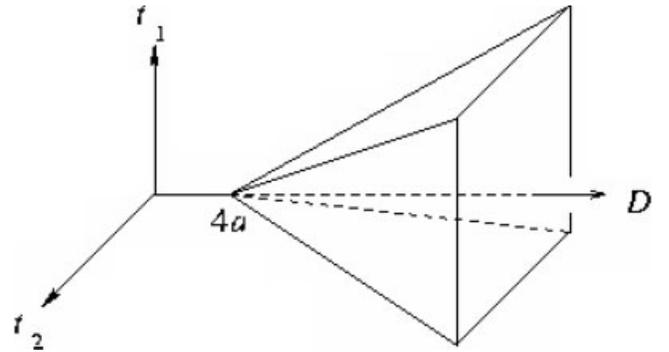
$$\begin{aligned} \frac{1}{K^2 C^{2n+2}} &\left\{ \left[ e_1^{(n+2)} - \frac{(r_7 + r_8 + r_9)a}{D} \right]^2 + \left[ e_2^{(n+2)} - \frac{(r_7 - r_8 - r_9)a}{D} \right]^2 \right. \\ &\quad \left. + \left[ e_3^{(n+2)} - \frac{(-r_7 + r_8 - r_9)a}{D} \right]^2 + \left[ e_4^{(n+2)} - \frac{(-r_7 - r_8 + r_9)a}{D} \right]^2 \right\} \end{aligned}$$

is less than 1. This expression can be shown to equal

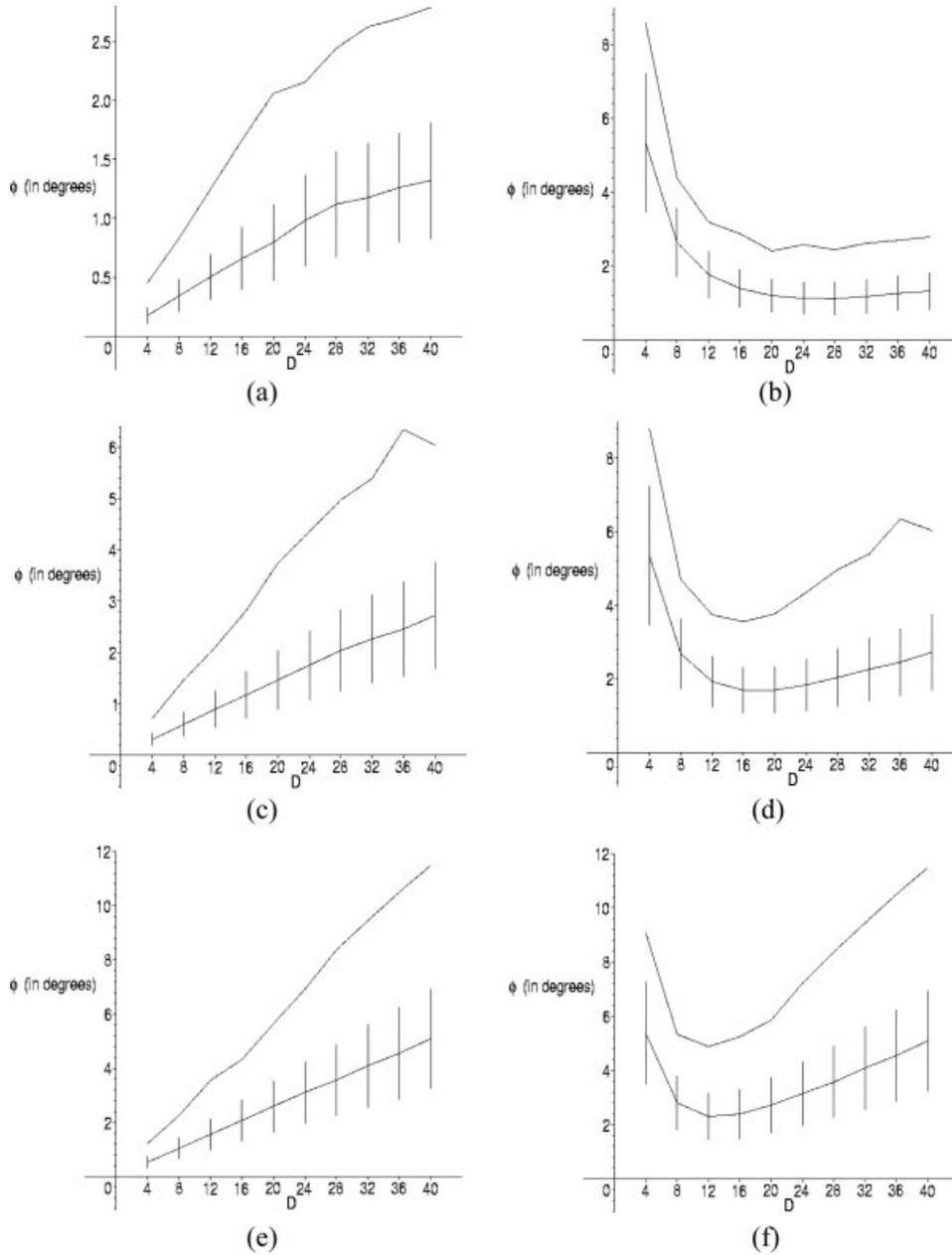
$$\begin{aligned} \frac{4a^2}{K^2 C^{2n+2} D^2 \text{denom}^{(2)}} &\{ 2[16a^2 B^2 \\ &\quad + 4C^n aDB(w^{(n+4)} D^3 + w^{(n+3)} aD^2 + w^{(n+2)} a^2 D + w^{(n+1)} a^3) \\ &\quad + C^{2n} D^2 (w^{(2n+8)} D^6 + w^{(2n+7)} aD^5 + w^{(2n+6)} a^2 D^4 + w^{(2n+5)} a^3 D^3 \\ &\quad + w^{(2n+4)} a^4 D^2 + w^{(2n+3)} a^5 D + w^{(2n+2)} a^6) \\ &\quad + C^{3n} D^3 (w^{(3n+8)} D^5 + w^{(3n+7)} aD^4 + w^{(3n+6)} a^2 D^3 + w^{(3n+5)} a^3 D^2 \\ &\quad + w^{(3n+4)} a^4 D + w^{(3n+3)} a^5) + C^{4n} D^4 (w^{(4n+8)} D^4 + w^{(4n+7)} aD^3 \\ &\quad + w^{(4n+6)} a^2 D^2 + w^{(4n+5)} a^3 D + w^{(4n+4)} a^4) \\ &\quad - \sqrt{2} [8aB + C^n D (w^{(n+4)} D^3 + w^{(n+3)} aD^2 + w^{(n+2)} a^2 D \\ &\quad + w^{(n+1)} a^3) + C^{2n} D^2 (h^{(2n+4)} D^2 + h^{(2n+3)} aD + h^{(2n+2)} a^2)] \\ &\quad \times \sqrt{\text{denom}^{(2)} + \text{denom}^{(2)}} \}, \end{aligned} \quad (10)$$

