POSE ESTIMATION FOR CAMERA CALIBRATION AND LANDMARK TRACKING[1]

M. A. Abidi and T. Chandra

Department of Electrical and Computer Engineering

The University of Tennessee

Knoxville, TN 37996-2100

ABSTRACT

Pose estimation is an important operation for many robotic tasks such as camera calibration and landmark tracking. In this paper, we propose a new algorithm of pose estimation based on the volume measurement of tetrahedra composed of feature-point triplets extracted from an arbitrary quadrangular target and the lens-center of the vision system. This method has been tested using synthetic and real data; it is efficient, accurate, and robust. Its speed, in particular, makes it a potential candidate for real-time robotic tasks.

1 Background

Several researchers have addressed the problem of self-location using standard marks. The central idea of the standard mark approach is as follows. By observing a single projection of a fixed mark, we are able to determine the position and orientation of a camera with respect to a fixed coordinate system. The mark itself is designed such that, when transformed under perspective projection, it yields enough geometric information to recover the relative target position (sometime referred to as interior orientation parameters), the fixed target position (exterior orientation parameters) and final pose (translation and rotation elements of a transformation matrix relating the target frame to the camera frame).

Haralick [1,2] has shown that it is possible to determine the camera parameters from the observed perspective projection of a 3-D rectangle of known size and unknown orientation and position. The author provided a broad review of the properties and uses of the transformation matrix for several computer vision reconstruction problems. He also showed how the orientation of a planar surface can be recovered by computing the perspective projection of vanishing points from a number of parallel lines lying on that surface.

Fischler and Bolles [3] have shown that, knowing the coordinates of a number of 3-D points and their corresponding image points, it is possible to compute the position and orientation of the camera using a geometric closed-form technique. They also described important results on the conditions under which multiple solutions exist for various numbers of correspondences between image and target, particularly for the Perspective-4-Point (P4P) and Perspective-3-Point (P3P) problems. They established that there are up to four solutions in the case of a three-point target. Multiple solutions may exist even in the case of four- or five-point targets when these points are unconstrained in space. A unique solution exists for matching four points of known location which are coplanar and noncollinear. The effect of lens distortion was also addressed.

Eason *et al.* [4] and Abidi *et al.* [5] have formulated the six-, four-, and three-point solutions to this problem. The three-point solution can be recovered by direct means. The four-point solution is also direct for an unconstrained quadrangle. Both the pose parameters and decomposition of the transformation matrix were accomplished simultaneously. No lens distortion was addressed analytically; however, during implementation, the ideal pixel-to-sensor mapping (linear for the pinhole model) was approximated by a cubic for each image coordinate.

Tsai [6] introduced a two-stage technique aimed at efficient computation of camera external position and orientation relative to an object reference coordinate system as well as the effective focal-length, radial lens distortion, and image scanning parameters of an imaging system. This method has a major advantage over many others in that the optimization used to recover the camera pose is linear if the lens distortion was not taken into account. A nonlinear optimization involving only four parameters was required to address the lens distortion problem. The initial guess fed into the nonlinear optimization was given by the linear optimization stage. This method also corrects for abberations caused by the camera system and produces an equivalent focal-length scaled by the distance between adjacent receptor elements. (Part of this work includes, in addition to the Camera Calibration problem [7], a Cartesian Robot-Hand calibration algorithm [8], and a Robot Eye-to-Hand calibration algorithm [9].)

Yuan [10] presented a general solution to the exterior orientation problem. He has shown that this problem can be formulated for an arbitrary number of features. He found a necessary condition for the existence of a solution. He also provided a proof of uniqueness for the case of four coplanar points. Two major drawbacks of this technique are worth noting. First, the proposed solution is iterative. This means that the algorithm may converge to the "wrong" physical solution if no proper initial guess for the initial solution is provided. Second, the unique configuration that generates a unique solution (four coplanar points), which is the only one attractive in practical applications, is not numerically robust under his formulation.

A more detailed summary of papers that relate to the use of standard marks for camera calibration and landmark tracking may be found in [11]. In this paper, we propose a new algorithm for pose estimation based on volume measurement of tetrahedra composed of target points and the lens center of the vision system. Using a pinhole model (lens distortion taken into account separately) and a quadrangular target, for which only the six distance measurements between all pairs of feature points are known, the complete pose is determined using an all-geometric closed-form solution for the six parameters of the pose (three translation components and three rotation components.) A diagram of the complete pose estimation procedure is highlighted in Fig. 1.

2 Recovery of Object Pose

2.1 Interior Orientation Parameters

In this section, we recover the position of the target relative to the camera coordinate system. This is often referred to as interior orientation parameters.

In a pinhole camera model, the image coordinates of a point, (x, y), are related to its camera coordinates, (X^c, Y^c, Z^c), by:

$$x = X^c \frac{f}{f - Z^c} \quad \text{and} \quad y = Y^c \frac{f}{f - Z^c}.$$

[1]This research was partially supported by the Office of Naval Research and Sensitive End-Effector Systems, Inc., Continuation of this work is performed under the DOE's University Program in Robotics for Advanced Reactors (Universities of Florida, Michigan, Tennessee, Texas, and the Oak Ridge National Laboratory) under grant DOE-DE-FG02-86NE37968.

