A Self-Calibration Technique for Active Vision Systems

Song De Ma, *Senior Member, IEEE*

Abstract— **Many vision research groups have developed the active vision platform whereby the camera motion can be controlled. A similar setup is the wrist-mounted camera for a robot manipulator. This head-eye (or hand-eye) setup considerably facilitates motion stereo, object tracking, and active perception. One of the important issues in using the active vision system is to determine the camera position and orientation relative to the camera platform. This problem is called the head-eye calibration in active vision, and the hand-eye calibration in robotics.**

In this paper we present a new technique for calibrating the head-eye (or hand-eye) geometry as well as the camera intrinsic parameters. The technique allows camera self-calibration because it requires no reference object and directly uses the images of the environment. Camera self-calibration is important especially in circumstances where the execution of the underlying visual tasks does not permit the use of reference objects. Our method exploits the flexibility of the active vision system, and bases camera calibration on a sequence of specially designed motion. It is shown that if the camera intrinsic parameters are known *a priori*, the orientation of the camera relative to the platform can be solved using 3 pure translational motions. If the intrinsic parameters are unknown, then two sequences of motion, each consisting of three orthogonal translations, are necessary to determine the camera orientation and intrinsic parameters. Once the camera orientation and intrinsic parameters are determined, the position of the camera relative to the platform can be computed from an arbitrary nontranslational motion of the platform. All the computations in our method are linear. Experimental results with real images are presented in this paper.

I. Introduction

Many artificial vision research groups have developed the active vision platform whereby the camera motion can be controlled [1]. A similar setup is the wrist-mounted camera on a robot manipulator. It has been shown [2], [3] that this head-eye (or hand-eye) setup considerably facilitates many visual tasks. One of the important issues in using such a system is to determine the orientation and position of the camera relative to the camera platform (or the robot's end effector). The camera platform's motion is expressed in a coordinate frame attached to the platform's base, whereas the image data are described in the camera coordinate system. Although the platform's motion can be read from the controller, the transformation between the two coordinate systems (called the head-eye geometry) has to be determined in order to compute the camera's motion from the platform's motion. This problem has been elegantly solved by Shiu and Ahmad [4], Tsai and Lenz [5], and Chen [6]. Although the motion representations adopted are different, these approaches are mathematically equivalent. In all these methods, the head-eye geometry is determined by activating a sequence of platform movements and simultaneously measuring the induced camera motion by observing an object of known structure (called the calibration reference). If $\mathbf{A}$ (or $\mathbf{B}$) represents the transformation between the camera frame and a world coordinate system attached to a calibration reference before (or after) the movement, then the induced camera motion is equal to $\mathbf{A}\mathbf{B}^{-1}$ (see Fig. 1), where $\mathbf{A}$ and $\mathbf{B}$ are called the camera extrinsic parameters (or the pose of the camera) and can be determined independently by means of camera calibration using a calibration reference ([7]–[13]). (Note, in this paper, camera calibration means the determination of both intrinsic and extrinsic camera parameters, while the head-eye calibration refers to the determination of the head-eye geometry.) It has been proved [4]–[6] that, for the head-eye calibration, two motions are necessary. Therefore, four calibration processes are needed to determine $\mathbf{A}_1$ and $\mathbf{B}_1$ (after the first motion), and $\mathbf{A}_2$ and $\mathbf{B}_2$ (after the second motion). Besides the computation cost involved, the previous methods also require a reference object which is not always available in practice.

The goal of the current work is to develop a head-eye camera self-calibration technique which determines both the head-eye geometry and the camera intrinsic parameters. The self-calibration technique requires no known reference object and performs calibration directly using the images of the environment. This is important since in many cases, the camera needs to be recalibrated but reference objects cannot easily be placed in the environment (e.g., for the remote control robot). Our method combines the head-eye calibration (the determination of the head-eye geometry) and the camera calibration (the determination of the intrinsic camera parameters) in one process. (Note we do not use the calibration reference which is a known object *in the world coordinate system*. Therefore, the camera extrinsic parameters, i.e., the position and orientation of the camera relative to the *world coordinate system*, are not involved in our method).

Few self-calibration methods have been presented in the literature. Maybank and Faugeras have described such a method for a moving camera [14]. They proved that from two sequential images of a moving camera, one can obtain two quadratic equations (called the Kruppa equations) of the camera intrinsic parameters. Therefore, at least 2 motions are required to

Manuscript received December 31, 1993; revised March 28, 1994. This work was supported by the Chinese National Science Foundation and the Chinese National High Technology Program.

