Calibrating a Cartesian Robot with Eye-on-Hand Configuration Independent of Eye-to-Hand Relationship

REIMAR K. LENZ AND ROGER Y. TSAI, MEMBER, IEEE

Abstract—**This paper describes a new approach for geometric calibration of Cartesian robots. This is part of a trio for real-time 3-D robotics eye, eye-to-hand, and hand calibration, which use a common setup and calibration object, common coordinate systems, matrices, vectors, symbols, and operations throughout the trio, and is especially suited to machine vision community. It is easier and faster than any of the existing techniques, and is ten times more accurate in rotation than any existing technique using standard resolution cameras, and equal to the state-of-the-art vision based technique in terms of linear accuracy. The robot makes a series of automatically planned movement with a camera rigidly mounted at the gripper. At the end of each move, it takes a total of 90 ms to grab an image, extract image feature coordinates, and perform camera extrinsic calibration. After the robot finishes all the movements, it takes only a few milliseconds to do the calibration. The key of this technique is that only one single rotary joint is moving for each movement while the robot motion can still be planned such that the calibration object remains within the field of view. This allows the calibration parameters to be fully decoupled, and converts a multidimensional problem into a series of one-dimensional problem. Another key is that eye-to-hand transformation is not needed at all during the computation (a rough estimation is needed however to keep the calibration object within the field of view). Results of real experiments are reported. The approach used in viewpoint planning can be of use to the general problem of automatic determination of camera placement.**

Index Terms—**Camera calibration, geometric vision, robot calibration, 3-D location determination, 3-D machine vision, 3-D robotics vision.**

I. INTRODUCTION

In order for a robot to use a video camera to estimate the 3-D position and orientation of a part or object relative to its own base within the work volume, it is necessary to know the relative position and orientation between the hand and the robot base, between the camera and the hand, and between the object and the camera. These three tasks require the calibration of robot, robot eye-to-hand, and camera (see Fig. 1). These three tasks normally require large scale nonlinear optimization, special setup, and expert skills. We have developed a trio for dealing with these three tasks. The trio is:

Manuscript received December 1, 1987; revised August 15, 1988. Recommended for acceptance by O. D. Faugeras.

R. K. Lenz is with the Lehrstuhl für Nachrichtentechnik, Technische Universität München, D-8000 München 2, West Germany.

R. Y. Tsai is with the IBM Thomas J. Watson Research Center, Yorktown Heights, NY 10598.

IEEE Log Number 8928490.

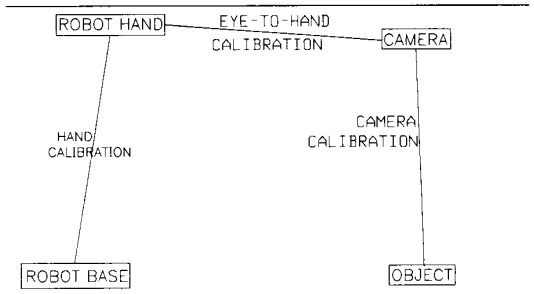


Fig. 1. For obtaining the 3-D position and orientation of the object relative to the robot world base, it is necessary to do three calibrations, namely, robot hand, eye-to-hand and eye (camera) calibration.

1) camera calibration (see [18], [19], [13]).
2) Robot eye-to-hand calibration (see [20], [21]).
3) Cartesian robot hand calibration (this paper).

Note that only the last one is restricted to Cartesian robot, although further research could be done to relieve this latter restriction (some discussion will be given in the Conclusion section). Since the camera calibration is used throughout the trio, it is summarized in Appendix A. It is a two-stage approach, and each stage requires only linear equation solving with a maximum of five unknowns. Radial lens distortion can be corrected. The whole calibration takes about 90 ms, including 25 ms for calibration and 65 ms for image grabbing and feature extraction. The robot eye-to-hand calibration method undergoes the same motion as the robot hand calibration described in this paper, and the same image feature extraction and camera extrinsic calibration are done. The actual computation time besides robot movement and camera calibration is a few milliseconds using a typical minicomputer.

This paper describes the third of the trio. It presents a new approach for robot kinematic calibration, which is easier, faster, and more accurate than comparable current techniques. The purpose of robot kinematic calibration is to determine the elements governing the transformation from the robot joint coordinates to the manipulator end effector Cartesian position and orientation, or the homogeneous transformation from a fixed robot base coordinate frame to its current end effector coordinate frame. This work actually came out as a by-product of an earlier at-

0162-8828/89/0900-0916$01.00 © 1989 IEEE

tempt to do robot eye-to-hand calibration with eye-on-hand configuration in such a way that is decoupled from the robot calibration itself (see [21]). With that goal, it became necessary to do robot calibration independent of eye-to-hand calibration, or else we would have a chicken egg paradox. Then, we succeeded on both counts in terms of developing techniques that do hand-to-eye calibration and robot calibration independently while both use the same equipment and setup.

Existing techniques require laborious procedure for setting up the experiment and time consuming large-scale nonlinear optimization (e.g., [6], [9], [3], [4], [12], [22], [15], [2], [14], [7], [8]). The Selspot related procedures, which use special tracking system to track positions of self-illuminating targets attached to robot (e.g., [10], [1], [5]) is more automatic, but requires infrared light-emitting diode markers (IREDS) that are mounted on some calibration object which is to be attached to the end effector and presurveyed at the factory (see [10]), plus lateral-effect photodiode cameras. The actual computation of the calibration parameters still requires full scale nonlinear optimization.

We propose a technique that requires just a few milliseconds to do the computation (after the robot finishes the moves), which is at least four orders of magnitude (10 000 times) faster than any existing techniques. Furthermore, the new technique can reach ten times better in rotation accuracy (2 mr or better) and equal linear accuracy (one part in four thousandth of the calibration target size) as the state of the art Selspot type technique (see [10]) but used a much more general purpose, off-the-shelf common CCD area array camera, and a calibration target that is used for common camera calibration and is much easier to obtain than the IRED arrangement in Selspot or Watsmart (a vendor system similar to Selspot) system that require special mounting and factory survey. One reason why rotation accuracy is better than state-of-the-art is because the rotation is measured directly from the camera motion, and no large scale nonlinear equations need be solved. Notice that only the linear accuracy depends on the target size or the work volume, not the rotation accuracy. The technique can be used very nicely for Cartesian robot with six degrees-of-freedom (see Fig. 2). Possibilities and prospects for extending to non-Cartesian robot are discussed briefly in the Conclusion.

Since camera extrinsic calibration is used extensively in this paper, a brief overview of the RAC based camera calibration technique (or two-stage camera calibration) is included in the Appendix.

A. Kinematic Calibration of Six-Degree-of-Freedom Cartesian Robot

Fig. 2 shows a typical six-degree-of-freedom Cartesian robot (the IBM 7565. The figure is taken from the IBM7575 Manual). Fig. 3 depicts the three linear joint motions (x, y, z), and three rotary joint motions (roll, pitch, and yaw). Fig. 4 gives another view of the linear and rotary joint actuators and motions. In order to represent the Cartesian position and orientation of the end effector at any given joint configuration, two coordinate frames must be defined. The box frame or robot world coordinate system (RW), and the gripper coordinate system (see Figs. 5 and 6). The RW is located at the intersection of the yaw, pitch, and roll axes when all the joint coordinates are at the origin, and the x, y, and z axes are parallel to the x, y, z linear joint motions. The gripper coordinate frame is used to designate the Cartesian position and orientation of the gripper relative to RW and is

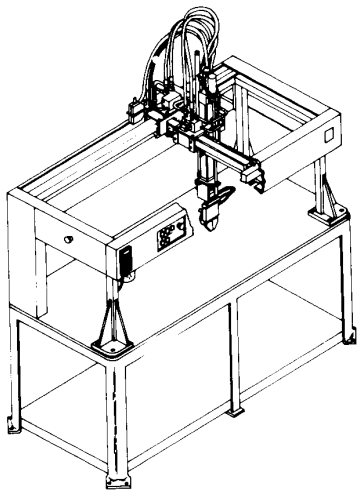


Fig. 2. Typical six-degree-of-freedom Cartesian robot.

