A New Technique for Fully Autonomous and Efficient 3D Robotics Hand/Eye Calibration

ROGER Y. TSAI AND REIMAR K. LENZ

Abstract—**This paper describes a new technique for computing 3D position and orientation of a camera relative to the last joint of a robot manipulator in an eye-on-hand configuration. This is part of a trio for real-time 3D robotics eye, eye-to-hand, and hand calibrations, 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 movements 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. A series of generic geometric properties or lemmas are presented, leading to the derivation of the final algorithms, which are aimed at simplicity, efficiency, and accuracy while giving ample geometric and algebraic insights. Besides describing the new technique, critical factors influencing the accuracy are analyzed, and procedures for improving accuracy are introduced. Test results of both simulation and real experiments on an IBM Cartesian robot are reported and analyzed.**

I. Introduction

A. The Calibration Trio

IN ORDER for a robot to use a video camera to estimate the 3D 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 three techniques to deal with these three tasks. They are as follows:

1) Camera Calibration (see [6], [10], [11], [13]).
2) Robot Eye-to-Hand Calibration (this paper).
3) Cartesian Robot Hand Calibration [5].

Manuscript received December 16, 1987; revised December 6, 1988. Part of the material in this paper was presented at the 4th Int. Symp. on Robotics Research, Santa Cruz, CA, Aug. 9–14, 1987.

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

R. K. Lenz was with the IBM Thomas J. Watson Research Center, Yorktown Height, NY 10598. He is now with Lehrstuhl für Nachrichtentechnik, Technische Universität München, D-8000, München 2, West Germany.

IEEE Log Number 8927074.

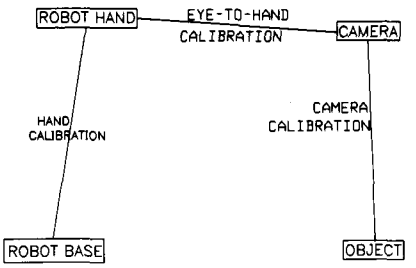


Fig. 1. To obtain the 3D position and orientation of an object relative to the robot world base, it is necessary to perform three calibrations; namely, robot hand, eye-to-hand, and eye (camera) calibration.

The advantages of the techniques are as follows:

1) They are faster than any other vision-based calibration technique by at least an order of magnitude.

The camera calibration takes only 25 ms on a 68000-based minicomputer plus 65 ms to read the image into the computer and extract the image feature coordinates (centers of 36 circular discs) with better than 1-μm accuracy in the image space.[1] For the other two calibrations, the robot makes a series of moves, and at the end of each move, a camera calibration is performed. At the end of all the moves, a few milliseconds are needed to finalize the calibration trio.

2) The techniques are at least as accurate as any existing technique using vision.

3) The three calibrations share the following: setup, calibration plate, image feature extraction procedure, definition of symbols and matrix representations, robot motion, and processing equipment.

4) The calibration need not be 3D. It can be coplanar. This makes the construction of high-accuracy target points possible (see the next section for the description of targets).

5) They are friendly to machine vision people and do not require special skills.

This paper describes the second calibration procedure.

B. 3D Robotics Hand/Eye Calibration

3D robotics hand/eye calibration is the task of computing the relative 3D position and orientation between the camera and the robot gripper in an eye-on-hand configuration, meaning that the camera is rigidly connected to the robot gripper. The camera is either grasped by the gripper, or just fastened to it. More specifically, this is the task of computing the rel-

[1] When doing feature extraction, off-the-shelf general-purpose image processing hardware boards were used for frame grabbing, thresholding, and boundary location estimation.

1042-296X/89/0600-0345$01.00

ative rotation and translation (homogeneous transformation) between two coordinate frames, one centered at the camera lens center, and the other at the robot gripper. The gripper coordinate frame is centered on the last link of the robot manipulator, and as we shall see in this paper, the robot manipulator must possess enough degrees of freedom so as to be able to rotate the camera around two different axes while at the same time keeping the camera focused on a stationary calibration object in order to resolve uniquely the full 3D geometric relationship between the camera and the gripper.[2]

C. The Difficulties of the Problem

It is obvious that if the robot knows the exact 3D positions of a number of points on a calibration setup in the robot world coordinate system as well as the 3D location of the gripper, while at the same time, the camera can view these points in a proper way, then it is possible to determine the 3D homogeneous transformation between the camera and the calibration world coordinate frame, making it a trivial matter to compute the homogeneous transformation between the camera and the manipulator. However, it is very difficult, if at all possible, for the robot to acquire accurate knowledge of the 3D positions of a number of feature points easy enough for the camera to view simultaneously with the right resolution, field of view, etc., while the position information has to be known in the robot world coordinate system. Some researchers treat the difference between the calibration world coordinate system and the robot world coordinate system as a 6-degree-of-freedom unknown, and incorporate them into a much larger nonlinear optimization process (see "The Reasons why the State-of-the-Art Is Deficient" later in the paper). We propose a much easier and faster approach.