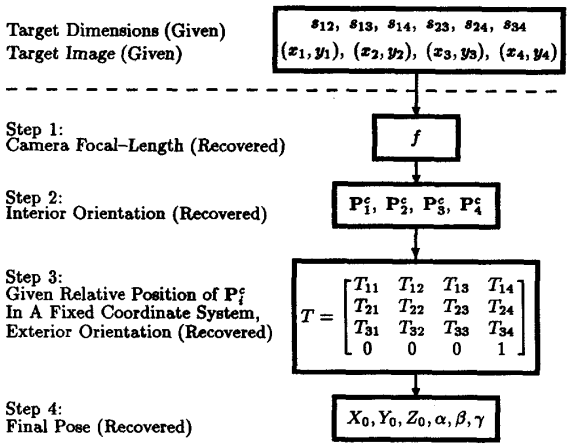

Figure 1: *Various steps using target dimensions and target image to recover the camera focal-length, interior orientation parameters, exterior orientation parameters, and final pose.*

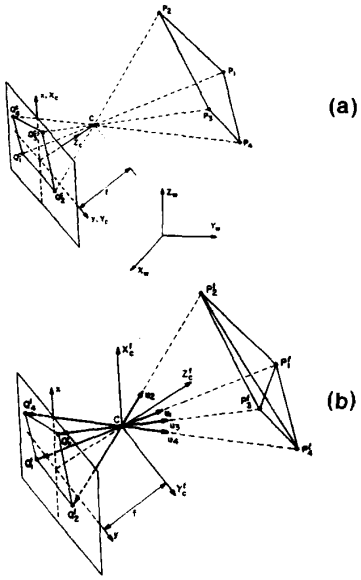

Figure 2: *Illustration of the tetrahedron volume measurement method for pose estimation.*

In practice, this model cannot be realized. Lens distortion often causes image deformation during the imaging process. There are many techniques to correct this problem [12,13,14]. This correction process can be done separately from the pose estimation problem. In addition, this process is needed only once for a given vision system. Brief reviews of these methods will be given in Section 3.

The transformation from world coordinate system to camera coordinate system in a homogeneous space can be represented by a 4 x 4 transformation matrix T:

$$\begin{bmatrix} X^c \\ Y^c \\ Z^c \\ 1 \end{bmatrix} = \begin{bmatrix} T_{11} & T_{12} & T_{13} & T_{14} \\ T_{21} & T_{22} & T_{23} & T_{24} \\ T_{31} & T_{32} & T_{33} & T_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

This matrix, T, can be decomposed into a translation and three rotations:

$$T = R_\alpha R_\beta R_\gamma D\,.$$

The translation, D, is defined by a vector $(-X_0, -Y_0, -Z_0, 1)'$ relating the origin of the world coordinate system and the origin of the camera coordinate system. The rotations, R_α, R_β, and R_γ, are defined by their Euler angles α, β, and γ. (The prime denotes vector or matrix transposition.) Hence the pose of the camera with respect to the world coordinate system can be represented by a pose vector: $\mathcal{P} = (X_0, Y_0, Z_0, \alpha, \beta, \gamma)'$.

The pose estimation problem involves the computation of the elements of the vector $\mathcal{P}$, given a number of world points and their corresponding image points. Here, we present a new analytic solution of a planar quadrangular target based on the measurement of tetrahedra volumes. The viewing geometry of this method is shown in Fig. 2-a.

With reference to this figure, the points of the quadrangular target are labeled P_1, P_2, P_3, and P_4. The vector emanating from the origin of the world coordinate system to the point P_i is labeled $\mathbf{P}_i$. The Euclidean distance between P_i and P_j is denoted by s_{ij}. The vector $(\mathbf{P}_i - \mathbf{P}_j)$ is denoted by $\mathbf{P}_{ij}$, for $i, j = 1, 2, 3$, and 4, $i \neq j$. For this quadrangle, the areas of the following triangles Δ's can be computed as follows:

$$\begin{aligned} A_1 &= \text{Area } \Delta(P_1, P_2, P_3) = |\mathbf{P}_{12} \times \mathbf{P}_{13}|/2 \\ A_2 &= \text{Area } \Delta(P_1, P_2, P_4) = |\mathbf{P}_{12} \times \mathbf{P}_{14}|/2 \\ A_3 &= \text{Area } \Delta(P_1, P_3, P_4) = |\mathbf{P}_{13} \times \mathbf{P}_{14}|/2 \\ A_4 &= \text{Area } \Delta(P_2, P_3, P_4) = |\mathbf{P}_{23} \times \mathbf{P}_{24}|/2\,. \end{aligned}$$

Using Heron's formula [15], these areas can be computed using the following relationships also.

$$\begin{aligned} A_1 &= [(s_{12}^2 + s_{13}^2 + s_{23}^2)^2 - 2(s_{12}^4 + s_{13}^4 + s_{23}^4)]^{1/2}/4 \\ A_2 &= [(s_{12}^2 + s_{14}^2 + s_{24}^2)^2 - 2(s_{12}^4 + s_{14}^4 + s_{24}^4)]^{1/2}/4 \\ A_3 &= [(s_{13}^2 + s_{14}^2 + s_{34}^2)^2 - 2(s_{13}^4 + s_{14}^4 + s_{34}^4)]^{1/2}/4 \\ A_4 &= [(s_{23}^2 + s_{24}^2 + s_{34}^2)^2 - 2(s_{23}^4 + s_{24}^4 + s_{34}^4)]^{1/2}/4\,. \end{aligned}$$

The volume of the following tetrahedra, Γ's, can be computed as follows:

$$\begin{aligned} V_1 &= \text{Volume } \Gamma(C, P_1, P_2, P_3) = hA_1/3 \\ V_2 &= \text{Volume } \Gamma(C, P_1, P_2, P_4) = hA_2/3 \\ V_3 &= \text{Volume } \Gamma(C, P_1, P_3, P_4) = hA_3/3 \\ V_4 &= \text{Volume } \Gamma(C, P_2, P_3, P_4) = hA_4/3\,. \end{aligned}$$

The factor h is the perpendicular distance from the lens center C to the plane containing the quadrangle. It is worth noting that the area measurements required for this experiment are performed only once for a given target.