where the coefficients  $w^{(i)}$  and  $h^{(i)}$  are more complicated polynomials in the  $k_i$  and the components of  $\mathbf{R}$  and  $\mathbf{T}$ .

Note the repetition of the  $w^{(i)}$  as the coefficients of the  $C^n$  term in both terms enclosed in brackets in the last line of Eq. (10). When  $[\text{denom}^{(2)}]^{1/2}$  is expanded into a series in ascending powers of  $C$ , it becomes  $4\sqrt{2} aB + (1/\sqrt{2}) C^n D [d^{(n+2)} D^3 + d^{(n+1)} aD^2 + d^{(n)} a^2 D + d^{(n-1)} a^3] + O(C^{2n})$ , and consequently the terms of order  $C^0$  and  $C^n$  within the braces in Eq. (10) vanish.



**Figure 1.** Pyramidal volume in which our algorithm is guaranteed to converge monotonically.  $\sqrt{3}a$  is the radius of the circumscribing sphere of the regular tetrahedral fiducial,  $D$  is the distance from the center of the tetrahedron to the camera plane, and  $t_1$  and  $t_2$  are the components of the translation of the tetrahedron in the  $x$  and  $y$  directions (orthogonal to the optic axis).



**Figure 2.** Average (with error bars of length  $2\sigma$ ) and maximum errors in the computed direction of the viewing angle as functions of  $D$  for the tetrahedral fiducial: (a–f), with no translation in the  $x$ - and  $y$ -directions; (g–l), with translation in the  $x$ - and  $y$ -directions randomly chosen from  $[-0/4, 0/4]$ . (a), (c), and (e) give the final results for the iterative algorithm, and (b), (d), and (f) show the results after a single iteration. In (a) and (b)  $\varepsilon = 0$ , in (c) and (d)  $\varepsilon = 1$ , and in (e) and (f)  $\varepsilon = 2$ . (g), (i), and (k) give the final results for the iterative algorithm, and (h), (j), and (l) show the results after a single iteration. In (g) and (h)  $\varepsilon = 0$ , in (i) and (j)  $\varepsilon = 1$ , and in (k) and (l)  $\varepsilon = 2$ .