The author is with the National Pattern Recognition Laboratory, Institute of Automation, Chinese Academy of Sciences, Beijing 100080 P.R. China (e-mail: masd@prlsun2.ia.ac.cn).

Publisher Item Identifier S 1042-296X(96)01062-7.

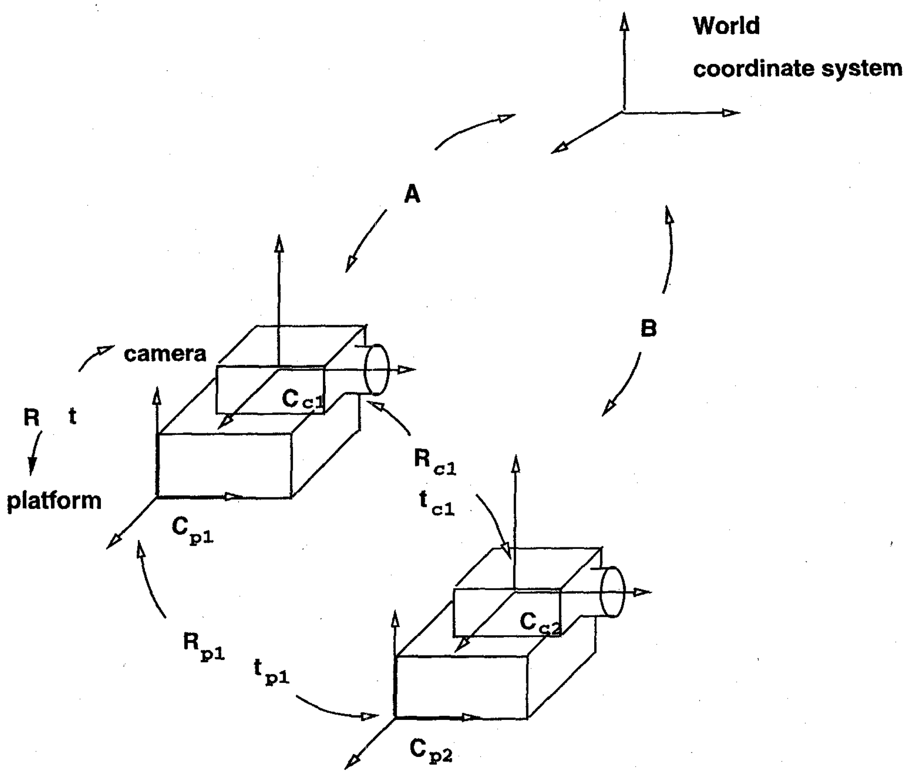

Fig. 1. Platform and camera coordinate systems.

determine 4 camera intrinsic parameters. A system of 4 quadratic equations has to be solved in this method.

Our method exploits the advantage of the active vision system, and performs calibration using a sequence of *specially designed camera motions* which allows linear computation at all stages. This paper is organized as follows. In Section II, we describe the parameters to be determined by our method. Section III presents the algorithm for computing the orientation of the camera from 3 motions of pure translation. We then show, in Section IV, that the camera intrinsic parameters can be obtained from 6 pure translations. In Section V, we discuss the computation of the position of the camera relative to the camera platform. Experiments with real images are presented in Section VI. We conclude the paper in Section VII.

II. Notations and Problem Definition

In an active vision system (or a robot hand-eye system), the camera is rigidly mounted on a pan-tilt-translation platform. The platform can be controlled to move (rotate and/or translate) along any desirable path, and the motion parameters can be read from the controller. Since the camera coordinate system does not coincide with that of the platform (see Fig. 1), the known motion parameters described by the platform's coordinate system are not, in general, identical to those described by the camera frame. The transformation between these two coordinate systems can be described by a rotation $\mathbf{R}$ and a translation $\mathbf{t}$, where $\mathbf{R}$ represents the orientation of the camera, and $\mathbf{t}$ represents the position of the optical center relative to the platform's coordinate system. Therefore, if $\mathbf{x}_c = (x_c, y_c, z_c)^T$ and $\mathbf{x}_p = (x_p, y_p, z_p)^T$ are the coordinates of a point, respectively, described by the camera frame and the platform frame, we have

$$\mathbf{x}_p = \mathbf{R}\mathbf{x}_c + \mathbf{t}. \tag{1}$$

The camera coordinate system is defined such that its origin is the optical center, its x and y axes are parallel, respectively, to the image row and image column, and its z axis is orthogonal to the image plane. Using the pin-hole camera model, we have

$$(u - u_0)dx = \frac{fx_c}{z_c}, \; (v - v_0)dy = \frac{fy_c}{z_c} \tag{2}$$

where (u, v) is the projection of the point $\mathbf{x}_c$ represented by the image pixel coordinates; (u_0, v_0) is the image center, i.e., the intersection of the image plane and the z axis; and dx and dy are, respectively, the sizes of a pixel in x and y directions. By denoting $f_x = f/dx$ and $f_y = f/dy$, (2) can be rewritten as

$$u - u_0 = \frac{f_x x_c}{z_c}, \; v - v_0 = \frac{f_y y_c}{z_c}. \tag{3}$$

We call u_0, v_0, f_x, and f_y the camera intrinsic parameters.