Figure 2-b shows the vector notations used in this method after translation of the camera system to the center of projection C. With reference to this figure, we denote the vector emanating from the center of projection C to Q_i^f by $\mathbf{Q}_i^f$. The vector joining C to P_i^f is similarly denoted by $\mathbf{P}_i^f$. The unit vector collinear to $\mathbf{Q}_i^f = (x_i, y_i, -f)'$ pointing to the target is denoted by:

$$\mathbf{u}_i = (u_{ix}, u_{iy}, u_{iz})' = -\mathbf{Q}_i^f / ||\mathbf{Q}_i^f|| = (-x_i, -y_i, f)'/F_i\,,$$

where,

$$F_i = \sqrt{x_i^2 + y_i^2 + f^2}\,.$$

Hence, the vector $\mathbf{P}_i^f$ can be expressed as:

$$\mathbf{P}_i^f = d_i \mathbf{u}_i\,,$$

where d_i is the distance between the center of projection C to the point P_i^f, for $i = 1, 2, 3$, and 4. At this point, note that the problem reduces to uniquely determining the focal-length f and the four distances d_1, d_2, d_3, and d_4.

To establish this method, we need to recall some basic properties of the base area and volume measurements for a tetrahedron. The scalar triple product, $\mathbf{a} \cdot (\mathbf{b} \times \mathbf{c})$, of three vectors $\mathbf{a} = (a_x, a_y, a_z)'$, $\mathbf{b} = (b_x, b_y, b_z)'$, and $\mathbf{c} = (c_x, c_y, c_z)^T$ is given by

$$\mathbf{a} \cdot (\mathbf{b} \times \mathbf{c}) = a_x(b_y c_z - b_z c_y) + a_y(b_z c_x - b_x c_z) + a_z(b_x c_y - b_y c_x)\,.$$

Using this scalar triple product, the volume of tetrahedron $\Gamma(C, P_1^f, P_2^f, P_3^f)$ is computed as

$$\begin{aligned} V_1 = A_1 h/3 &= |\mathbf{P}_1^f \cdot (\mathbf{P}_2^f \times \mathbf{P}_3^f)|/6 \\ &= d_1 d_2 d_3 |\mathbf{u}_1 \cdot (\mathbf{u}_2 \times \mathbf{u}_3)|/6\,. \end{aligned}$$

Applying the preceding steps to the three remaining tetrahedra, substituting for $\mathbf{u}_i = (-x_i, -y_i, f)'/F_i$, and solving for h in each equation yields:

$$\begin{aligned} h &= d_1 d_2 d_3 \frac{f}{2F_1F_2F_3}\frac{B_1}{A_1} \\ h &= d_1 d_2 d_4 \frac{f}{2F_1F_2F_4}\frac{B_2}{A_2} \\ h &= d_1 d_3 d_4 \frac{f}{2F_1F_3F_4}\frac{B_3}{A_3} \\ h &= d_2 d_3 d_4 \frac{f}{2F_2F_3F_4}\frac{B_4}{A_4}\,, \end{aligned}$$

where,

$$\begin{aligned} B_1 &= x_1(y_3 - y_2) + y_1(x_2 - x_3) + y_2 x_3 - x_2 y_3 \\ B_2 &= x_1(y_4 - y_2) + y_1(x_2 - x_4) + y_2 x_4 - x_2 y_4 \\ B_3 &= x_1(y_4 - y_3) + y_1(x_3 - x_4) + y_3 x_4 - x_3 y_4 \\ B_4 &= x_2(y_4 - y_3) + y_2(x_3 - x_4) + y_3 x_4 - x_3 y_4\,. \end{aligned}$$

The B_i's are equal to twice the area of the triangle formed by three of the four image points corresponding to the subscripts of the coordinates. Using the four equations involving h, we may express d_2, d_3, and d_4 as a function of d_1:

$$\begin{aligned} d_2 &= \frac{B_3}{A_3}\frac{A_4}{B_4}\frac{F_2}{F_1}d_1 = C_{12}\frac{F_2}{F_1}d_1 \\ d_3 &= \frac{B_2}{A_2}\frac{A_4}{B_4}\frac{F_3}{F_1}d_1 = C_{13}\frac{F_3}{F_1}d_1 \\ d_4 &= \frac{B_1}{A_1}\frac{A_4}{B_4}\frac{F_4}{F_1}d_1 = C_{14}\frac{F_4}{F_1}d_1 \\ d_3 &= \frac{B_2}{A_2}\frac{A_3}{B_3}\frac{F_3}{F_2}d_2 = C_{23}\frac{F_3}{F_2}d_2 \\ d_4 &= \frac{B_1}{A_1}\frac{A_3}{B_3}\frac{F_4}{F_2}d_2 = C_{24}\frac{F_4}{F_2}d_2 \\ d_4 &= \frac{B_1}{A_1}\frac{A_2}{B_2}\frac{F_4}{F_3}d_3 = C_{34}\frac{F_4}{F_3}d_3\,. \end{aligned}$$

(Note the redundancy in these equations; this will be used later to obtain a more accurate solution.) For a complete solution of this system, we need to solve for both f and d_1. The remaining parameters, d_2, d_3, and d_4 are readily computed using the first three of the six previous equations. In order to solve for f, we need to relate it to two (or more) non-redundant measurements from the target. The two additional measurements can be the length of two line segments sharing a point which are part of the quadrangular target. These measurements can be selected from the following 12-element set:

$$\begin{aligned} S = \{&(P_1P_2, P_1P_3), (P_1P_2, P_1P_4), (P_1P_3, P_1P_4), (P_2P_1, P_2P_3), \\ &(P_2P_1, P_2P_4), (P_2P_3, P_2P_4), (P_3P_1, P_3P_2), (P_3P_1, P_3P_4), \\ &(P_3P_2, P_3P_4), (P_4P_1, P_4P_2), (P_4P_1, P_4P_3), (P_4P_2, P_4P_3)\}. \end{aligned}$$