D. The Importance of 3D Hand/Eye Calibration

The calibration is important in several aspects:

Automated 3D Robotics Vision Measurement: When vision is used to measure 3D geometric relationships between different parts of an object in a robotics work cell, it is often necessary to use the manipulator to move the vision sensor to different positions in the work space in order to see different features of the object (see [15]). At each point, the 3D position and orientation of the feature measured by the vision system is only relative to the vision sensor. As the manipulator moves the sensor to different positions, the measurements taken at different positions are not related to one another unless we know the 3D relative position and orientation of the sensor at different locations. If the robot system is capable of knowing where the gripper is in the robot world coordinate system to some degree of accuracy, then it should know how much 3D motion it has undertaken from one position to another. Since the camera is rigidly connected to the gripper, of course it also undergoes the same rigid body motion, *but only* in the robot world coordinate system. If the hand/eye calibration is not done, one does not know the 3D homogenous transformation between the camera 3D coordinate systems at different locations simply from the motion of the robot manipulator.

Automated Sensor Placement Planning: In order to do automated 3D measurement with robot vision, sensor planning is vital in order to automatically determine the optimum positions of the sensor so that all the desired features can be viewed while taking care of problems of occlusion, depth of focus, resolution, field of view, etc. However, even if the robot knows where to put the sensor for optimum viewing, it does not know where the manipulator should be in order to achieve this goal, unless the 3D geometric relationships between the last link of the manipulator and the sensor are known.

Automated Part Acquisition or Assembly: When vision is used to aid the robot in grasping an object for automated assembly or part transport with eye-on-hand configuration, unless iterative visual feedback is used, the vision system may be able to determine where the part is relative to the sensor, but the robot does not know how to place the manipulator to grasp it. This problem can be resolved by performing robot hand/eye calibration.

Stereo Vision: If only one camera is used to do stereo vision, one way to create a stereo base is to move the camera with the manipulator.[3] Although the robot system may know how much the manipulator has moved, it does not know the homogeneous transformation between the 3D camera coordinate system, even if the camera undergoes the same rigid body motion as the gripper does (since the rigid body motion is defined only in the robot world coordinate system). Again, when the hand/eye calibration is performed, this problem is solved.

E. The Reasons why the State-of-the-Art is Deficient

From our literature survey, there are two categories of approaches for doing robot hand/eye calibration:

Coupling Hand/Eye Calibration with Conventional Robot Kinematic Model Calibration: References (partial list): [1], [3], [7]. In this approach, global nonlinear optimization is done over the robot kinematic model parameters an the hand/eye parameters simultaneously, making the number of unknowns generally over 30. Such large-scale nonlinear optimization is very time-consuming, and needs a very good initial guess and accurate data for convergence. It also cannot easily exploit the use of redundant images and stations for reducing error since the computation would become prohibitive.

Decoupling Hand/Eye Calibration from Conventional Robot Kinematic Model Calibration: References (partial list): [9], [12], this paper. As far as we know, Shiu and Ahmad's work [9] and the work reported in Tsai and Lenz [12] as well as this paper are the first attempts to decouple the hand/eye calibration from robot model calibration and not use global high-dimensional nonlinear optimization. The starting point in these two works are similar (although independently developed), the solutions are very different. In Shiu and Ahmad's method, the number of unknowns to solve for is twice the number of degrees of freedom, since they treat sin and cos functions as independent. We found it advantageous to use

[2] It takes at least two rotary joints and one linear joint, or three rotary joints. It is possible to use just two rotary joints, but the rotation axes for these two joints must coincide at the calibration block.

[3] This is not highly recommendable except in low accuracy applications. It is better for the robot to carry a stereo pair of cameras or a laser-camera pair, or to use one camera with model-based location determination.

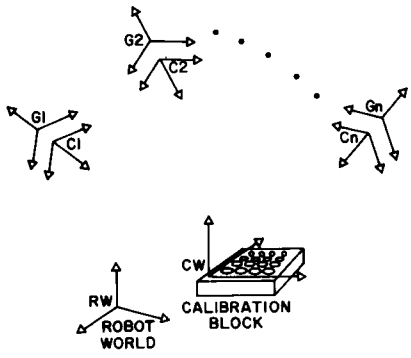


Fig. 2. Basic setup for robot hand/eye calibration. C_i and G_i are coordinate frames for the camera and gripper, respectively.