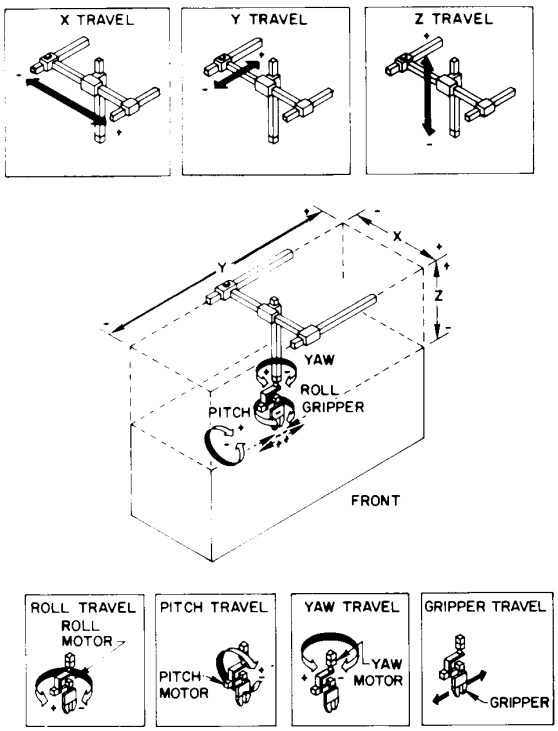


Fig. 3. Depictions of the three-degree-of-freedom linear joint motions and three-degree-of-freedom rotary joint motions

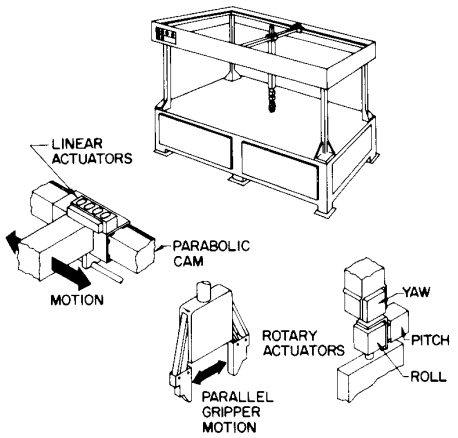


Fig. 4. A close-up look at the six-degree-of-freedom motions plus the gripper grasping motion.

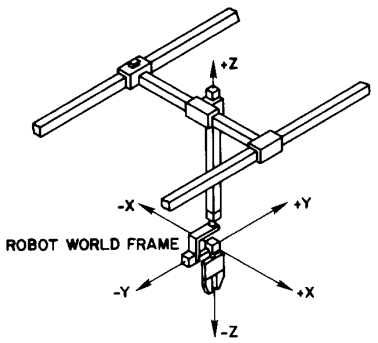


Fig. 5. The robot world coordinate system (RW) is situated at the intersection of the three rotary axes with the x, y, z axes aligned with the x, y, z motions of the robot while all joints are at the origin.

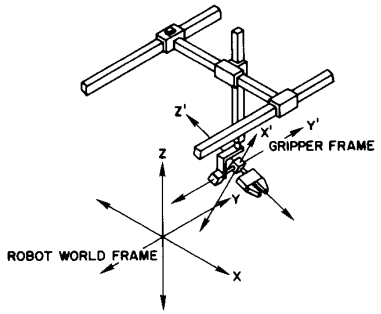


Fig. 6. The gripper frame coordinate system is the same as the robot world coordinate system except that it moves with the gripper as the gripper moves.

also centered on the intersection of the three rotary joint axes. It coincides with RW when all joints are at the origin. Notice in Fig. 6 that the gripper coordinate frame is not centered at the physical gripper fingers or at the tip. The actual offset from the gripper coordinate system to the finger tips is immaterial and is absorbed by the hand-to-eye transformation when a vision sensor is mounted on the end effector (see [20]).

Note that the transformation from robot world RW (in Fig. 5) to gripper frame (in Fig. 6) through yaw, pitch and roll motion is equivalent to rotation around the z axis (yaw), followed by rotation around the new y axis (pitch), followed by rotation around the new z axis (roll). Thus the rotation matrix from gripper coordinate to robot world coordinate system is simply given by

$$R_{gr}(\equiv R_g) = \begin{bmatrix} c_w & s_w & 0 \\ -s_w & c_w & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c_p & 0 & s_p \\ 0 & 1 & 0 \\ -s_p & 0 & c_p \end{bmatrix} \begin{bmatrix} c_r & s_r & 0 \\ -s_r & c_r & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{1}$$

$$= \begin{bmatrix} c_w c_p c_r - s_w s_r & c_w c_p s_r & c_w s_p \\ -s_w c_p c_r - c_w s_r & -s_w c_p s_r + c_w c_r & -s_w s_p \\ -s_p c_r & -s_p s_r & c_p \end{bmatrix} \tag{2}$$

where

$c_w = \cos w$, $s_w = \sin w$.
$c_p = \cos p$, $s_p = \sin p$.
$c_r = \cos r$, $s_r = \sin r$.
w = relative yaw rotation angle from robot world (RW) to gripper frame configuration.
p = relative pitch rotation angle from RW to gripper frame configuration.
r = relative roll rotation angle from RW to gripper frame configuration.

Thus the homogeneous transformation from gripper frame to robot world coordinate system is given by

$$H_{gr}(\equiv H_g) = \begin{bmatrix} R_{gr} & T_{gr} \\ 0\ 0\ 0 & 1 \end{bmatrix} \tag{3}$$

where T_{gr} is simply composed of exactly the x, y, z linear joint coordinates, and R_{gr} is given in (1) and (2).

B. Parameters to Be Calibrated

There are three groups of parameters that relate to the validity of (1) and (2) in terms of transformation from RW grame to gripper frame. They are scales, scale origins, and axes alignment. When the first two groups are corrected or calibrated, (1) and (2) can be used as they are. The situation is different for the third group. After the calibration, unless the axes alignment are perfect, (1) has to be changed in terms of its forms. Equation (19) in Section II-C gives the procedure for making such changes. Only when the axes are perfectly aligned can (1) and (2) be applied. Our experience with IBM Cartesian Box Frame (7565 or Clean Room Robot) indicate that the axis misalignments are minor and insignificant. If the axis misalignment is significant, (1) would become much more complicated.

1) Scales: These are the scale conversion factor from the commanded joint coordinates to the actual angles *roll*, *pitch*, *yaw*, and actual translation T_{gr}.

2) Scale Offsets (Scale Origin): The scale offsets must be such that the gripper frame coincides with the RW frame when all rotary joints are at the origin, and the RW frame x, y, z axes are parallel to the x, y, z joint motions. Only rotary joint scale origins are important. In fact, the Roll Offset is not important. This is due to the fact that it is absorbed by the gripper-to-tool transformation or gripper-to-camera (hand-to-eye) transformation. Only the yaw and pitch offsets are important. They are described in the following.

a) Yaw Offset O_w: When yaw joint is at the origin, ideally, the pitch axis is aligned with box frame x axis. However, due to the kind of axes misalignment, the yaw offset is defined as the yaw joint position where the pitch axis and x axis are best aligned.

b) Pitch Offset O_p: When the pitch joint is at the origin, ideally, the yaw and roll axes are coincident. However, due to yaw-to-roll misalignment, the pitch offset is defined as the pitch joint position where yaw and roll axes are best aligned.

3) Axis Alignment: There are three kinds of axis alignment: rotary-to-rotary, linear-to-linear, and rotary-to-linear.

a) Rotary-to-Rotary:

Directional:

i) Angle between roll and pitch axes (θ_{rp}). This nominally should be 90°.

ii) Angle between pitch and yaw axes (θ_{pw}). This nominally should be 90°.

iii) Angle between roll and yaw axes (θ_{rw}) with pitch axis at its origin. This nominally should be 0. Note that with the proper selection of the pitch origin, the difference between roll and yaw axes in the direction normal to pitch axis should be eliminated. Thus the only misalignment left is along the direction of pitch axis.

Positional: Distance between any two rotary axis: nominally should be zero.

b) Linear-to-Linear: All three linear axes should be mutually orthogonal.

c) Rotary-to-Linear: Yaw and z axis should coincide.

In this paper, we shall focus on scales and scale offsets calibration. For the robot we have used, axis alignment has been quite good, though in this paper, we still will discuss the measurement of directional axis misalignment.