In this paper, we present a head-eye self-calibration technique which allows us to determine both the head-eye geometry ($\mathbf{R}$ and $\mathbf{t}$) and the camera intrinsic parameters u_0, v_0, f_x, and f_y.

III. Camera Orientation from 3 Pure Translations

The basic operation involved in the calibration process is to move the platform. Let C_{c1} and C_{c2} (or C_{p1} and C_{p2}) be, respectively, the camera (or platform) coordinate systems before and after the motion (see Fig. 1). The motion can be represented by $\mathbf{R}_{p1}$ and $\mathbf{t}_{p1}$ using the platform coordinate system. Suppose $\mathbf{x}_{p1}$ and $\mathbf{x}_{p2}$ are, respectively, the coordinates of a point P before and after motion. We have

$$\mathbf{x}_{p1} = \mathbf{R}_{p1}\mathbf{x}_{p2} + \mathbf{t}_{p1}. \tag{4}$$

Substituting (1) into (4) yields

$$\mathbf{R}\mathbf{x}_{c1} + \mathbf{t} = \mathbf{R}_{p1}(\mathbf{R}\mathbf{x}_{c2} + \mathbf{t}) + \mathbf{t}_{p1}. \tag{5}$$

Notice that $\mathbf{x}_{c1}$ and $\mathbf{x}_{c2}$ are the coordinates of the same point P in the camera coordinate system. The motion represented by the camera coordinate system can be described by $\mathbf{R}_{c1}$ and $\mathbf{t}_{c1}$, and from (5), we have

$$\mathbf{R}_{c1} = \mathbf{R}^{-1}\mathbf{R}_{p1}\mathbf{R} \tag{6}$$

$$\mathbf{t}_{c1} = \mathbf{R}^{-1}(\mathbf{R}_{p1}\mathbf{t} + \mathbf{t}_{p1} - \mathbf{t}). \tag{7}$$

Since the camera is mounted rigidly on the platform, the camera motion ($\mathbf{R}_{c1}$, $\mathbf{t}_{c1}$) and the platform motion ($\mathbf{R}_{p1}$, $\mathbf{t}_{p1}$) are in fact the same motion but described by different coordinate frames. We see from (6) and (7) that the two representations of the same motion are related by the head-eye geometry $\mathbf{R}$ and $\mathbf{t}$ which we want to determine in our calibration process.

In the previous work of the head-eye calibration [4]–[6], it is assumed that the platform motion can be obtained from the controller and the induced camera motion can be computed by observing a known reference object in front of the camera. In this case, $\mathbf{R}$ and $\mathbf{t}$ can be solved from (6) and (7). But in our method, no reference object is used and new ways have to be found to compute the induced camera motion.

In order to simplify the method and to use the advantage of the active vision system, we design a sequence of special motions of the platform. From (6), we see that if the motion is a pure translation, then $\mathbf{R}_{c1} = \mathbf{R}_{p1} = \mathbf{I}$, where $\mathbf{I}$ is a 3×3 identity matrix. Hence both the platform motion and the induced camera motion are pure translation. In this case, we have from (7)

$$\mathbf{t}_{p1} = \mathbf{R}\mathbf{t}_{c1}. \tag{8}$$

We will prove that from three noncoplanar translations, the orientation of the camera relative to the platform can be obtained. We have the following proposition.

Proposition 1: If the camera motion is a pure translation, the displacement vectors in the image (i.e., the lines in the image plane obtained by connecting matched points) intersect at a point known as the focus of expansion (FOE) [17], and the vector $\mathbf{O}$ (FOE) connecting the optical center $\mathbf{O}$ of the camera and the point FOE is parallel to the translation.

This proposition is illustrated in Fig. 2. Since the motion between the camera and the scene is relative, the 3-D displacement vectors $\mathbf{d}$ are shown on the object (although in our experiment, the object is stationary and the camera moves in the direction of $-\mathbf{d}$). Since the camera motion is a rigid pure translation, all the displacement vectors $\mathbf{d}$ in 3-D space are parallel. From projective geometry [15], the projections of these vectors in the image intersect at the FOE. In addition, since the line connecting the optical center $\mathbf{O}$ and the FOE passes through the FOE which is the projection of a point at infinity along $\mathbf{d}$, the vector $\mathbf{O}$ (FOE) is parallel to the motion direction.