Hence, there are 12 expressions from which we can compute the focal-length, f. They all give the same solution because of the redundancy shown earlier in computing h. This fact is exploited in reducing the effect of statistical random errors by averaging the values obtained for f from each expression or by selecting the median of the 12 possible solutions (see illustration in Section 4). Using the pair of line segments (P_iP_j, P_iP_k), (i, j, and k all different), we compute the squared distances:

$$s_{ij}^2 = (X_j^f - X_i^f)^2 + (Y_j^f - Y_i^f)^2 + (Z_j^f - Z_i^f)^2$$

$$s_{ik}^2 = (X_k^f - X_i^f)^2 + (Y_k^f - Y_i^f)^2 + (Z_k^f - Z_i^f)^2\,.$$

The parametric representation of the line joining C to P_i^f in terms of t is

$$\overline{CP_i}: \begin{cases} X_i^f = u_{ix}\ t = -(x_i/F_i)\ t \\ Y_i^f = u_{iy}\ t = -(y_i/F_i)\ t \\ Z_i^f = u_{iz}\ t = +(f/F_i)\ t \end{cases} \qquad \text{for } i = 1, 2, 3, \text{ and } 4\,.$$

For $t = d_i$, the point on the line coincides with the target point P_i^f. Substituting in s_{ij}^2 and s_{ik}^2, and substituting d_j and d_k by their expressions as a function of d_i, we obtain the following two equations:

$$s_{ij}^2 = d_i^2[(x_i - C_{ij}x_j)^2 + (y_i - C_{ij}y_j)^2 + f^2(1 - C_{ij})^2]/F_i^2$$

$$s_{ik}^2 = d_i^2[(x_i - C_{ik}x_k)^2 + (y_i - C_{ik}y_k)^2 + f^2(1 - C_{ik})^2]/F_i^2\,.$$

(Note that $C_{ij} = 1/C_{ji}$ and $s_{ij} = s_{ji}$.) Dividing the first equation by the second, denoting H_{ij}^2 by

$$H_{ij}^2 = (x_i - C_{ij}x_j)^2 + (y_i - C_{ij}y_j)^2\,,$$

and solving for f yield

$$f = f_{ijk} = \sqrt{\frac{s_{ik}^2 H_{ij}^2 - s_{ij}^2 H_{ik}^2}{s_{ij}^2(1 - C_{ik})^2 - s_{ik}^2(1 - C_{ij})^2}}\,.$$

If the optical axis of the vision system is normal to the target plane, the target and its image are similar, *i.e.*,

$$\frac{r_{12}}{s_{12}} = \frac{r_{13}}{s_{13}} = \frac{r_{14}}{s_{14}} = \frac{r_{23}}{s_{23}} = \frac{r_{24}}{s_{24}} = \frac{r_{34}}{s_{34}} = \frac{f}{Z^c - f}\,,$$

$$\frac{B_1}{A_1} = \frac{B_2}{A_2} = \frac{B_3}{A_3} = \frac{B_4}{A_4}\,,$$

$$Z_1^c = Z_2^c = Z_3^c = Z_4^c = Z^c\,.$$

(The parameter r_{ij} denotes the distance between Q_i^c and Q_j^c.) Under these conditions $C_{ij} = 1$ and $r_{ij}^2 = H_{ij}^2$, which make both numerator and denominator of the expression giving f vanish. Only the ratio $f/(f - Z^c)$ can be computed. Hence, to obtain a unique solution to the pose problem when the target plane and image plane are parallel, f must be given.

If the optical axis of the vision system is not normal to the target plane, the focal-length can be recovered and the interior orientation problem can be recovered as follows. Substituting the

value of f in each of the expressions giving the s_{ij}'s finally yields 6 equivalent expressions for d_1:

$$\begin{aligned} d_1 &= s_{12}F_1[H_{12}^2 + f^2(1-C_{12})^2]^{-1/2} \\ &= s_{13}F_1[H_{13}^2 + f^2(1-C_{13})^2]^{-1/2} \\ &= s_{14}F_1[H_{14}^2 + f^2(1-C_{14})^2]^{-1/2} \\ &= s_{23}F_1[H_{23}^2 + f^2(1-C_{23})^2]^{-1/2}/C_{12} \\ &= s_{24}F_1[H_{24}^2 + f^2(1-C_{24})^2]^{-1/2}/C_{12} \\ &= s_{34}F_1[H_{34}^2 + f^2(1-C_{34})^2]^{-1/2}/C_{13} \,. \end{aligned}$$

Regardless of the shape of the quadrangle $P_1P_2P_3P_4$, all terms $[H_{ij}^2 + f^2(1-C_{ij})^2]$ are nonzero unless two of the target points are coincident, which contradicts our first assumption. If the target measurements and image data are error–free, only one expression for d_1 is necessary.

Substituting back the values of C_{ij}, f, F_i, H_{ij} and $\mathbf{u}_i$ using the two equations $\mathbf{P}_i^f = d_i\mathbf{u}_i$ and $\mathbf{P}_i^c = \mathbf{P}_i^f + (0,0,f)'$, $\mathbf{P}_i^c$ can be expressed solely as a function of s_{ij}, x_i, and y_i, for $i,j = 1,2,3$, and 4, for i, j, and k all different.

$$\begin{aligned} \mathbf{P}_1^c &= (-x_1s_{12}^2E_{123}/G_{123}, -y_1s_{12}^2E_{123}/G_{123}, \\ &\quad s_{12}^2/G_{123} + |s_{13}^2H_{12}^2 - s_{12}^2H_{13}^2|^{1/2}/E_{123})' \\ \mathbf{P}_2^c &= (-x_2s_{23}^2E_{234}/G_{234}, -y_2s_{23}^2E_{234}/G_{234}, \\ &\quad s_{23}^2/G_{234} + |s_{24}^2H_{23}^2 - s_{23}^2H_{24}^2|^{1/2}/E_{234})' \\ \mathbf{P}_3^c &= (-x_3s_{34}^2E_{314}/G_{314}, -y_3s_{34}^2E_{314}/G_{314}, \\ &\quad s_{34}^2/G_{314} + |s_{34}^2H_{31}^2 - s_{31}^2H_{34}^2|^{1/2}/E_{314})' \\ \mathbf{P}_4^c &= (-x_4s_{41}^2E_{412}/G_{412}, -y_4s_{41}^2E_{412}/G_{412}, \\ &\quad s_{41}^2/G_{412} + |s_{42}^2H_{41}^2 - s_{41}^2H_{42}^2|^{1/2}/E_{412})' \,, \end{aligned}$$

where,

$$\begin{aligned} H_{ij}^2 &= (x_i - C_{ij}x_j)^2 + (y_i - C_{ij}y_j)^2 \\ E_{ijk}^2 &= |s_{ij}^2(1-C_{ik})^2 - s_{ik}^2(1-C_{ij})^2| \\ G_{ijk}^2 &= |H_{ij}^2(1-C_{ik})^2 - H_{ik}^2(1-C_{ij})^2| \,. \end{aligned}$$