II. The New Approach

A. Basic Principles

In order to do robot kinematic calibration, it is necessary to obtain some estimate of the motion of robot end-effector as a result of robot commanded motion. We mount a TV camera rigidly to the end-effector and fix the view on a stationary calibration object. There are several advantages of doing it this way. First, the calibration object can be the same as that used for any camera calibration, and thus is easy to obtain and the accuracy can be high. Furthermore, one can use solid state TV camera while at the same time capable of reaching 1 part in 4000 accuracy, similar to that obtained using special purpose photodetectors. Finally, the setup is the same as those applications using robot eye-on-hand configuration.

The key idea is to allow only one rotary joint (plus several linear joints) to move at a time, while at the same time keeping the calibration object within the field of view. Furthermore, the final computation is independent of the eye-to-hand relationship. The advantages are described below.

1) Advantages of Allowing Only One Rotary Joint to Move: By allowing only one rotary joint to move, the rotational component of the resultant motion of the end-effector in any "fixed" coordinate system (such as the calibration block world coordinate system) is due solely to the single rotary joint axis, and thus the rotation angle and axis of that particular joint can be readily obtained by estimating the rotational motion of the end-effector. Note that the reason why existing technique require large scale nonlinear optimization is that more than one rotary joint is moving for each single motion.

2) Advantage of Decoupling the Kinematic Calibration with Robot Hand-to-Eye Calibration: One obvious advantage is that the problem becomes a much simpler one when the six-degree-of-freedom transformation from hand-to-eye is out of consideration. The dimensionality of the problem is largely reduced. Also, the camera mounting need not be precise. The only requirement is to be rigid. Furthermore, this makes it possible for Tsai and Lenz's eye-to-hand calibration method [21] to work, since it requires a calibrated robot before doing the hand-eye calibration.

3) Basic Setup: Fig. 7 is a schematic depiction of the basic setup. Figs. 8 and 9 show two photos of the actual setup. The robot carrying a camera makes a series of motions with the camera acquiring a picture of a calibration object at the end of each motion. The calibration object, shown in Fig. 10, is a block with an array of target points on the top surface. The position of each calibration point is known very accurately relative to an arbitrarily selected coordinate system setup on the block (the same calibration target has been used in [18], [13], [19]). The following is a list of definitions for the various coordinate frames. (*Note:* All coordinate frames mentioned here are Cartesian coordinate frames in 3-D.)

G_i The gripper coordinate system at position i. That is, the coordinate frame fixed on the robot gripper and as the robot moves, it moves with the gripper.

C_i The camera coordinate system. That is, the coordinate frame fixed on the camera, with the z axis coinciding with the optical axis, and the x, y axes parallel to the image X, Y axes.

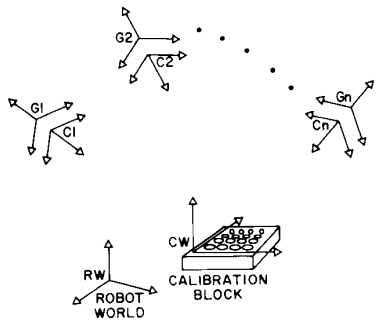


Fig. 7. Basic setup for robot hand/eye calibration. C_i and G_i are coordinate frames for the camera and gripper, respectively.

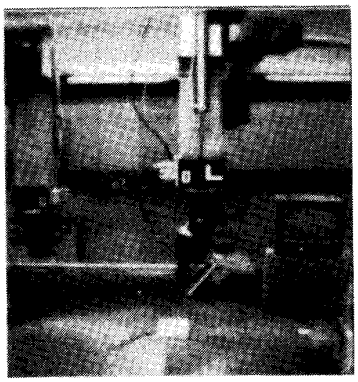


Fig. 8. The physical setup. A CCD camera is rigidly mounted on the last joint of an IBM clean room robot for performing hand-eye calibration.

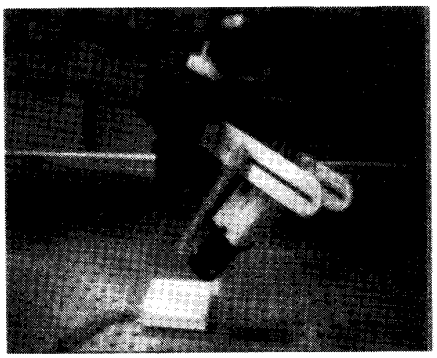


Fig. 9. The physical setup. A CCD camera is rigidly mounted on the last joint of an IBM clean room robot for performing hand/eye calibration.

CW The calibration block world coordinate frame. This is an arbitrarily selected coordinate frame set on the calibration block so that the coordinate of each target point on the calibration block is known *a priori* relative to *CW*.

RW The robot world coordinate frame or the box frame described earlier. It is fixed in the robot work station, and as the robot arm moves around, the encoder output of all the robot joints enable the system to tell where the gripper is relative to *RW*.

Fig. 10. Calibration block is a clear glass plate with 36 disks printed on it using photographic emulsion. The accuracy is 1 micron.

Definition of a List of Homogeneous Transformation Matrix: H_{gi} defines coordinate transformation from G_i to *RW*:

$$H_{gi} \equiv \begin{bmatrix} R_{gi} & T_{gi} \\ 0\ 0\ 0 & 1 \end{bmatrix}. \tag{4}$$

H_{ci} defines coordinate transformation from *CW* to C_i:

$$H_{ci} \equiv \begin{bmatrix} R_{ci} & T_{ci} \\ 0\ 0\ 0 & 1 \end{bmatrix}. \tag{5}$$

H_{gij} defines coordinate transformation from G_i to G_j:

$$H_{gij} \equiv \begin{bmatrix} R_{gij} & T_{gij} \\ 0\ 0\ 0 & 1 \end{bmatrix}. \tag{6}$$

H_{cij} defines coordinate transformation from C_i to C_j:

$$H_{cij} \equiv \begin{bmatrix} R_{cij} & T_{cij} \\ 0\ 0\ 0 & 1 \end{bmatrix}. \tag{7}$$

H_{cg} defines coordinate transformation from C_i to G_i

$$H_{cg} \equiv \begin{bmatrix} R_{cg} & T_{cg} \\ 0\ 0\ 0 & 1 \end{bmatrix}. \tag{8}$$

In the above, i, j range from 1 to N, where N is the number of stations in Fig. 7 where the camera grabs pictures of the calibration block. Fig. 11 illustrates the relationship between the homogeneous matrices and the various coordinate frames in Fig. 7. Note that H_{cg} does not have any station index (i or j). This is because the camera is rigidly mounted on the gripper of the robot arm and therefore, H_{cg} is the same for all stations.

B. Motion Planning

Before we present the complete algorithm for Cartesian robot kinematic calibration, we first describe the motion planning, which is used frequently in the algorithm in the next section. The purpose of motion planning is to plan a trajectory for the manipulator such that only one rotary joint is moving at one time, and that the calibration block is always approximately centered in the camera's field of view.

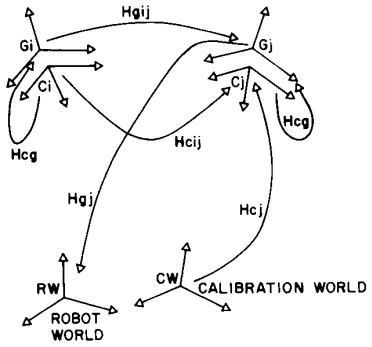


Fig. 11. Relationship between the homogeneous matrices and the coordinate frames.

1) Outline of the Strategy or Approach in Motion Planning:

Step 1: Find a system of equations for compensating the change of one rotary joint motion with linear joint motions.

This system of equations should have the following characteristics.

1) It should not have any unknowns other than the rotary and linear joint motions. All other quantities should be known or directly computable.

2) It should represent the necessary and sufficient condition for the camera to be at proper focusing distance when reviewing a calibration block, and the calibration block should be centered in the field of view.

3) It should impose only three constraints on the linear joint motions, so that the three degree-of-freedom linear joints can solve it uniquely.

4) It should have closed form solutions.

Step 2: Single rotary joint stepping.

For each of the rotary joints, traverse a discrete number of steps (N) from one end of the physical limit to the other, while keeping the other two rotary joints fixed. For each step, use the equations in Step 1 to determine the necessary x, y, z linear joint configurations needed to keep the calibration block within the field of view.

Note: If the x, y, z linear joints have unequal accuracy, then for each rotary joint that is supposed to vary, try to arrange the other two fixed rotary joints such that the equations in Step 1 will yield minimum movement for the least accurate linear joint. This turns out to be very important for our robot, since we have some problems with the x axis.