As after the first iteration, we find a bound on the value of Eq. (10) by converting the rotation components into Euler angle form. The search space is reduced by making the observations that Eq. (10) is unchanged when  $(t_1, t_2, \theta, \phi, \psi)$  is replaced by  $(t_1, t_2, \pi + \theta, \pi - \phi, \pi + \psi)$ ,  $(-t_1, -t_2, \pi + \theta, \phi, \psi)$ , or  $(t_1, -t_2, \pi/2 + \theta, \phi, \psi)$ . This implies that we only need to search the region  $0 \leq t_1, t_2 \leq T, 0 \leq \theta, \phi, \psi \leq \pi$  subject to the previous conditions that  $a = D/4$  and  $T = D/4$ . The maximum value was found to occur when  $a = D/4$  and the  $|t_i|$  and  $\sum [k_i^{(n+2)}]^2$  are as large as possible. The

$k_i^{(n+1)}$  were then expressed as  $k_1^{(n+1)} = K \cos \alpha, k_2^{(n+1)} = K \sin \alpha \cos \beta, k_3^{(n+1)} = K \sin \alpha \sin \beta \cos \gamma, k_4^{(n+1)} = K \sin \alpha \sin \beta \sin \gamma$  to enforce the condition  $\sum [k_i^{(n+2)}]^2 = K^2$ , and then the expression within the braces in the last line of Eq. (10), divided by  $\text{denom}^{(2)}$ , was maximized over  $\theta, \phi, \psi, \alpha, \beta$  and  $\gamma$ . This expression is unchanged when  $(\alpha, \beta, \gamma)$  is replaced by  $(\alpha, -\beta, \pi + \gamma)$ ,  $(-\alpha, \pi - \beta, \pi + \gamma)$ , or  $(-\alpha, \pi + \beta, \gamma)$ , so we just need to search the region where  $0 \leq \alpha, \beta, \gamma \leq \pi$ . A place where the expression within the braces in Eq. (10) divided by  $\text{denom}^{(2)}$  is maximized is when