It is worth noting that the $\mathbf{P}_i^c$'s do not depend explicitly on the focal–length of the vision system. This means that we could have solved for all of them without knowing or determining the focal–length of the camera. As mentioned earlier, there are six different versions for each $\mathbf{P}_i^c$; for brevity, we listed only one of them. For the experimental results presented in Section 4, all six solutions are taken into consideration.

If the target plane is parallel to the image plane, then

$$\begin{aligned} \mathbf{P}_1^c &= (-x_1s_{12}R_{12}^{-1}, -y_1s_{12}R_{12}^{-1}, f[s_{12}R_{12}^{-1} + 1])' \\ \mathbf{P}_2^c &= (-x_2s_{23}R_{23}^{-1}, -y_2s_{23}R_{23}^{-1}, f[s_{23}R_{23}^{-1} + 1])' \\ \mathbf{P}_3^c &= (-x_3s_{34}R_{34}^{-1}, -y_3s_{34}R_{34}^{-1}, f[s_{34}R_{34}^{-1} + 1])' \\ \mathbf{P}_4^c &= (-x_4s_{41}R_{41}^{-1}, -y_4s_{41}R_{41}^{-1}, f[s_{41}R_{41}^{-1} + 1])' \,, \end{aligned}$$

where,

$$R_{ij} = [H_{ij}^2 + f^2(1-C_{ij})^2]^{1/2} \,.$$

For an ideal system based on the pinhole model, of infinite resolution, no lens–distortion, and error–free image processing, the six expressions of $\mathbf{P}_i^c$ yield the same numerical answer. In practice, however, all these factors contribute to some degree in the disparity of the numerical solution. The arithmetic average or median of these values constitute the "best" approximation for $\mathbf{P}_i^c$. This completes the interior orientation parameters estimation problem.

2.2 Exterior Orientation and Final Pose

At this step, we have recovered the relative position of the target with respect to the camera coordinate system. In some circumstances, it is desirable to describe the position of the camera with respect to a fixed coordinate system using the transformation T explicitly. A detailed development of this transformation may be found in [4,11]. Once the exterior orientation parameters are determined, the objective is to recover the pose vector of the camera: $\mathcal{P} = (X_0, Y_0, Z_0, \alpha, \beta, \gamma)'$ using the computed T_{ij}'s. A detailed development of this transformation may be found in [4,11].

3 Lens Distortion Correction

In practice, almost every lens used for image acquisition contains some form and degree of distortion. The image deformation caused by this lens distortion degrades the performance of the pose estimation algorithm developed in this paper because it depends heavily on the exact location of image data. Hence, before implementing the pose estimation algorithm, the deformed image has to be transformed into ideal system using some image correction processes. These processes can be done using least–squares fitting methods with a set of polynomial functions [13,14] or using Bezier patches [12]. The coefficients of these polynomials or the control vertices for the Bezier patches can be determined off–line once for a given vision system. In our Robotic Laboratory, a method to map lens–distorted image coordinates in pixels to the ideal image coordinates has been developed and implemented for various robotic tasks. The pixel–to–sensor coordinates conversion is approximated by a set of polynomials of third order:

$$\begin{aligned} x &= a_0 + a_1I + a_2J + a_3I^2 + a_4J^2 + a_5IJ + \\ &\quad a_6I^3 + a_7J^3 + a_8I^2J + a_9IJ^2 \\ y &= b_0 + b_1I + b_2J + b_3I^2 + b_4J^2 + b_5IJ + \\ &\quad b_6I^3 + b_7J^3 + b_8I^2J + b_9IJ^2 \,. \end{aligned}$$

where (I, J) are the pixel coordinates of a point. The a_i and b_i are determined once for the camera through a least–squares fitting technique using a number of reference points. The values of a_i and b_i are listed below for the camera used in this experiment.

i	a_i	b_i
0	$+2.7514$	-4.5047
1	-1.8716×10^{-2}	$+7.7559 \times 10^{-4}$
2	-2.5888×10^{-4}	$+1.7739 \times 10^{-2}$
3	$+3.6307 \times 10^{-6}$	-1.7485×10^{-6}
4	$+1.2161 \times 10^{-6}$	-2.0289×10^{-6}
5	$+1.0169 \times 10^{-6}$	-2.1701×10^{-6}
6	-5.1562×10^{-9}	$+3.5263 \times 10^{-10}$
7	-5.3668×10^{-11}	$+3.9632 \times 10^{-9}$
8	$+3.2664 \times 10^{-10}$	$+4.6629 \times 10^{-9}$
9	-4.7269×10^{-9}	-3.7592×10^{-9}

This conversion will be used through out this paper to carry out the experiments using real data.

4 Experimental Results

In this section, we present experimental results using real data to illustrate the four–step pose estimation algorithm. (Several experiments using synthetic data were used to evaluate the performance of this algorithm; the resulting interior and exterior parameters as well as pose were all as expected.)

This experiment was conducted using a $T^3 - 726$ Cincinnati Milacron industrial robot having an accuracy of 3 % and a repeatability of 0.1 mm. The vision system consists of a Fairchild 3000 CCD camera (f = 13 mm) and a Perceptics 9200 image processor.