2) System of Equations for Compensating the Change of One Rotary Joint Motion with Linear Joint Motions: The equation sought in step 1 of Section II-B1) is

$$T_g = T_{wr} - R_g T_{wg} \tag{9}$$

where

$$T_{wg} = R_{cg} T_c + T_{cg} \tag{10}$$

$$T_c = \begin{bmatrix} 0 \\ 0 \\ D \end{bmatrix} \tag{11}$$

D is the focusing distance

T_g, R_g are defined in (1)

T_{cg} and R_{cg} are defined in (8).

The magnification of the optics is arranged such that the calibration object approximately fills the field of view. Before we prove it, first observe that the only unknown variables in (9) are T_g and R_g. All other variables, such as T_{wr} (translation from CW to RW), R_{cg}, T_{cg}, are easily computed using the hand-to-eye calibration technique described in [21]. Note that although the latter requires the knowledge of robot eye-to-hand transformation, this does not contradicts the statement made earlier that the final computation for the proposed calibration method does not depend on hand-to-eye transformation. The reason is that the motion planning is only intended for placing the camera such that the calibration object is within the field of view. The precision is not of concern here.

Derivation of (9): By observing the fact that H_g is an abbreviation of H_{gr} and by using the simple chain rule for the product of homogeneous transformation matrices (i.e., $H_{bc} H_{ab} = H_{ac}$ for Cartesian systems a, b and c), we have

$$H_g H_{wg} = H_{wr}. \tag{12}$$

Equation (12) is equivalent to

$$\begin{bmatrix} R_g & T_g \\ 0\ 0\ 0 & 1 \end{bmatrix} \begin{bmatrix} R_{wg} & T_{wg} \\ 0\ 0\ 0 & 1 \end{bmatrix} = \begin{bmatrix} R_{wr} & T_{wr} \\ 0\ 0\ 0 & 1 \end{bmatrix}$$

which yields (9).

Derivation of (10): Again, using the simple chain rule argument, we have

$$H_{wg} = H_{cg} H_{wc} = H_{cg} H_c.$$

By expanding H into R and T, (10) follows.

Derivation of (11): Since the calibration block center (or origin of CW coordinate system) should be at the center of the field of view, it must also be on the optical or z axis in camera 3-D coordinate system. Thus it must be at position $[0\ 0\ D]^T$ in the camera coordinate system where D is the focusing distance. However, the position of the origin of CW coordinate system in the camera coordinate system is exactly T_c, since $[0\ 0\ 0\ 1]^T$ is the homogeneous coordinate of the origin, and

$$H_c \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} T_c \\ 1 \end{bmatrix}.$$

Notice that (9) satisfies all the four criteria or characteristics listed in Step 1 in Section II-B1), and can be easily employed in Step 2 when varying one of the rotary joint while using T_g to balance the equations. We now briefly describe the motion planning for each rotary joint separately.

3) Yaw Motion Planning: The yaw joint is supposed to vary from one physical limit to the other while keeping the pitch and roll joints constant, using linear joints to compensate it so that the calibration object maintains within the field of view. However, by properly selecting the fixed pitch and roll joint values, T_g can stay constant independent of the yaw joint. It can be shown that the fixed values for roll and pitch should be as follows:

$$r = \tan^{-1}\left(\frac{T_{wgx}}{T_{wgy}}\right) \tag{13}$$

$$p = \tan^{-1}\left(\frac{c_r T_{wgx} + s_r T_{wgy}}{T_{wgz}}\right] \tag{14}$$

where T_{wgx}, T_{wgy}, T_{wgz} are the x, y, z components of T_{wg} in (9). In such case, T_g stays at the following fixed value:

$$T_g = T_{wr} + \begin{bmatrix} 0 \\ 0 \\ |T_{wg}| \end{bmatrix}. \tag{15}$$

The derivations are given in Appendix A.

4) Pitch Motion Planning: Due to the structure of R_g and (9), it can be shown that by setting yaw to be fixed at 90 degrees, the x component of T_g need not change while the pitch joint varies from one extreme to the other. This has great advantage in our case since our x axis is the least accurate one among the three linear joints. The roll joint is arbitrarily set at the value used in (13) in yaw motion planning, so that the transition from yaw motion to pitch motion would entail least amount of motion.

5) Roll Motion Planning: Again, due to the structure of R_g and (9), if yaw is set at 90° and pitch at 0 degree, the z component of T_g need not change while the roll joint varies from one limit to the other. Also, the x component would undergo very minor motion.

C. The Cartesian Robot Calibration Algorithm

1) Estimation of Yaw Rotation Axis P_w and Rotation Angle θ_{cwi}:

a) Divide the total range of yaw joint motion into N divisions. For every joint position, associate an index i, where $i = 1, \cdots, N-1$. Initialize i to 0.

b) Move the manipulator to the ith position according to the Yaw motion planning in Section II-B.

c) Record the robot joint coordinate $J_i(= \langle x_i, y_i, z_i, r_i, p_i, w_i \rangle)$ obtained through Robot's own joint position encoding capability. Compute the relative commanded rotation angle from position 0 to position i by $\theta_{jwi} = w_i - w_0$.

d) Take a view. Compute image coordinates of the calibration points. Do camera extrinsic calibration (see [19]) to obtain H_{ci}, the homogeneous transformation from calibration block to camera. The combined process of image grabbing, feature extraction and camera extrinsic calibration takes only about 90 milliseconds on a 68000 based minicomputer and standard image grabbing, feature extraction hardware. A brief overview of the RAC based camera calibration technique is included in the Appendix.

e) If $(i > 0)$, then compute the rigid body motion in calibration world coordinate system (CW) that transforms the robot gripper from position 0 to position i for each i. It is shown in Appendix B that this rigid body motion with rotation R_i and translation T_i is simply given by

$$\begin{bmatrix} R_i & T_i \\ 0\ 0\ 0 & 1 \end{bmatrix} = H_{ci}^{-1} H_{c0} \tag{16}$$

which is independent of H_{cg}, the eye-to-hand transformation. From (16), R_i is obtained, and the yaw rotation axis in the CW coordinate system is computed from

$$P_w = \frac{\text{Skew}^{-1}(R_i)}{\left|\text{Skew}^{-1}(R_i)\right|} \tag{17}$$

and the relative rotation angle θ_{wi} from position i to position 0 is given by

$$\theta_{cwi} = \tan^{-1}\left[\tfrac{1}{2}\left|\text{Skew}^{-1}(R_i)\right|\right] \tag{18}$$

where $\text{Skew}^{-1}(R)$ is defined as

$$\text{Skew}^{-1}(R) \equiv \begin{bmatrix} r_{32} - r_{23} \\ r_{13} - r_{31} \\ r_{21} - r_{12} \end{bmatrix}$$

and r_{ij} is defined as the ijth element of R.

f) If $i < N - 1$, then $i = i + 1$, go to b).

2) Estimation of Pitch Rotation Axis P_p and Rotation Angle θ_{cpi}: The procedure is the same as yaw motion in 1 except that the motion sequence planned for pitch rather than yaw in Section II-B is used, and the rotation axis P_p and rotation angle θ_{cpi} are computed using (17) and (18) as well.

3) Estimation of Roll Rotation Axis P_r and Rotation Angle θ_{cri}: The procedure is the same as yaw motion in 1 except that the motion sequence planned for pitch rather than yaw in Section II-B is used, and the rotation axis P_r and rotation angle θ_{cri} are computed using (17) and (18) as well.

4) Estimation Of Y Motion Axis P_y:

a) Move the manipulator to a position where the calibration object is approximately centered at the field of view, but instead of filling the field of view as in previous cases, the calibration object fills 80 percent (lengthwise) of the field of view. This is done by increasing D in (11) by 20 percent, and use (10) and (9) with R_g borrowed from that used in position 0 for the yaw motion planning to obtain T_g.

b) Move manipulator in the $-y$ direction with a distance $1/10$ of the size of calibration object. Since D has been increased by 20 percent, this guarantees that the calibration object is still within the field of view. Take a view. Do extrinsic calibration. Compute H_{c1}.

c) Move the manipulator to $+y$ direction for $1/5$ of the size of calibration object. Take a view. Do extrinsic calibration. Compute H_{c2}.

d) From H_{c1} and H_{c2}, compute the vector representing the manipulator motion in calibration world coordinate system. Unlike the rotary joint cases, (16) is not used to compute the motion. Since the motion is purely translational, it can be estimated by computing the linear motion of the origin of G_i or C_i coordinate frames in CW coordinate system. The origin of C_i in the CW coordinate system is $-R_{ci}^T T_{ci}$ and the origin of C_0 in CW is $-R_{c0}^T T_{c0}$. Thus P_y is computed by

$$P_y = -R_{ci}^T T_{ci} - (-R_{c0}^T T_{c0}) = R_{c0}^T T_{c0} - R_{ci}^T T_{ci}.$$

e) Repeat a through d several times to obtain an average estimation of P_y.