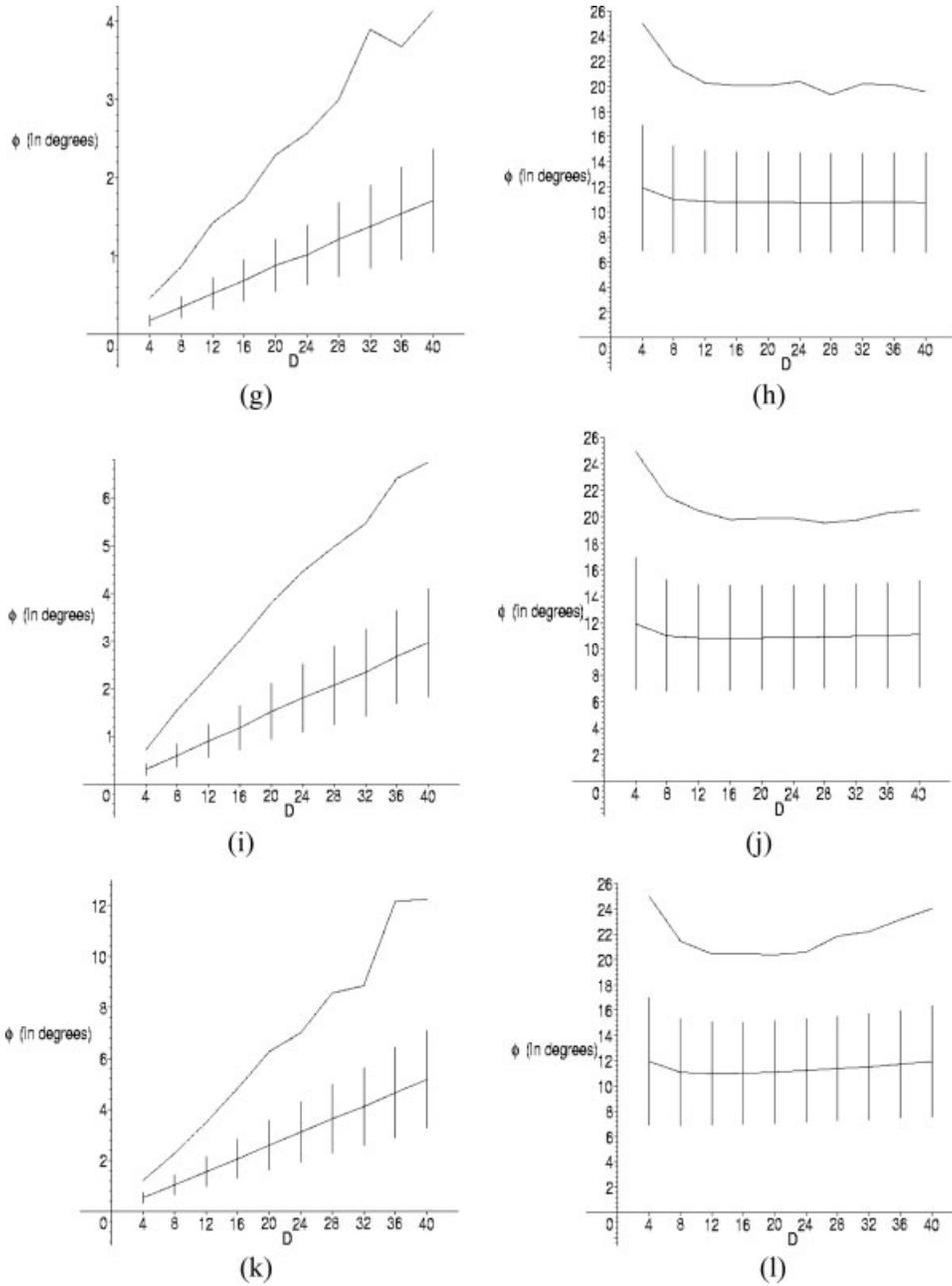


Figure 2. (Continued)

$(t_1, t_2, \alpha, \beta, \gamma, \theta, \phi, \psi) = (1/4, 1/4, 2.90, 1.46, 1.39, 2.30, 2.30, 1.61)$  and that maximum value is 0.221. Taking into consideration the remaining factors  $4a^2/(K^2C^{2n+2}D^2)$  of Eq. (10), the maximum value of Eq. (10) is less than  $[4(1/4)^2/(K^2C^2)] [0.221 < 0.69 < 1]$ . Therefore at the end of the  $(n + 2)^{\text{nd}}$  iteration, the sum of the squared errors of the  $e_i^{(n+2)}$  is less than  $C^2 = 0.81$  times the sum of the squared errors of the  $e_i^{(n+1)}$ .

To summarize, we have proved the following:

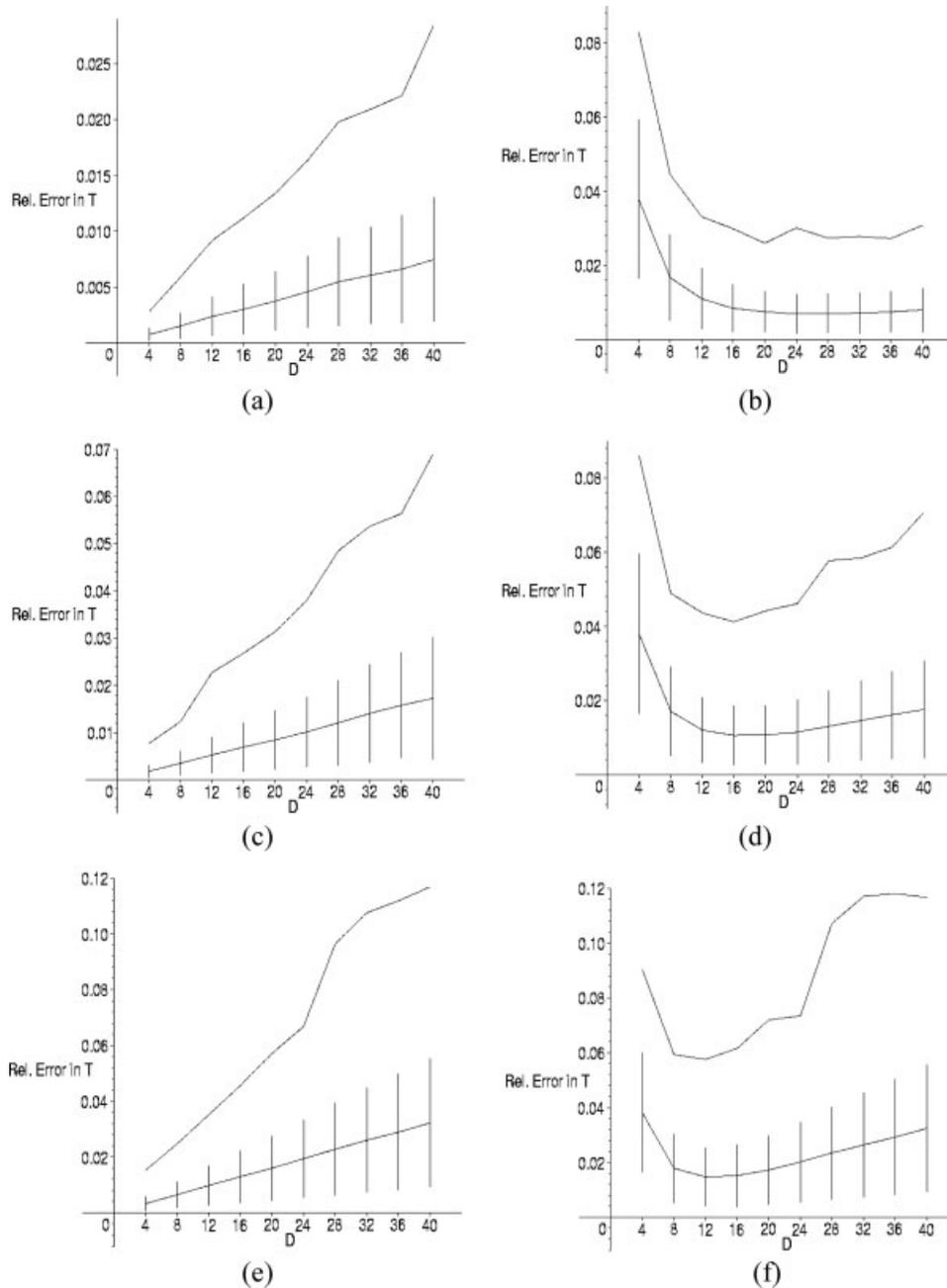
**A. Theorem.** For a regular tetrahedral (optimal) fiducial with circumradius  $\sqrt{3}a$  whose center is at distance  $D$  from the camera plane which may also be translated up to distance  $T$  in the  $x$  and  $y$  direc-

tions (perpendicular to the optic axis), if  $a \leq D/4$  and  $a + T \leq D/2$ , then the iterative pose estimation algorithm described in this article converges monotonically to the correct pose.

For fixed  $a$ , this region in  $Dt_1 t_2$ -space is a pyramid, with vertex at  $(4a, 0, 0)$  and edges whose projections onto the  $Dt_1$ - and  $Dt_2$ -planes are rays with slope  $\pm 1/2$ . This volume is depicted in Figure 1.

## V. NUMERICAL RESULTS

To compare our results fairly with those of Dementhon and Davis (1995), we use similar (distance to camera)/(object size) ratios. In Dementhon and Davis (1995), one feature point, called the



**Figure 3.** Average (with error bars of length  $2\sigma$ ) and maximum errors in the computed direction of the translation as functions of  $D$  for the tetrahedral fiducial: (a–f), with no translation in the  $x$ - and  $y$ -directions; (g–l), with translation in the  $x$ - and  $y$ -directions randomly chosen from  $[-0/4, 0/4]$ . (a), (c), and (e) give the final results for the iterative algorithm, and (b), (d), and (f) show the results after a single iteration. In (a) and (b)  $\varepsilon = 0$ , in (c) and (d)  $\varepsilon = 1$ , and in (e) and (f)  $\varepsilon = 2$ . In all of these there is no translation in the  $x$ - and  $y$ -directions. (g), (i), and (k) give the final results for the iterative algorithm, and (h), (j), and (l) show the results after a single iteration. In (g) and (h)  $\varepsilon = 0$ , in (i) and (j)  $\varepsilon = 1$ , and in (k) and (l)  $\varepsilon = 2$ .

reference point, was taken to be on the optic axis. In their (rectangular) tetrahedron fiducial, the size of the object was taken to be the length of one of the edges. Specifically, the size was equal to the greatest possible distance from any of the other feature points to the reference point. Consequently, we use that measure for determining our object size. For a regular tetrahedron inscribed in the unit sphere ( $a = 1/\sqrt{3}$ ), with the center of the sphere being the reference point, that maximum distance is 1. Thus  $D$  is taken to be the

distance from the center of the tetrahedron to the focal point. The focal length  $F$  is taken to be 760 pixels as in Dementhon and Davis, (1995).