In this section, we assume that the camera intrinsic parameters are known *a priori*. If not, they can also be determined by our self-calibration technique as shown in the next section.

The operation of our calibration method to determine $\mathbf{R}$ is to control the platform to translate along three noncoplanar directions. Let $\mathbf{t}_{p1}, \mathbf{t}_{p2}$, and $\mathbf{t}_{p3}$ be three normalized translation vectors. Let F_1, F_2, and F_3 be three corresponding FOE detected from the images. From Proposition 1, OF_1, OF_2, and OF_3 are parallel, respectively, to $\mathbf{t}_{p1}, \mathbf{t}_{p2}$, and $\mathbf{t}_{p3}$. Suppose $(u_i, v_i)(i = 1 \sim 3)$ are the image coordinates of $F_i (i = 1 \sim 3)$, then we have

$$\begin{aligned} OF_i &= ((u_i - u_0)dx, (v_i - v_0)dy, f)^T \\ &= f((u_i - u_0)/f_x, (v_i - v_0)/f_y, 1)^T \end{aligned} \tag{9}$$

where $i = 1 \sim 3$.

Since we assume in this section that the intrinsic camera parameters (u_0, v_0, f_x, f_y) are given, then from the coordinates of the three FOE (F_i) and (9), we can compute the direction of OF_i. Let $\mathbf{t}_{ci}(i = 1 \sim 3)$ be the normalized vectors of $OF_i(i = 1 \sim 3)$. Then $\mathbf{t}_{ci}$ and $\mathbf{t}_{pi}$ are the same normalized vectors of the translations but described respectively by the camera frame and the platform frame. From (8), we have $\mathbf{t}_{pi} = \mathbf{R}\mathbf{t}_{ci}(i = 1 \sim 3)$, or in matrix form,

$$(\mathbf{t}_{p1}, \mathbf{t}_{p2}, \mathbf{t}_{p3}) = \mathbf{R}(\mathbf{t}_{c1}, \mathbf{t}_{c2}, \mathbf{t}_{c3}). \tag{10}$$

Since $\mathbf{t}_{p1}, \mathbf{t}_{p2}, \mathbf{t}_{p3}$ are noncoplanar, the matrix $(\mathbf{t}_{c1}, \mathbf{t}_{c2}, \mathbf{t}_{c3})$ is nonsingular, and the rotation matrix $\mathbf{R}$ can be computed from (10)

$$\mathbf{R} = (\mathbf{t}_{p1}, \mathbf{t}_{p2}, \mathbf{t}_{p3})(\mathbf{t}_{c1}, \mathbf{t}_{c2}, \mathbf{t}_{c3})^{-1} \tag{11}$$

where $\mathbf{t}_{pi}(i = 1 \sim 3)$ can be read from the controller, and $\mathbf{t}_{ci}(i = 1 \sim 3)$ are the camera motion computed from the image data F_i.

IV. Camera Intrinsic Parameters from 6 Pure Translations

The camera often needs to be recalibrated since its intrinsic parameters may be changed during the execution of the visual task (e.g., after adjusting the focal length). We present a self-calibration approach by which we can compute the camera intrinsic parameters without using a known reference object. It will be shown that our self-calibration process is almost the same as that for the head-eye geometry calibration described in the previous section. However, since more parameters need to