5) Scale and Offset Computation:

a) Rotary Joint Scale: Let S_w, S_p, S_r be the scale factors converting the computer rotary joint coordinates to real joint coordinates. Then compute S_w, S_p, S_r by

$$S_w = \frac{1}{N}\sum_{i=1}^{N}\frac{\theta_{cwi}}{\theta_{jwi}}$$

$$S_p = \frac{1}{N}\sum_{i=1}^{N}\frac{\theta_{cpi}}{\theta_{jpi}}$$

$$S_r = \frac{1}{N}\sum_{i=1}^{N}\frac{\theta_{cri}}{\theta_{jri}}.$$

Here, we assume the scale is linear. Of course any other kind of curve fitting can be employed for higher order scale.

b) Linear Joint Scale: Due to the simplicity of the linear joint motions, the scale factor calibration can be done easily with a variety of ways. One way is simply to place a long and narrow calibrated object (like a calibrated ruler. Also Newport Corp. offers long and narrow high precision calibration objects) along the linear axis motion direction, and move the linear joint while the camera focuses on one portion of the calibration object. The scale factor S_x, S_y, S_z between the actual movement and the command space movement gives the scale factor conversion. Note that the alignment between the calibrated ruler and the linear joint axis need not be precise since that would only have a cosine effect on the scale. Another way is to use laser interferrometer. For linear motion, this is very easy to do.

c) Rotary Scale Offset:

$$\text{Yaw offset} = O_w = \angle(P_w, P_y) - \frac{\pi}{2}$$

$$\text{Pitch offset} = O_p = \angle(P_w, P_r)$$

where P_w, P_y are computed in 1)-e) and 4)-e). $\angle(P_w, P_y)$ means the angle between P_w and P_y

6) Rotary-to-Rotary Axis Alignment:

$$\theta_{rp} = \angle(P_r, P_p)$$

$$\theta_{pw} = \angle(P_p, P_w)$$

$$\theta_{rw} = \angle(P_r, P_w)$$

where P_r, P_p, P_w, are computed in (17) and (18).

7) Computation of Real Joint Values (w_r, p_r, r_r, x_r, y_r, z_r) from Commanded Joint Values (w_c, p_c, r_c, x_c, y_c, z_c):

$$w_r = (w_c - O_w)S_w$$

$$p_r = (p_c - O_p)S_p$$

$$r_r = (r_c - O_r)S_r$$

$$x_r = x_c S_x$$

$$y_r = y_c S_y$$

$$z_r = z_c S_z.$$

8) Correction of R_{gr} in (1) Given the Rotary-to-Rotary Axis Alignment Calibration: Note that R_{gr} in (1) is derived from the following three successive rotations

$$\{\textit{rotate around } z \textit{ with } \theta_w\}\{\textit{rotate around } y' \textit{ with } \theta_p\}\{\textit{rotate around } z' \textit{ with } \theta_r\}$$

where the y' and z' mean the new y and z axes after the preceding rotation. With the directional rotary-to-rotary axes misalignment calibrated, R_{gr} actually consists of five rotations:

$$\{\textit{rotate around } z \textit{ with } \theta_w\}\{\textit{rotate around } x' \textit{ with } \theta_{pw}\}\{\textit{rotate around } y' \textit{ with } \theta_p\}\{\textit{rotate around } x' \textit{ with } \theta_{rp}\}\{\textit{rotate around } z' \textit{ with } \theta_r\}.$$

From this, the new R_{gr} is given by

$$R_{gr}(\equiv R_g) = \begin{bmatrix} c_w & s_w & 0 \\ -s_w & c_w & 0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta_{wp} & \sin\theta_{wp} \\ 0 & -\sin\theta_{wp} & \cos\theta_{wp} \end{bmatrix} \cdot \begin{bmatrix} c_p & 0 & s_p \\ 0 & 1 & 0 \\ -s_p & 0 & c_p \end{bmatrix}\begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta_{pr} & \sin\theta_{pr} \\ 0 & -\sin\theta_{pr} & \cos\theta_{pr} \end{bmatrix} \cdot \begin{bmatrix} c_r & s_r & 0 \\ -s_r & c_r & 0 \\ 0 & 0 & 1 \end{bmatrix}. \quad (19)$$

III. Experimental Results

We test the accuracy of the proposed technique by how accurately the calibrated robot can place or position a camera in the robot work volume. This is tested in the following steps.

Step 1: Calibrate the robot using the procedure described in this paper. The number of stepping N for each rotary joint is taken be be 10.

Step 2: Move the manipulator of $2M$ different positions where M is greater than 2. For each station i, compute camera to calibration block homogenous transformation H_{ci} using extrinsic calibration. This takes about 90 ms per station. The calibrated robot gripper position and orientation relative to robot world, which is H_{gi} in (1) and (2), is also recorded.

Step 3: Compute H_{cg} using procedures described in [21] using data from station 1–M.

Step 4: For each station k (k from 1 to M), compute homogenous matrix H_{RC} (homogenous transformation from robot world frame RW to calibration block world frame CW) by

$$H_{RC} = H_{gi}^{-1} H_{cg}^{-1} H_{ci}^{-1}.$$

Make an average of H_{RC} computed from these M stations.

Step 5: Let stations $M + 1$ through $2M$ be called verification stations. For each of the verification stations, predict the position and orientation of the camera relative to robot world base coordinate RW by $H_{cg}^{-1} H_{gk}^{-1}$ where k is the station index, and H_{gk} is the calibrated robot gripper position and orientation relative to robot world and is computed from robot joint coordinates. Compare this predicted position and orientation to $H_{ck}^{-1} H_{RC}$, where H_{ck} is computed in Step 1 while H_{RC} is computed in Step 3.

The results of a series of experiments yield the following table:

	New Camera Pose Prediction Error	
M	Rotation error	Translation error
4	4.568 mrad	23.238 mil
6	3.304 mrad	19.078 mil
8	3.264 mrad	26.712 mil
10	2.888 mrad	14.642 mil
12	2.782 mrad	12.516 mil

In the above table, mil means one thousandth of an inch. Notice that the number of stations M in Step 4 is different in meaning from the number of divisions N for the rotary joint stepping. The effect of M is described in [21]. Observe that the error of predicted camera pose includes both the error of the calibrated hand/eye relationship and the robot's positioning error. Note from the table that for ten stations ($M = 10$), the rotation error is about 2.88 mr. This is contributed from four sources: 1) The robot calibration error. 2) The robot eye-to-hand calibration. 3) The robot's repeatability. 4) Robot positioning resolution. The repeatability for the rotary joint of the robot we are using is about 2 mr, and the resolution is about 1 mr. According to the accuracy formula (32) in [20], the rotational component of the eye-to-hand transformation error is about 2.557 mr. These all mean that the robot calibration error should be of the order of magnitude of 2.5 mr. Since the accuracy of the proposed calibration scheme should be of the order of accuracy for the camera extrinsic calibration, which is indeed of this order of magnitude (see [19]), this means that the experimental results agree quite well with theory.

Notice from the table that for 10 stations, the translation error is about 14 mil. Using the error formula in (32) of [20], scaled by $\sqrt{10/3}$ to reflect the effect of M, the error of T_{cg} is predicted to be 10.66 mil. For the linear joints, our robot's repeatability and resolution are of the order of 4 mil. This means that the robot calibration error is anywhere from 4 mil to 10 mil, which is quite reasonable.

IV. Conclusion

The main advantages of the proposed approach are as follows.

1) It is part of a unified approach to robot eye, eye-to-hand, and hand calibration trio. This greatly facilitates the ease of implementation and uncertainty analysis for the purpose of sensor planning.