An arbitrary target is selected for this experiment. Its dimensions are given as follows (all dimensions are in millimeters):

$$s_{12} = 077.5 \qquad s_{13} = 177.5 \qquad s_{14} = 162.0$$

$$s_{23} = 160.0 \qquad s_{24} = 191.5 \qquad s_{34} = 104.5\,.$$

Using these dimensions, the areas A_1, A_2, A_3, and A_4 are computed. Recall that these measurements are made only once for a given target. The remaining operations, however, are made every time the pose estimation algorithm is invoked.

The processing of the image yields the pixel coordinates for each target. Applying the conversion method described earlier, we obtain the image, pixel, and sensor coordinates for the four target points:

Target	(I, J)	(x, y)
P_1	(190,101)	(−2.214, −0.936)
P_2	(199,169)	(−2.564, +1.420)
P_3	(061,184)	(+2.442, +1.948)
P_4	(054,095)	(+2.657, +1.143)

Knowing the x_i's and y_i's, the B_i's, C_{ij}'s, F_i's, and H_{ij}'s are computed as shown in Section 2.1. The d_i's are also readily computed by plugging the appropriate parameters: $d_1 = 443.6$, $d_2 = 431.5$, $d_3 = 443.8$, and $d_4 = 462.4$. The relative target position (or interior orientation parameters) is computed by taking the median value of each $\mathbf{P}_i^c$ set:

$$\begin{aligned}
\mathbf{P}_1^c &= (+72.1, +30.5, +436.5)' \\
\mathbf{P}_2^c &= (+80.6, -44.6, +421.5)' \\
\mathbf{P}_3^c &= (-78.8, -62.8, +432.2)' \\
\mathbf{P}_4^c &= (-89.7, +38.6, +452.0)'\,.
\end{aligned}$$

The world–to–camera transformation (or exterior orientation parameters) is computed. The resulting transformation matrix is given by:

$$T = \begin{bmatrix} +0.109 & -0.997 & -0.087 & +072.1 \\ -0.969 & -0.116 & +0.186 & +030.5 \\ -0.194 & +0.066 & -0.979 & +436.5 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

The pose vector is given by:

$$\begin{aligned}
\mathcal{P} &= (X_0, Y_0, Z_0, \alpha, \beta, \gamma)' \\
&= (72.1, 30.5, 436.5, 168.2^o, 115.0^o, 18.9^o)'\,.
\end{aligned}$$

The preceding results are typical over a wide range of target size and distance between target and camera. There is no absolute ground truth against which these results can be compared in order to decide upon the accuracy of the method. We used the robot itself to verify the d_i's. A tool was attached to the robot end–effector. The relative distance between the center of the imaging system and the tip of each target is located by bringing the tool in contact with each target point. A comparative table of the distances computed using the algorithm and those obtained using the relative robot motion is given below.

i	Algorithm	Robot Motion	Disparity
d_1	443.6	443.4	0.2
d_2	431.5	431.3	0.2
d_3	443.8	447.4	3.6
d_4	462.4	468.2	5.8

These results show tentatively that the pose estimation algorithm performs well. Recall, that the method is all plug–in, hence its uniqueness is guaranteed. Since the robot itself has good repeatibility but not as good of an accuracy, more precise evaluation of these results is not possible.

5 Performance Evaluation

5.1 Error Analysis

As to the sensitivity of the this method, we have analytically studied and experimentally tested the relative variation of the error of these results as a function of errors in the 14 inputs (length of six line-segments on the target and coordinates of the four image points). Some samples of the experiments showing the relative errors of d_1, d_2, d_3, and d_4 as a function of the errors of s_{12}, x_1, and y_1 are given in Fig. 3; these were performed for the two poses, $\mathcal{P}_1 = (135, 240, 500, 35°, 165°, 20°)'$ and $\mathcal{P}_2 = (135, 240, 500, 35°, 180°, 20°)'$. The four–point targets are arbitrary chosen in these experiments. These graphs show the solution is well–behaved. These operations were performed for all inputs and all outputs using a large number of poses, all resulting in the same conclusions.

5.2 Performance Improvement of the Algorithm

In the presence of noise due to quantization and other errors, although the averaging process of the new algorithm yields fairly good results, these may differ, to some extent, from the optimum. Exploiting the redundancy in computing d_1, an optimization method is used to further improve these results. For this algorithm, we estimated the errors in the computed pose parameters in terms of errors in the image coordinates. Noise in the image coordinates is assumed to have a zero mean and known variance with the range corresponding to the width of the pixels. For a Fairchild's 3000 CCD camera, which is used as an imaging sensor in our experiment, the width of a pixel of a 256 × 256 image is approximately equal to 0.05 mm. We model the noise with a uniformly–distributed random number. The image coordinates after the noise is added are

$$\begin{aligned}
x_i &= x_i^* + \eta\xi\ , \\
y_i &= y_i^* + \eta\xi\ ,\ i = 1, \ldots, 4\,,
\end{aligned}$$

where η is the level of noise in pixels and ξ a zero–mean uniformly–distributed noise in the range of $[-0.5, +0.5]$. We demonstrate this optimization process using simulation data to see how it can improve the results of the solution for the pose. The Polak–Ribiere's Conjugate–Gradient algorithm [16] is used as an optimization algorithm. The computed image coordinates are used as initial guesses in the optimization process.

Simulation Using Synthetic Data: The location of the targets used in these experiments are given below.

	Target 1	*Target 2*
P_1	(0.0, 0.0, 0.0)	(0.0, 0.0, 0.0)
P_2	(200.0, 0.0, 0.0)	(400.0, 0.0, 0.0)
P_3	(200.0, 150.0, 0.0)	(400.0, 300.0, 0.0)
P_4	(0.0, 150.0, 0.0)	(0.0, 300.0, 0.0)

The effects of increasing the noise level on the solution for the pose are presented in Fig. 4. As we can see from these six graphs, the values of the standard deviations, STD's, increase as the noise level increases. However, with the optimization process, the values of STD's can be reduced to approximately half of those obtained without optimization. From these graphs, we can see also that even with noise level of 2.0 pixels, the errors are still tolerable.