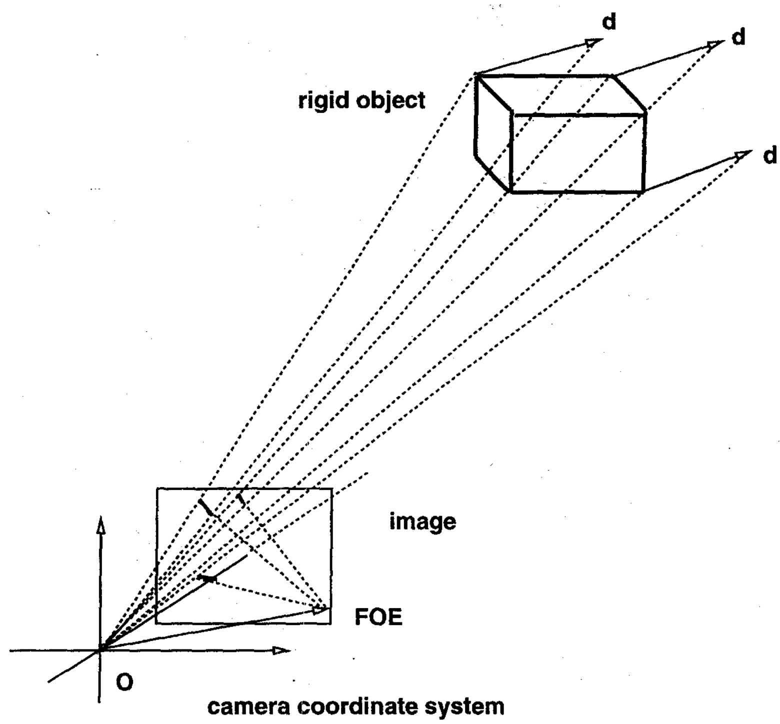

Fig. 2. Focus of expansion of a pure translation.

be determined, more motions are required to obtain all these parameters.

We control the platform to translate along 3 directions. In order to make the computation linear, the three directions $\mathbf{t}_{p1}, \mathbf{t}_{p2}, \mathbf{t}_{p3}$ are set to be orthogonal to each other. As in the previous section, three corresponding FOE's are detected to obtain from the image data three vectors $OF_i(i = 1 \sim 3)$ which are also orthogonal to each other. Then from (9) we obtain three constraints: $(OF_i)^T(OF_j) = 0(i, j = 1 \sim 3, i \neq j)$. We write explicitly these constraints as

$$\frac{1}{f_x^2}(u_1 - u_0)(u_2 - u_0) + \frac{1}{f_y^2}(v_1 - v_0)(v_2 - v_0) + 1 = 0 \quad (12)$$

$$\frac{1}{f_x^2}(u_1 - u_0)(u_3 - u_0) + \frac{1}{f_y^2}(v_1 - v_0)(v_3 - v_0) + 1 = 0 \quad (13)$$

$$\frac{1}{f_x^2}(u_2 - u_0)(u_3 - u_0) + \frac{1}{f_y^2}(v_2 - v_0)(v_3 - v_0) + 1 = 0. \quad (14)$$

Equations (12)–(14) are nonlinear, but by subtracting (13) and (14) from (12), and by denoting $x = u_0, y = \frac{v_0 f_x^2}{f_y^2}, z = \frac{f_x^2}{f_y^2}$, we obtain two linear equations of three variables x, y, z:

$$(u_1 - u_3)x + (v_1 - v_3)y - v_2(v_1 - v_3)z = u_2(u_1 - u_3) \quad (15)$$

$$(u_2 - u_3)x + (v_2 - v_3)y - v_1(v_2 - v_3)z = u_1(u_2 - u_3). \quad (16)$$

In summary, from 3 pure orthogonal translations, we can obtain 2 linear equations of the camera intrinsic parameters. Similar constraints as (15) and (16) have been independently presented by us [11] and by Carple and Torre [13] in using vanishing points for camera calibration. However in the methods presented in [11] and [13], a special known object has to be used as the calibration reference, and on its surface, there should be three sets of parallel lines being orthogonal to each other, whereas in the method of this paper, such known structures are not required.

There are three unknowns in (15) and (16), but only two linear equations are available from three orthogonal translations. Hence, the platform is controlled to move along $\mathbf{t}'_{p1}, \mathbf{t}'_{p2}, \mathbf{t}'_{p3}$ which are also orthogonal to each other but different from the previous directions $\mathbf{t}_{p1}, \mathbf{t}_{p2}, \mathbf{t}_{p3}$ in order to obtain four independent linear equations from which x, y, z can be solved.

After x, y, and z are determined from these linear equations, we can compute all the camera intrinsic parameters. We have $u_0 = x$ and $v_0 = y/z$. By substituting u_0, v_0, and $\frac{f_x^2}{f_y^2} = z$ into (12), we obtain f_x and f_y.

We can combine the two calibration processes described in Sections III and IV. The process of the calibration is as follows.

1) Control the platform to translate along six directions $\mathbf{t}_{p1}, \mathbf{t}_{p2}, \mathbf{t}_{p3}, \mathbf{t}'_{p1}, \mathbf{t}'_{p1}, \mathbf{t}'_{p1}$. The values of these vectors in our experiments are given in Table I (see Section VI).
2) From each pair of the images obtained before and after one motion, compute the FOE to obtain the induced camera translation.
3) From the camera translations, we can obtain 4 linear equations (see (15) and (16)). Then we obtain the camera intrinsic parameters by solving these equations.