2) It is fast. The robot makes a series of automatically planned moves, and at the end of each move, it takes total of 90 ms to do image grabbing, feature extraction and camera extrinsic calibration. After the robot finishes the moves, it takes only a few milliseconds to do the final computations, as opposed to several minutes to hours for the conventional full scale nonlinear optimization approach.

3) It is fully autonomous. The camera used is common rugged TV camera, and can be mounted on the robot hand in an arbitrary fashion (no special or painstaking alignment is needed). The algorithm does not need manual intervention of any kind.

4) It is simple. The algorithm is very easy to implement.

5) It is accurate. The new technique can reach ten times better in rotation accuracy (2 mr or better) and equal linear accuracy (one part in four thousandth of the calibration target size) as the state of the art vision-based Selspot type technique.

6) It is convenient. It uses the same setup as common robotics vision eye-on-hand applications, and does not require any special tools or setup other than those used for common machine vision camera calibration.

As discussed before, among the complete set of parameters, the most critical parameters are the scales, scale origins and directional rotary axis alignment. Other parameters are not critical (at least for the robot we have been using). The idea of using as few rotary joints as possible while keeping the calibration object within the field of view can also be applied to robots other than six-degree-freedom Cartesian robots, so long as if the calibration plate can remain within the field of view while the robot performs single rotary joint stepping. If this is impossible, then the same spirit can be applied except that the number of rotary joints that are in operation should be kept minimum while keeping the calibration object within the field of view. Some small dimensional nonlinear equations probably are not avoidable for non-Cartesian robot, but at least it is much better than full scale nonlinear min-

imization. The other main advantages are still maintained, namely, the simplicity of the calibration object and the calibration procedure. It is also possible that for some non-Cartesian robots, only part of the parameters can be calibrated using the proposed approach. Even so, this technique can still be useful for non-Cartesian robots, since when part of the parameters are calculated with great ease, the dimensionality of the problem is greatly reduced, and thus the problem becomes easier.

Appendix A
Review of RAC-Based Camera Calibration

Extrinsic camera calibration is an essential part of the robot calibration task described in this paper. The specific algorithm used for extrinsic camera calibration was the RAC-based Camera Calibration described in [18], [19]. In this Appendix, we briefly overview its basic approach. We will only overview the case where the calibration points are coplanar, since this is the only one used in this paper.

Camera calibration is the problem of determining the elements that govern the relationship or transformation between the 2-D image that a camera perceives and the 3-D information of the object. There are two kinds of parameters that define this 2-D/3-D relationship: the intrinsic and extrinsic parameters. The intrinsic parameters characterize the inherent properties of the camera and optics, including image center (C_x, C_y), image scale factor S_x, lens principal distance ("effective focal length") and lens radial distortion coefficient. Some have criticized that the RAC-based calibration requires the *a priori* knowledge of the manufacturer supplied y scale factor. Note that there is *no need* to determine the y component of the image scale factor. The reason is simply that only the product of focal length and scale factors can be determined uniquely, not its individual magnitude. This is an inherent property and is true for all camera calibration techniques. However, if one would like to have an explicit value of focal length to be determined, then it is necessary to have an explicit value for one of the x or y scale factors to be known *a priori*. Since the y scale value is always available from the camera manufacturer up to *submicron* accuracy when a solid state discrete array camera is used, the focal length can always be determined uniquely. The extrinsic parameters represent the position and orientation of the camera with respect to the world coordinate system, including translation (T_x, T_y, T_z) and rotation about the X, Y, Z axes. In the following, we briefly review the camera model and the algorithm for the RAC-based calibration.

A. The Camera Model

Fig. 12 illustrates the basic geometry of the camera model. The overall transformation from the 3-D object coordinate (x_w, y_w, z_w) to the computer image frame buffer coordinate (X_f, Y_f) can be decomposed into the following four steps.

Step 1: Rigid body transformation from the object

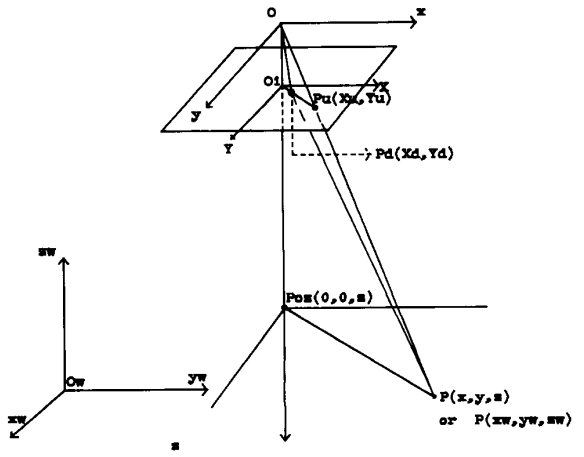


Fig. 12. Camera geometry with perspective projection and radial lens distortion.

world coordinate system (x_w, y_w, z_w) to the camera 3-D coordinate system (x, y, z)

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = R \begin{bmatrix} x_w \\ y_w \\ z_w \end{bmatrix} + T \tag{A1}$$

where R and T are the 3×3 rotation matrix and the translation vector

$$R \equiv \begin{bmatrix} r_1 & r_2 & r_3 \\ r_4 & r_5 & r_6 \\ r_7 & r_8 & r_9 \end{bmatrix}, \qquad T \equiv [T_x \quad T_y \quad T_z]^T. \tag{A2}$$

Parameters to be calibrated: R and T.

Step 2: Transformation from 3-D camera coordinate (x, y, z) to ideal (undistorted) image coordinate (X_u, Y_u) using perspective projection with pin hole camera geometry

$$X_u = f\frac{x}{z} \qquad Y_u = f\frac{y}{z}. \tag{A3}$$

Parameters to be calibrated: effective focal length f.

Step 3: Radial lens distortion:

$$X_d = 2X_u/D \qquad Y_u = 2Y_u/D \tag{A4}$$

where (X_d, Y_d) is the distorted or true image coordinate on the image plane

$$D = 1 + (1 - 4\kappa_1 R_u^2)^{1/2} \qquad R_u^2 = X_u^2 + Y_u^2$$

Note that (A4) is equivalent to $X_u = X_d/(1 + \kappa_1 R_d^2)$, $Y_u = Y_d/(1 + \kappa_1 R_d^2)$ with $R_d^2 = X_d^2 + Y_d^2$. Parameters to be calibrated: distortion coefficients κ_1.

Note: Note that only one distortion coefficient is sufficient, and taking more distortion parameters into account only does harm than good, since the contribution of the other parameters only makes a difference of less than 1/30

of a pixel, with a price of increased computational load and numerical instability.

Step 4: Real image coordinate (X_d, Y_d) to computer image coordinate row and column coordinate (X_f, Y_f) transformation

$$X_f = S_x X_d + C_x \qquad Y_f = S_y Y_d + C_y \tag{A5}$$

where

(C_x, C_y)	Computer image coordinate for origin in image plane.
S_x	Scale factors in x.
S_y	Scale factors in y.

Parameters to be calibrated: S_x and the image origin (C_x, C_y).

B. Approach and Algorithm

In order to avoid large scale nonlinear optimization while at the same time maintaining exact modeling without making any approximations, we use radial alignment constraint (RAC) to decouple the calibration parameters into two groups, each group can be solved easily and rapidly. RAC is not a newly imposed artificial constraint, but rather follows trivially from the definition of the camera model. It states the fact that the direction of the vector extending from the origin in the image plane to the image point is radially aligned (or parallel) with the vector extending from the optical axis (or more precisely, the projection point of the object point on the optical axis) to the object point. The validity of RAC is independent of the radial lens distortion, the focal length and the z component of the translation. Therefore, the algebraic equations established as a necessary and sufficient condition of RAC must be only functions of the rest of the parameters, which consist primarily of extrinsic parameters. This leads to the following strategy:

- Solve group 1 with radial alignment constraint.
- Solve group 2 with projective equations.