To study the sensitivity of the algorithm to the variations of the target size, we used *Target 1* as the original target; then through varying the four sides of *Target 1* by a size factor from 1 to 2 in increment of 0.1, we computed the STD in % of each of the parameters of the pose with noise level of 0.5 and 1.0 pixels. These results are given in Fig. 5. For the case where the algorithm is run without optimization (dashed curves), the variations of the target size have a significant effect on the accuracy of the algorithm; the errors in the pose increase drastically as the target

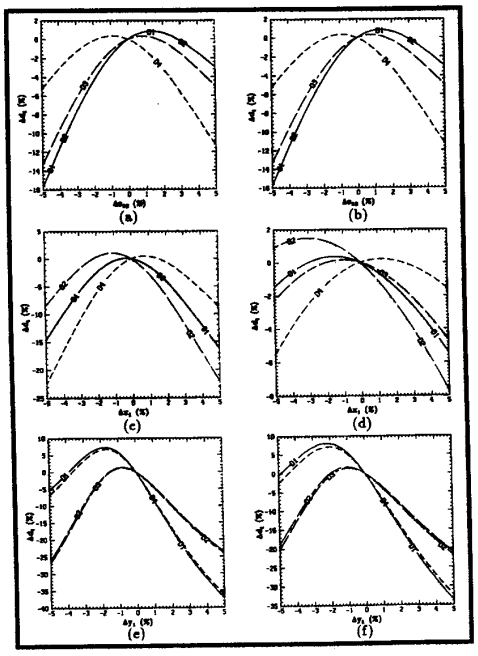

Figure 3: *Sensitivity of the d's due to the perturbation errors in s_{12}, x_1, and y_1. Pose used in (a), (c), and (d) is $\mathcal{P}_1$. Pose used in (b), (d), and (f) is $\mathcal{P}_2$.*

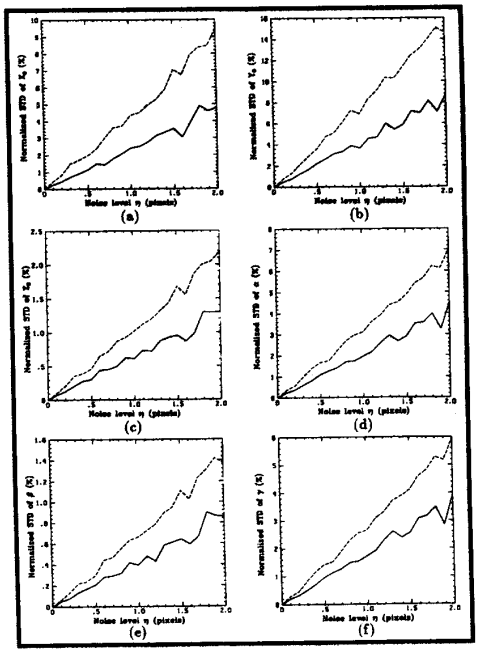

Figure 4: *Improvement of the algorithm performance with optimization method. Dashed curves are the results without optimization. Solid curves are with optimization. Pose used is* $\mathcal{P} = (497, 308, 1000, 126°, 165°, -143°)'$.

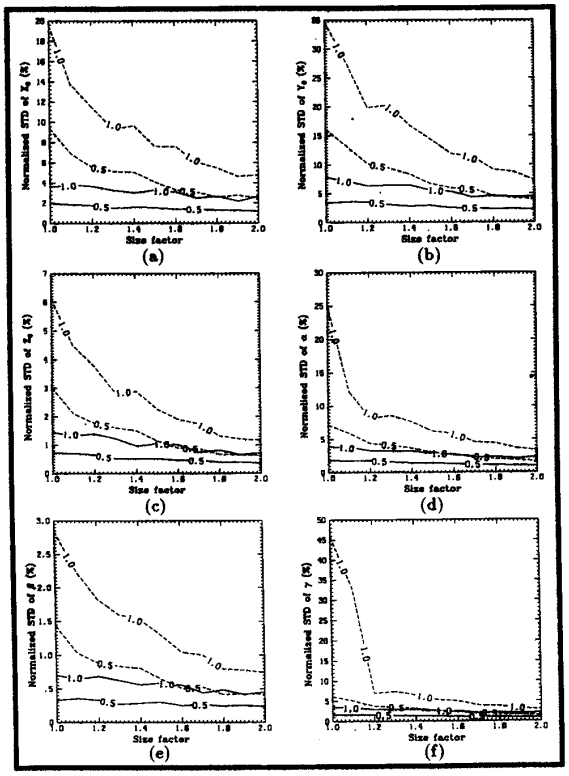

Figure 5: *Improvement of the algorithm performance with optimization method. Noise levels used are 0.5 and 1.0 pixels as indicated in the dashed-pattern of the curves. Pose used is* $\mathcal{P} = (497, 308, 1000, 126°, 165°, -143°)'$.

size decreases. This is not the case for the results obtained with optimization process (solid curves). Hence, the optimization process not only improves the accuracy of the algorithm, but also reduces the sensitivity of the algorithm to the variations of the target size.

Experiments Using Real Data: Here, we examine the discrepancy of the results by measuring their means and standard deviations for each pose of the camera using a target consisting of 10 black dots on a white background. Using a set of coordinates of combinations of four noncollinear centroid points and their corresponding image coordinates, the pose can be solved using the new algorithm. This is performed for a set of 25 target quadruples. In this experiment, the camera was placed at several poses. At each pose, a 256 x 256 image of the target was acquired, thresholded, and the coordinates of the centroid of the dots in pixels were obtained. These pixel coordinates are then converted to the camera coordinate system.

The means and standard deviations of the results with two poses were given below.