In Figures 2 and 3 the errors in the computed viewing angle and translation are recorded as functions of  $D$ , with noise of  $\varepsilon = 0, 1$  and 2 pixels added, and compared with the results obtained after just one iteration. In Figure 2, the tetrahedron is only rotated about its center, with no translation added, whereas in Figure 3, a

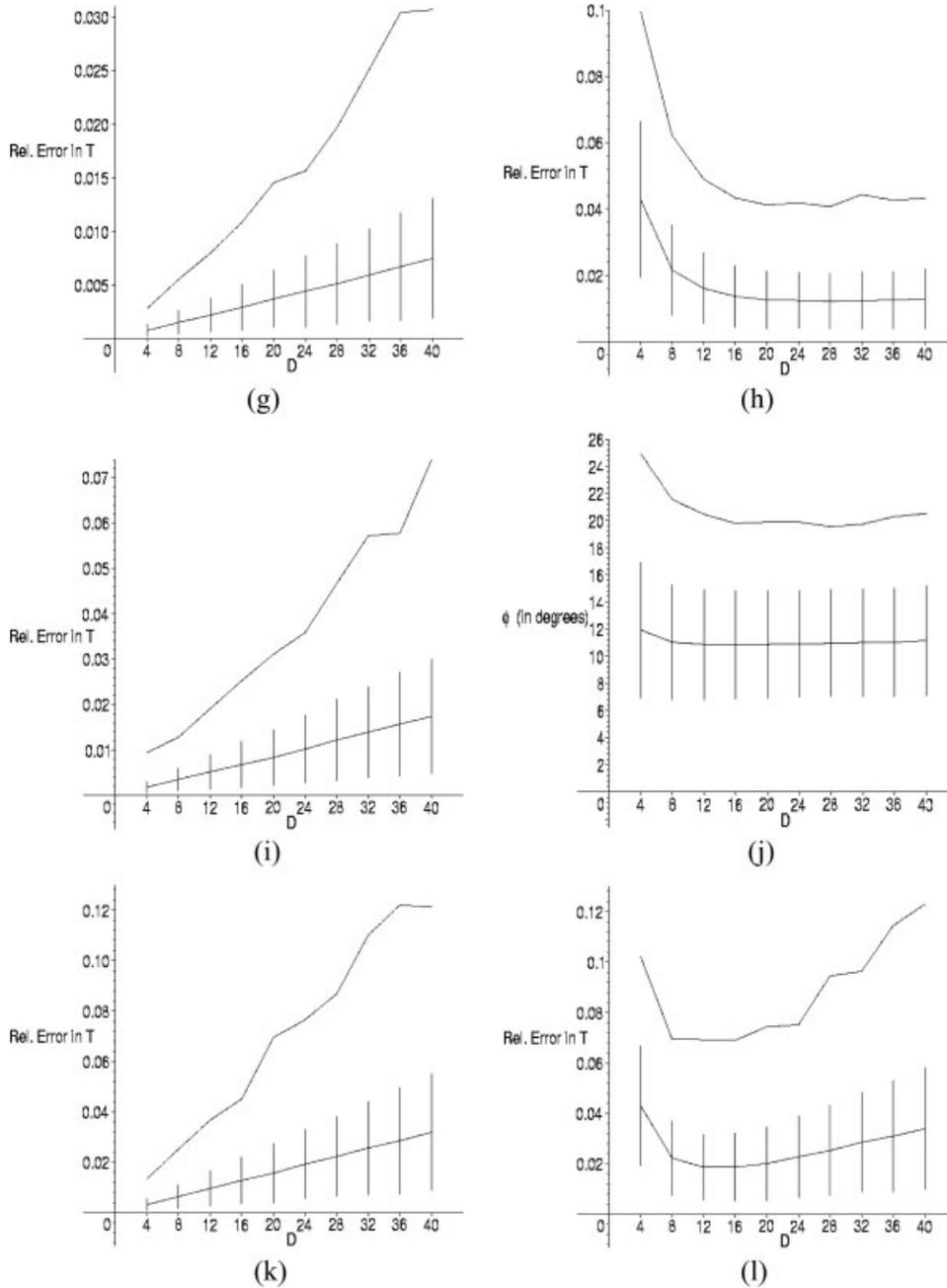


Figure 3. (Continued)

translation chosen randomly and uniformly from  $[-D/4, D/4]$  is added to the  $x$ - and  $y$ -coordinates of each image point.