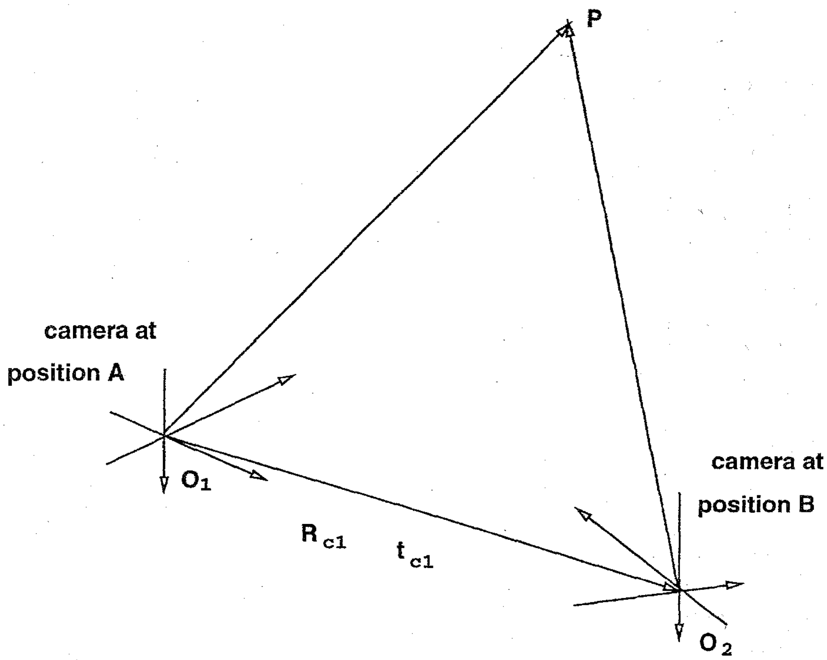

Fig. 3. Computing the induced camera motion.

TABLE I
TWO SETS OF THE PURE TRANSLATION DIRECTIONS

	x	y	z		x	y	z
t_{p1}	0.000	0.447	-0.894	t'_{p1}	0.000	-0.447	-0.894
t_{p2}	0.667	-0.667	-0.333	t'_{p2}	-0.667	0.667	-0.333
t_{p3}	-0.745	-0.596	-0.298	t'_{p3}	0.745	0.596	-0.298

4) From $\mathbf{t}_{p1}, \mathbf{t}_{p2}, \mathbf{t}_{p3}$, the corresponding induced camera translations and the camera intrinsic parameters, compute the camera orientation $\mathbf{R}$ (see (11) in Section III).

V. CAMERA POSITION FROM A GENERAL MOTION

So far we have solved the camera intrinsic parameters and the orientation of the camera. In this section, we show that using these determined parameters, we can obtain the last unknown parameter $\mathbf{t}$, the position of the camera optical center relative to the platform.

From (6) and (7), we see that if the camera motion is a pure translation, $\mathbf{R}_{p1}$ and $\mathbf{R}_{c1}$ are identity matrices, and $\mathbf{t}$ disappears from (6) and (7). This implies that $\mathbf{t}$ cannot be solved from pure translational motion.

Suppose that the motion is not set to be a pure translation, then from (7) we can solve $\mathbf{t}$ by

$$\mathbf{t} = (\mathbf{R}_{p1} - \mathbf{I})^{-1}(\mathbf{R}\mathbf{t}_{c1} - \mathbf{t}_{p1}) \tag{17}$$

where $\mathbf{R}$ has been determined by the method of Section III, and $\mathbf{R}_{p1}$ and $\mathbf{t}_{p1}$ represent the platform's motion known from the controller. Hence, in order to recover $\mathbf{t}$, we should know the induced camera translation $\mathbf{t}_{c1}$. In the previous works presented in [4]–[6], $\mathbf{t}_{c1}$ was obtained by observing a known reference object before and after the motion. Here we present a technique which can be used in cases where no reference object is available as in the previous sections. Our calibration process is as follows.