		Unoptimized		*Optimized*	
		Mean	*STD*	*Mean*	*STD*
$\mathcal{P}_1$	X_0 *(mm)*	488.1	7.8	497.4	7.6
	Y_0 *(mm)*	323.4	12.7	308.0	8.6
	Z_0 *(mm)*	1409.2	4.5	1410.0	2.8
	α *(degree)*	129.7	1.4	126.9	1.5
	β *(degree)*	164.8	0.5	164.9	0.3
	γ *(degree)*	−140.2	1.4	−142.9	1.5
$\mathcal{P}_2$	X_0 *(mm)*	486.6	29.3	504.9	10.2
	Y_0 *(mm)*	331.9	34.8	304.2	10.8
	Z_0 *(mm)*	1505.3	10.5	1505.4	3.5
	α *(degree)*	130.4	5.3	125.5	1.3
	β *(degree)*	165.3	1.1	165.5	0.5
	γ *(degree)*	−139.5	5.2	−144.2	1.3

In both poses, the STD's in optimized cases are smaller than those in unoptimized cases. Even in the second pose, at which the camera is about 1500 mm from the target, the optimized STD's are within 11 mm for the translation and 1.5^0 for the rotation which can be considered fairly accurate.

6 Summary and Conclusions

In this paper, we described a new technique for tracking a known-size target. The resulting algorithm has unique features compared to previously published other algorithms, particularly those that were designed for tracking purposes. The sequential recovery of the equivalent focal-length, interior orientation parameters, exterior orientation parameters, and final pose is a very attractive feature for many operations. For a number of applications (including the one for which this algorithm was originally conceived), only the interior orientation parameters are necessary because the camera is rigidly fixed to the robot end-effector, with respect to which all motions required by manipulation tasks are performed. Hence, the exterior orientation parameters and final pose need to be computed every time. The algorithm is being implemented as a vision-based tracking system on a mobile platform carrying a six-degree-of-freedom robotic arm to perform manipulation functions at various stations of a manufacturing floor. The sensor used in this system is an off-the-shelf camera equipped with an auto-iris and auto-focus mechanisms, in addition to a variable zoom system servoed to keep the image target within preset dimensions in order to increase the efficiency of the tracking algorithm. Under these circumstances, the recovery of the effective focal-length of the camera is necessary each time the vision system acquires an image in its tracking sequence. Using synthetic and real data, we have shown that the method is accurate.

References

[1] R. M. Haralick, "Using Perspective Transformations in Scene Analysis," *Comp. Vision, Graphics, and Image Processing*, vol. 13, no. 3, pp. 191–221, 1980.

[2] R. M. Haralick, "Determining Camera Parameters from the Perspective Projection of a Rectangle," *Pattern Recognition*, vol. 22, no. 3, pp. 225–230, 1989.

[3] M. A. Fischler and R. C. Bolles, "Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography," *Comm. of ACM*, vol. 24, no. 6, pp. 381–395, June 1981.

[4] R. O. Eason, M. A. Abidi, and R. C. Gonzalez, "A Method for Camera Calibration Using Three World Points," in *Proc. of IEEE Int'l Conf. on Sys., Man, and Cyber.*, (Halifax, Nova Scotia, Canada), pp. 280–289, October 1984.

[5] M. A. Abidi, R. O. Eason, and R. C. Gonzalez, "Camera Calibration In Robot Vision," in *Proc. 4th Scandinavian Conf. Image Analysis*, (Trondheim, Norway), pp. 471–478, June 1985.

[6] R. Y. Tsai, "A Versatile Camera Calibration Technique for High-Accuracy 3D Machine Vision Metrology Using Off-the-Shelf TV Cameras and Lenses," *IEEE J. Robotics and Automation*, vol. RA-3, no. 4, pp. 323–344, August 1987.

[7] S. Ganapathy, "Decomposition of Transformation Matrices for Robot Vision," in *Proc. of Int'l Conf. on Robotics and Automation*, (Rome, Italy), pp. 130–139, November 1984.

[8] R. Lenz and R. Y. Tsai, "Calibrating a Cartesian Robot with Eye-on-Hand Configuration Independent of Eye-to-Hand Relationship," in *Proc. IEEE Computer Vision and Pattern Recognition Conf.*, (Ann Arbor, MI), pp. 67–75, 1988.

[9] R. Y. Tsai and R. K. Lenz, "A New Technique for Fully Autonomous and Efficient 3D Robotics Hand/Eye Calibration," *IEEE Trans. Robotics and Automation*, vol. RA-5, no. 3, pp. 345–358, June 1989.

[10] J. S.-C Yuan, "A General Photogrammic Method for Determining Object Position and Orientation," *IEEE Trans. on Robotics and Automation*, vol. RA-5, no. 2, pp. 129–142, April 1989.

[11] M. A. Abidi and R. C. Gonzalez, "The Use of Multisensor Data In Robotic Inspection and Manipulation Tasks," *IEEE Trans. Robotics and Automation*, vol. RA-6, no. 2, April 1990.

[12] A. Goshtasby, "Correction of Image Deformation from Lens Distortion Using Bezier Patches," *Comp. Vision, Graphics, and Image Processing*, vol. 47, no. 3, pp. 385–394, 1989.

[13] N. Yokobori, P. Yeh, and A. Rosenfeld, "Selective Geometric Corecction of Images," in *Proc. of IEEE Int'l Conf. on Robotics and Automation*, (San Francisco, CA), pp. 448–453, April 1986.

[14] H. Ziemann and S. F. El-Hakim, "System Calibration and Self Calibration Part I: Rotationally Symmetrical Lens Distortion and Image Deformation," *Photogrammetric Engineering and Remote Sensing*, vol. 52, no. 10, pp. 1617–1625, October 1986.

[15] K. Rektorys, (Ed.), *Survey of Applicable Mathematics*. Cambridge, MA: MIT Press, 1969.

[16] E. Polak, *Computational Methods in Optimization*. New York, NY: Academic Press, 1971.