This results in the following two-stage approach:

Stage 1:

Radial Alignment Constraint $\xrightarrow{\text{linear}}$ 5 parameters $\xrightarrow{\text{trick}}$ R, T_x, T_y

Stage 2:

R, T_x, T_y + projective constraint $\xrightarrow{\text{linear}}$ f, T_z, κ_1

The image scale factor can be computed if multiplane calibration points are used. The algorithm can be found in [18]. The scale and image center can also be estimated using techniques described in [13]. From the radial alignment constraint to the five intermediate parameters, only linear equations with five unknowns need be solved. The linear equation corresponding to each calibration point i is

$$[Y_{di}x_{wi} \quad Y_{di}y_{wi} \quad Y_{di} \quad -X_{di}x_{wi} \quad -X_{di}y_{wi}] \cdot L = X_{di} \tag{A6}$$

where L is a column vector containing the five unknowns $T_y^{-1}r_1$, $T_y^{-1}r_2$, $T_y^{-1}r_4$, $T_y^{-1}r_5$, $T_y^{-1}T_x$. With N (the number of object points) larger than five, an overdetermined system of linear equations can be established and solved for the five unknowns. Since the algorithm is quite fast, even with 36 points, the computation time with a minicomputer is less than 25 ms. From the five parameters to R, T_x and T_y, it takes a trick. It is called a trick because it is tricky to derive and prove, but is simple and easy to compute.

The linear equation to be solved in Stage 2 can be simply derived from (A1)–(A5), giving

$$H_{xi}f + H_{xi}R_{di}^2 f\kappa_1 - X_{di}T_z = X_{di}(x_{wi}r_7 + y_{wi}r_8)$$
$$H_{yi}f + H_{yi}R_{di}^2 f\kappa_1 - Y_{di}T_z = Y_{di}(x_{wi}r_7 + y_{wi}r_8) \tag{A7}$$

where

$$H_{xi} = x_{wi}r_1 + y_{wi}r_2 + T_x \quad H_{yi} = x_{wi}r_4 + y_{wi}r_5 + T_x,$$
$$R_{di}^2 = X_{di}^2 + Y_{di}^2$$
$$X_{di} = S_x^{-1}(X_{fi} - C_x), \; Y_{di} = S_y^{-1}(Y_{fi} - C_y).$$

In (A7) above, all the unknowns are printed as bold letters. It is seen that if f, $f\kappa_1$, and T_z are regarded as unknowns (notice that they are independent), then (A7) is linear. This makes stage 2 linear too. The complete set of calibration parameters in stage 1 and 2 can be solved with linear equations with no more than five knowns. This makes the RAC-based calibration technique faster than any existing techniques, including the DLT (see [Abdel-Aziz and Karara, 1971], [1974], [Karara, 1979]). Since 25 ms (excluding image feature extraction time, which is 65 ms) is the total computation time, it is at least 10 times faster than DLT related algorithm, and is 10 times more accurate. It is more than 1000 times faster than the full scale nonlinear optimization based algorithm but is equally accurate.

Appendix B
Derivations for the Yaw Motion Planning

We now prove that by fixing roll and pitch joints to be at the values given in (13) and (14), T_g need not change (fixed at the value given by (15) when the yaw joint moves while still keeping the calibration object within the field of view and under proper focusing distance.

Note from (9) that normally, when the yaw varies, R_g varies, and thus $R_g T_{wg}$ varies, implying that T_g has to change (note that T_{wr} in (9) is fixed since RW and CW frames are fixed). If T_g does not change while yaw varies, this must put some additional constraint on R_g in order to make this possible. In fact, this constraint is enough to fix roll and pitch to unique values, as to be seen.

Substituting (1) into (9) gives

$$T_g = T_{wr} - \begin{bmatrix} c_w & s_w & 0 \\ -s_w & c_w & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c_p & 0 & s_p \\ 0 & 1 & 0 \\ -s_p & 0 & c_p \end{bmatrix} \cdot \begin{bmatrix} c_r & s_r & 0 \\ -s_r & c_r & 0 \\ 0 & 0 & 1 \end{bmatrix} T_{wg} \quad \text{(B1)}$$

$$= T_{wr} - A_w B \quad \text{(B2)}$$

where

$$B = \begin{bmatrix} c_p & 0 & s_p \\ 0 & 1 & 0 \\ -s_p & 0 & c_p \end{bmatrix} \begin{bmatrix} c_r & s_r & 0 \\ -s_r & c_r & 0 \\ 0 & 0 & 1 \end{bmatrix} T_{wg} \quad \text{(B3)}$$

$$A_w = \begin{bmatrix} c_w & s_w & 0 \\ -s_w & c_w & 0 \\ 0 & 0 & 1 \end{bmatrix}. \quad \text{(B4)}$$

In order for $A_w B$ to be independent of w, it is necessary that the 3×1 vector B be of the following form:

$$B = \begin{bmatrix} 0 \\ 0 \\ b \end{bmatrix} \quad \text{(B5)}$$

where b is some constant. Combining (B2), (B3), and (B5), we have

$$\begin{bmatrix} 0 \\ 0 \\ T_{wrz} - T_{gz} \end{bmatrix} = \begin{bmatrix} c_p & 0 & s_p \\ 0 & 1 & 0 \\ -s_p & 0 & c_p \end{bmatrix} \begin{bmatrix} c_r & s_r & 0 \\ -s_r & c_r & 0 \\ 0 & 0 & 1 \end{bmatrix} T_{wg} = C_{pr} T_{wg} \quad \text{(B6)}$$

where

$$C_{pr} = \begin{bmatrix} c_p c_r & c_p s_r & s_p \\ -s_r & c_r & 0 \\ -s_p c_r & -s_p s_r & c_p \end{bmatrix}$$

T_{wrz}, T_{gz} are the z components of T_{wr}, T_g.

From the second row of (B5), we have

$$0 = s_r T_{wgx} - c_r T_{wgy} \quad \text{(B7)}$$

thus (13) follows. Substituting (B7) into the first row of (B6) yields (14).

Since the first two rows of (B6) are 0, T_{wg} must be simultaneously orthogonal to the fist and second rows of C_{pr}. But since C_{pr} is itself an orthonormal matrix, we see that T_{wg} is colinear with the third row of C_{pr}, i.e., the third row of C_{pr} is $\pm T_{wg}/|T_{wg}|$. Thus,

$$T_{gz} = T_{wrz} \pm |T_{wg}|. \quad \text{(B8)}$$

We now show that only the plus sign is valid. Note that T_{wr} is the translation vector for the coordinate transformation from CW to RW frames. Thus it is the vector pointing from the origin of RW to the origin of CW in CR coordinate system. Since the calibration block is on the robot table, and CR origin is in the middle of the work volume, T_{wr} is pointing down and is negative. If T_{gz} in (B8) is positive, then the sign for $|T_{wg}|$ must be positive. If however that T_{gz} is negative, its magnitude must be smaller than that of T_{wr}, or else the manipulator will hit the table. Thus, the sign for $|T_{wg}|$ is always positive, leading to

$$T_g = T_{wr} + \begin{bmatrix} 0 \\ 0 \\ |T_{wg}| \end{bmatrix} \quad \text{(B9)}$$

which is constant independent of yaw joint value.

Q.E.D.

Appendix C
Computation of Rigid Body Motion of Robot Gripper in CW Coordinate System

Let R_i, T_i be defined as the 3-D rigid body rotation and translation motions that the gripper undergoes from position 0 to position i in the CW coordinate system, and let H_i be defined as

$$H_i = \begin{bmatrix} R_i & T_i \\ 0\ 0\ 0 & 1 \end{bmatrix}.$$

It is important to note that H_i is not simply $H_{gi}^{-1} H_{g0}$, which is the coordinate transformation from G_0 to G_i. The latter is dependent on the eye-to-hand transformation R_{cg} and T_{cg}. Now we show that H_i is actually $H_{gi} H_{g0}^{-1}$ But since H_{ci}'s are the only observables, we further show that the rigid body motion H_i is $H_{ci}^{-1} H_{c0}$. To do so, it is most convenient to find four points whose coordinates are known in a fixed coordinate system CW before and after the motion. The end points of the unit coordinate frame G_0 plus its origin suit the purpose here. By definition, the homogeneous coordinates of the end points of x, y, and z axes and the origin of G_0 in CW are

$$H_{g0} \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \end{bmatrix}, \quad H_{g0} \begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \end{bmatrix}, \quad H_{g0} \begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \end{bmatrix}, \quad H_{g0} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix},$$

respectively. Thus we define H_a as a 4×4 matrix where the homogeneous coordinates of the four said points are its four columns. Then

$$H_a = H_{g0} \begin{bmatrix} & & & 0 \\ & I & & 0 \\ & & & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix}.$$

Similarly, if we define H_b to be the same as H_a except that it is used after the motion. Then

$$H_b = H_{gi}\begin{bmatrix} & & & 0 \\ & I & & 0 \\ & & & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

Obviously, the rigid body motion H_i satisfies the following

$$H_b = H_i H_a.$$

Thus

$$H_i = H_b H_a^{-1} = H_{gi}\begin{bmatrix} & & & 0 \\ & I & & 0 \\ & & & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix}\left\{ H_{g0}\begin{bmatrix} & & & 0 \\ & I & & 0 \\ & & & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix}\right\}^{-1}$$

$$= H_{gi}H_{g0}^{-1}. \qquad \text{(C1)}$$

Note that $H_{gi} = H_{ci}^{-1}H_{cg}^{-1}$ and $H_{g0} = H_{c0}^{-1}H_{cg}^{-1}$. Substituting these two into (C1) gives

$$H_i = H_{ci}^{-1}H_{cg}^{-1}H_{cg}H_{c0} = H_{ci}^{-1}H_{c0}.$$

Equation (16) is now proved.