1) When the platform is in the original position A (see Fig. 3), we reconstruct a point P in the scene by stereo vision (for the robustness of the algorithm, we reconstructed a number of points in our experiments). For simplicity, the reconstruction is performed by translating the platform along the direction of the camera's x axis. This can be done because we know the orientation $\mathbf{R}$ of the camera relative to the platform coordinate system. From (1), a unit vector along the x axis is $(1, 0, 0)^t$ in the camera coordinate system, and is $\mathbf{r}_1$ in the platform coordinate system, where $\mathbf{r}_1$ is the first column of $\mathbf{R}$. So when we control the platform to move along $\mathbf{r}_1$, the induced camera motion is along the camera's x axis. The length of the base line can be read from the controller and the vector O_1P can be reconstructed by the conventional stereo vision method. Once O_1P has been reconstructed, the platform is moved back to its original position A.
2) The platform is then controlled to move to a second position B. The motion parameter $\mathbf{R}_{p1}$ and $\mathbf{t}_{p1}$ can be read from the controller.
3) As in the first step, we reconstruct the same point P in the scene by stereo method. We obtain the vector O_2P represented by the second camera frame at B. From Fig. 3, we see that $O_2P + \mathbf{t}_{c1} = O_1P$. By representing both O_1P and O_2P in the camera coordinate frame at A, we

TABLE II
COMPARISON OF THE TRANSLATIONS READ FROM CONTROLLER AND COMPUTED

	x	y	z	x	y	z
t'_{p1}	0.000	-0.447	-0.894	0.002	-0.439	-0.899
t'_{p2}	-0.667	0.667	-0.333	-0.670	0.665	-0.324
t'_{p3}	0.745	0.596	-0.298	0.742	0.610	-0.278

TABLE III
ACTUAL LENGTHS AND THE COMPUTED LENGTHS OF THE EDGES OF A POLYGON

actual	120	80	40	40	80	120
computed	118.30	80.06	38.74	39.08	79.09	119.09

have

$$\mathbf{R}_{c1}(O_2P) + \mathbf{t}_{c1} = O_1P. \tag{18}$$

4) Using $\mathbf{R}_{c1}$ obtained by (6), O_1P and O_2P, $\mathbf{t}_{c1}$ can be computed by (18).
5) From $\mathbf{t}_{c1}$, we then obtain the position of the camera $\mathbf{t}$ by (17).

VI. EXPERIMENTS

We have verified our method by experiments with real image data. In our experiments, a CCD camera (resolution 380×460) was mounted on the end effector of an IRB 2000 robot manipulator. The manipulator has 6 degrees of freedom. With this robot, we can control its end-effector to move along any direction. The motion parameters can be read from the controller. In principle, we can use the image data of any scene during the calibration. But to simplify the matching of the characteristic points in computing the FOE (see Sections III and IV) and in reconstructing the characteristic points in 3-D world (see Section V), we assume that in the scene there exists a planar polygonal object with n vertexes $(n > 4)$ so that we can use the cross-ratio invariants for vertex matching. From projective geometry [15], for any vertex of a polygon $(n > 4)$, the cross ratio of the four lines, respectively, connecting this vertex and the other four vertexes of the polygon is invariant under any projective transformation. This invariant can be used as the vertex's feature in matching. Once we have matched the points in two images observed before and after motion, FOE can be obtained which is the intersection of the straight lines passing through the matched points. If we have n characteristic points detected in each image, we have n such lines. The intersection point is obtained by the least squares method.

In our experiments, the camera intrinsic parameters obtained by the method in Section IV are $u_0 = 299.50, v_0 = 227.03, f_x = 1325.25, f_y = 2010.26$. We have shown in Section IV that for the calibration we should move the platform along six directions. The six translation vectors described by the platform coordinate system are given in Table I.

We can verify the orientation of the camera computed from the first three orthogonal translations using the second three orthogonal translations $\mathbf{t}'_{p1}, \mathbf{t}'_{p2}$, and $\mathbf{t}'_{p3}$ which can be read from the controller and are shown in the first three columns in Table II. The induced camera motion can be detected from the FOE as described in Section III. Then we transform the motions obtained from the image data to the platform coordinate system using the computed camera orientation $\mathbf{R}$. The computed $\mathbf{t}'_{p1}, \mathbf{t}'_{p2}, \mathbf{t}'_{p3}$ are shown in the last three columns of Table II which should ideally be identical to those read from the controller. Table II shows the comparison.

All the parameters computed (camera orientation and position, intrinsic parameters) have been verified by a stereo vision experiment. We control the end effector from $\hat{A}$ to $\hat{B}$, and take two images. The relative orientation and position between the two views can be computed from the end effector's motion and the head-eye geometry computed by our method. Using these data and the camera intrinsic parameters computed, we reconstruct a polygon placed in front of the camera at a distance of about 1.5 m. Table III shows the actual and the computed lengths of the edges of the polygon. The average reconstruction error is about 1 mm.

VII. CONCLUSION

We have presented a novel camera calibration technique for determining the camera intrinsic parameters and the head-eye geometry. The technique allows camera self-calibration so that a calibration reference is not needed. The camera platform is active and can be controlled to move to any desired orientation and position. The operation of our calibration approach is based on specially designed platform motions so that the camera intrinsic parameters, the orientation of the camera, and the position of the camera can be obtained separately by linear computation.