We now give examples of monotone convergence, and of non-monotone convergence and divergence when the object is too far from the optic axis. First, an example where the algorithm converges monotonically near the edge of the region of monotone convergence occurs when  $a = D/4$ ,  $t_1 = t_2 = D/2$ , and  $(\theta, \phi, \psi) = (\pi/4, 0.28, \pi/4)$ . In this case the sum of the squared errors follows this sequence as shown in Table II

There is a region where the algorithm converges, but not monotonically. For instance, if  $a = D/4$ ,  $t_1 = t_2 = 3D/4$ , and  $(\theta, \phi, \psi) =$

Table II. Position Errors

Iteration	Sum of Squared Errors
1	0.196553
2	0.012368
3	0.000616
4	0.000440
5	0.000064
6	0.000036
7	0.000007
8	0.000002
9	0.000001

**Table III.** Position Errors

Iteration	Sum of Squared Errors
1	0.272823
2	0.027865
3	0.004201
4	0.002723
5	0.002037
6	0.001635
7	0.001179
8	0.000397
9	0.000541
10	0.000179
11	0.000294
12	0.000055
13	0.000131
14	0.000023
15	0.000065
16	0.000009
17	0.000028
18	0.000005
19	0.000013
20	0.000003

**Table IV.** Position Errors

Iteration	Sum of Squared Errors
5	0.014407
10	0.009068
15	0.030425
20	0.023515
25	0.003676
30	0.006113
35	0.014708
40	0.008637
45	0.020369
50	0.033141
55	0.008210
60	0.003401
65	0.012148
70	0.011922
75	0.011623
80	0.034064
85	0.017032
90	0.002752
95	0.008218
100	0.014646

$(\pi/4, 0.28, \pi/4)$ , then the sum of the squared errors follows this sequence as shown in Table III.

As the  $|t_i|$  increase, we reach a region where the algorithm does not converge at all. For example, if  $a = D/4$ ,  $t_1 = t_2 = D$ , and  $(\theta, \phi, \psi) = (\pi/4, 0.28, \pi/4)$ , then the sum of the squared errors follows this sequence with no discernible pattern as shown in Table IV.

Thus, there definitely are limits to the region where the iterative algorithm pose recovery is useful.

## VI. CONCLUSIONS

We have shown that a slightly modified iterative algorithm of Dementhon and Davis, (1995) significantly improves the pose determination performance in conjunction with optimal fiducials. We have analyzed the behavior of the algorithm and showed that in the absence of noise it converges monotonically to the correct solution under certain conditions. These conditions are that the distance of the center of the fiducial, a regular tetrahedron, to the camera plane is at least  $4/\sqrt{3}$  times the radius of the circumscribing sphere of the tetrahedron, and that the sum of that radius and the distance of the center of the tetrahedron to the optic axis is less than half the distance of the center of the tetrahedron to the camera plane.

## REFERENCES

- A.M. Bruckstein, R.J. Holt, T.S. Huang, and A.N. Netravali, Optimum fiducials under weak perspective projection, *Int J Comput Vis* 35 (1999), 223–244.
- C.-C. Chang and W.-H. Tsai, Determination of head pose and facial expression from a single perspective view by successive scaled orthographic approximation, *Int J Comput Vis* 46 (2002), 170–199.
- C. Tomasi and T. Kanade, Shape and motion from image streams under orthography: A factorization method, *Int J Comput Vis* 9 (1992), 137–154.
- D.D. Dementhon and L. Davis, Model-based object pose in 25 lines of code, *Int J Comput Vis* 15 (1995), 123–141.
- R. Horaud, F. Dornaika, B. Lamiroy, and S. Christy, Object pose: The link between weak perspective, paraperspective, and full perspective, *Int J Comput Vis* 22 (1997), 173–189.
- S. Ullman and R. Basri, Recognition by linear combinations of models, *IEEE Trans Pattern Anal Mach Intell* 13 (1991), 992–1005.