References

[1] E. K. Antonsson, "A 3-D kinematic acquisition and intersegmental dynamic analysis system for human motion," Ph.D. dissertation, Dep. Mech. Eng., M.I.T., 1982.

[2] M. Bowman and A. Forrest, "Robot model optimization," 1987, submitted for publication.

[3] L. M. Chao and J. C. S. Yang, "Development and implementation of a kinematic parameter identification technique to improve the positioning accuracy of robots," in *Robots 10 Conf. Proc.*, Chicago, IL, 1986.

[4] J. Chen and L. M. Chao, "Positioning error analysis for robot manipulators with all rotary joints," in *Proc. IEEE Int. Conf. Robotics and Automation*, San Francisco, CA, 1986.

[5] A. Dainis and M. Juberts, "Accurate remote measurement of robot trajectory motion," in *Proc. IEEE Int. Conf. Robotics and Automation*, St. Louis, MO, 1985.

[6] L. P. Foulloy and R. B. Kelley, "Improving the precision of a robot," in *Proc. IEEE Int. Conf. Robotics*, Atlanta, GA, 1984.

[7] S. A. Hayati, "Robot arm geometric link parameter estimation," in *Proc. 22nd IEEE Conf. Decision and Control*, San Antonio, TX, 1983.

[8] S. A. Hayati and M. Mirmirani, "Improving the absolute positioning accuracy of robot manipulators," *J. Robot. Syst.*, vol. 2, 1985.

[9] S. A. Hayati and G. P. Roston, "Inverse kinematic solution for near-simple robots and its application to robot calibration," in *Recent Trends in Robotics: Modeling, Control, and Education.*, M. Jamshidi, J. Y. S. Luh, and M. Shahinpoor, Eds. New York: Elsevier Science, 1986.

[10] J. M. Hollerbach and D. J. Bennett, "Automatic kinematic calibration using a motion tracking system," in *Proc. 4th Int. Symp. Robotics Research*, Santa Cruz, CA, August 9–14, 1987.

[11] A. Izaguirre, J. Summers, and P. Pu, "A new development in camera calibration, calibrating a pair of mobil TV cameras," to appear in *Int. J. Robotics Res.*

[12] R. P. Judd and A. B. Knasinski, "A technique to calibrate industrial robots with experimental verification," in *Proc. IEEE Int. Conf. Robotics and Automation*, Raleigh, NC, 1987.

[13] R. Lenz and R. Tsai, "Techniques for calibration of the scale factor and image center for high accuracy 3D machine vision metrology," in *Proc. IEEE Int. Conf. Robotics and Automation*, Raleigh, NC, 1987. Also, *IEEE Pattern Anal. Machine Intell.*, vol. PAMI-9, Sept. 1987.

[14] B. W. Mooring and G. R. Tang, "An improved method for identifying the kinematic parameters in a six-axis robot," in *ASME Proc. Int. Computers in Engineering Conf.*, Las Vegas, NV, 1984.

[15] G. Puskorius and L. Feldkamp, "Calibration of robot vision," in *Proc. IEEE Int. Conf. Robotics and Automation*, Raleigh, NC, 1987.

[16] ——, "Camera calibration methodology based on linear perspective transformation error model," in *Proc. IEEE Int. Conf. Robotics and Automation*, Raleigh, NC, 1988.

[17] D. F. Rogers and J. A. Adams, *Mathematical Elements for Computer Graphics*. New York: McGraw-Hill, 1976.

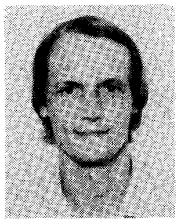
[18] R. Tsai, "A versatile camera calibration technique for high accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses," *IEEE J. Robotics Automat.*, vol. RA-3, no. 4, Aug. 1987; a preliminary version appeared in *Proc. 1986 IEEE Int. Conf. Computer Vision and Pattern Recognition*, Miami, FL, June 22–26, 1986.

[19] R. Tsai and R. Lenz, "Review of the two-stage camera calibration technique plus some new implementation tips and new techniques for center and scale calibration," in *Proc. Second Topical Meeting Machine Vision*, Opt. Soc. Amer., Lake Tahoe, Mar. 18–20, 1987.

[20] —, "A new technique for fully autonomous and efficient 3D robotics hand-eye camera," in *Proc. 4th Int. Symp. Robotics Research*, Santa Cruz, CA, Aug. 9–14, 1987. Also, *IEEE Robotics Automat.*, Summer, 1989.

[21] ——, "Real time versatile robotics hand/eye calibration using 3D machine vision," in *Proc. Int. Conf. Robotics and Automation*, Philadelphia, PA, Apr. 24–29, 1988.

[22] D. E. Whitney, C. A. Lozinski, and J. M. Rourke, "Industrial robot forward calibration method and results," *J. Dynamic Syst., Meas., Contr.*, 1986.



Reimar K. Lenz was born in Aachen, West Germany on January 10, 1956. He received the B.S. degree from the Technical University of Stuttgart in 1977, and the M.S. and Ph.D. degrees from the Technical University of München in 1980 and 1986, respectively, all in electrical engineering. His Ph.D. dissertation was on fast algorithms to estimate geometric transformations in image sequences.

Since 1980 he has been a Research Assistant at the Lehrstuhl für Nachrichtentechnik in Munich, interrupted by research assignments at the Lehrstuhl für theoretische Nachrichtentechnik in Hannover in 1981 (Data Compression, 64kbit/s TV) and at the IBM Research Labs in Yorktown Heights, NY, in 1986–1987 (manufacturing research, robot vision, real-time vision). His current research interests are camera calibration and pattern recognition.

Roger Y. Tsai (M'82) received the M.S. degree from Purdue University, West Lafayette, IN, in 1980, and the Ph.D. degree from the University of Illinois at Urbana-Champaign in 1981, both in electrical engineering.

He was employed by Bell-Northern Research/INRS-Telecommunications, Montreal, P.Q., Canada, for three months during the summer of 1979 as a Visiting Scientist, working on moving image registration and enhancement. During the summer of 1980, he was employed by the Signal Processing Group, EPFL, Lausanne, Switzerland, for three months working on 3-D time-varing scene analysis. In the summer of 1981, he again visited BNR/INRS, Montreal, for three months, working on image sequence analysis and computer vision. He is now with IBM Thomas J. Watson Research Center, Yorktown Heights, NY. His major research interests include stereo matching and 3-D measurement, 3-D range sensing, image feature extraction, SEM metrology, 3-D robotics and geometric vision, sensor planning, modeling and calibration.

Dr. Tsai was the recipient of the Best Paper Award for 1986 IEEE International Conference on Computer Vision and Pattern Recognition (CVPR) and the 1986, 1987, and 1988 IBM External Honor Recognition. He has served as Chairman of the Computer Vision committee for the IEEE Society of Robotics and Automation for 1988 and 1989. He has served as Session Chairman for 1986 and 1989 IEEE International Conference on CVPR, 1987 and 1988 IEEE International Conference on Robotics and Automation. He is on the program committee for 1989 IEEE CVPR conference. He is the Technical Editor of *Robot Vision and Inspection Systems* and IEEE TRANSACTIONS ROBOTICS AND AUTOMATION. He is also on the editorial board for Robotics Review, MIT Press.