The computed calibration data has been used in stereo reconstruction and conics-based vision [16]. It has been shown that the results obtained by our method are rather faithful and stable.

ACKNOWLEDGMENT

The author thanks Dr. T. N. Tan of the Intelligent System Research Group at the University of Reading, U.K., for substantial help in the editing of this paper.

REFERENCES

[1] M. J. Swain and M. Stricker, "Promising direction in active vision," Univ. Chicago, Tech. Rep. CS-91-27, Nov. 1991.
[2] R. Bajcsy, "Active perception vs. passive perception," in *Proc. IEEE Workshop Computer Vision: Representation and Control*, Bellaire, MI, Oct. 1985, pp. 55–62.
[3] J. Aloimonos, I. Weiss, and A. Bandyopadhyay, "Active vision," in *Proc. 1st Int. Conf. Computer Vision*, London, U.K., 1987, pp. 35–54.
[4] Y. C. Shiu and S. Ahmad, "Calibration of wrist-mounted robotic sensors by solving homogeneous transform equations of the form $AX = XB$," *IEEE Trans. Robot. Automat.*, vol. 5, no. 1, pp. 16–29, 1989.
[5] R. Y. Tsai and R. K. Lenz, "A technique for fully autonomous and efficient 3D robotics hand/eye calibration," *IEEE Trans. Robot. Automat.*, vol. 5, no. 3, pp. 345–358, 1989.
[6] H. H. Chen, "A screw motion approach to uniqueness analysis of hand-eye geometry," in *Proc. CVPR'91*, June 1991, pp. 145–151.
[7] Y. Yakimovsky and R. Cunningham, "A system for extracting three-dimensional measurements from a stereo pair of TV cameras," *CVGIP*, vol. 7, pp. 195–201, 1978.

[8] O. Faugeras and G. Toscani, "The calibration problem for stereo," in *Proc. CVPR'86*, 1986, pp. 15–20.
[9] R. Y. Tsai, "An efficient and accurate camera calibration technique for 3D machine vision," in *Proc. CVPR'86*, 1986, pp. 364–374.
[10] G. Q. Wei and S. D. Ma, "Implicit and explicit camera calibration: Theory and experiments," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 16, no. 5, pp. 469–480, May 1994.
[11] ——, "Camera calibration by vanishing point and cross ratio," in *Proc. IEEE Int. Conf. ASSP*, May 1989, pp. 1630–1633.
[12] J. Weng, P. Cohen, and M. Herniou, "Calibration of stereo cameras using a non-linear distortion model," in *Proc. Int. Conf. Pattern Recognition*, 1990, pp. 246–253.
[13] B. Caprle and V. Torre, "Using vanishing points for camera calibration," *Int. J. Comput. Vision*, vol. 4, no. 2, pp. 127–140, 1990.
[14] S. J. Maybank and O. Faugeras, "A theory of self-calibration of a moving moving camera," *Int. J. Comput. Vision*, vol. 8, no. 2, pp. 123–151, 1992.
[15] J. G. Semple and G. T. Kneebone, *Algebraic Projective Geometry*. Oxford: Clarendon, 1979. (Reprint.)
[16] S. D. Ma, "Conics-based stereo, motion estimation and pose determination," *Int. J. Comput. Vision*, vol. 10, no. 1, pp. 7–25, 1993.
[17] R. Jain, "Direct computation of focus of expansion," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-5, pp. 58–64, 1983.

Song De Ma (SM'91) received the B.S. degree in automatic control from Hsinghua University. He received the Ph.D. degree from the University of Paris 6. He received the "Doctorat d'Etat es Science" degree in image processing and computer vision from INRIA, France.

He was an Invited Researcher in the Computer Vision Laboratory, University of Maryland, College Park, in 1983, and the Robot Vision Laboratory in INRIA, France, during 1984-1986. Since 1986, he has been a Research Professor and the Director of the National Pattern Recognition Laboratory of Institute of Automation in Chinese Academy of Sciences. He is the author or co-author of over 90 journal and conference papers. His current research interests include image processing, 3-D computer vision, robotics, computer graphics, and multimedia technology.

Dr. Ma is a member of the Executive Expert Committee of the Chinese National High Technology Program. He obtained the Best Paper Award and the Best Technique Award in EURGRAPHIC'85 in 1985 and obtained the Outstanding Young Scientist Award of the Chinese Academy of Sciences in 1990. He is an Associate Editor of the *International Journal of Pattern Recognition and Artificial Intelligence*.