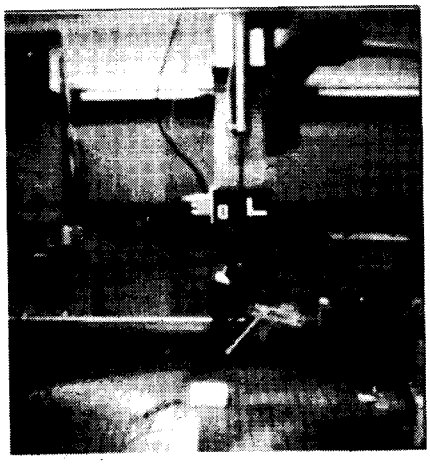


Fig. 3. The physical setup. A CCD camera is rigidly mounted on the last joint of an IBM Clean Room Robot for performing hand/eye calibration.

redundant frames to improve accuracy, but in our algorithm, the number of unknowns stays the same no matter how many frames are used simultaneously, and for each additional frame, only 60 additional arithmetic scalar operations are needed (each operation takes less than half a microsecond on a typical minicomputer). In Shiu and Ahmad's method, the number of unknowns increases by two for each extra frame. Our procedure is simpler and faster, and the derivation procedure is also simpler. We have also done extensive error analysis, simulation and real experiments for testing the accuracy potential or problems of hand/eye calibration, and propose means for improving accuracy.

II. The New Approach

A. Basic Setup

Fig. 2 is a schematic depiction of the basic setup. Figs. 3 and 4 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 pause of each motion. The calibration object 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 (see [6], [10], [11], [13]). A detailed description of the setup can be found in Section IV. The following is a list of definitions for the various coordinate frames. (Note: All coordinate frames mentioned here are Cartesian coordinate frames in 3D):

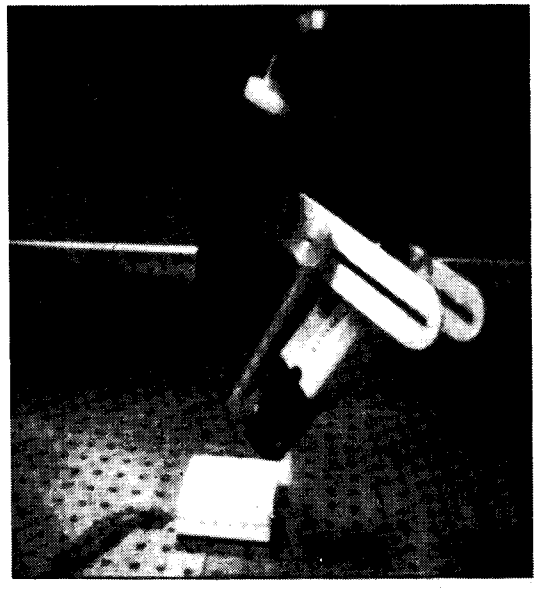


Fig. 4. The physical setup. A CCD camera is rigidly mounted on the last joint of an IBM Clean Room Robot for performing hand/eye calibration.

- G_i: The gripper coordinate system. 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.
- 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. It is fixed in the robot work station, and as the robot arm moves around, the encoder output of all the robot joints enables the system to tell where the gripper is relative to RW.

Definition of a List of Homogeneous Transformation Matrices:

H_{gi} defines coordinate transformation from G_i to RW (1)

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

H_{ci} defines coordinate transformation from CW to C_i (2)

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

H_{gij} defines coordinate transformation from G_i to G_j (3)

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

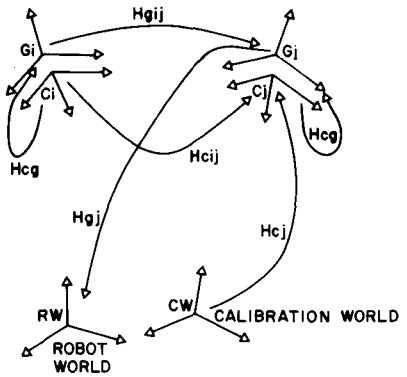


Fig. 5. Relationship between the homogeneous matrices and the coordinate frames.

H_{cij} defines coordinate transformation from C_i to C_j (4)

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

H_{cg} defines coordinate transformation from C_i to G_i (5)

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

In the preceding equations, i, j range from 1 to N, where N is the number of stations in Fig. 2 where the camera grabs pictures of the calibration block. Fig. 5 illustrates the relationship between the homogeneous matrices and the various coordinate frames in Fig. 2. 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.

from G_j to RW, obviously

$$H_{gij} = H_{gj}^{-1} H_{gi}. \tag{6}$$

Similarly

$$H_{cij} = H_{cj} H_{ci}^{-1}. \tag{7}$$

Notice that (6) and (7) are incompatible in terms of where the inverse signs are placed. This is due to the fact that H_{gi} is from G_i to RW while H_{ci} is from CW to C_i.

Final: H_{cg}

Notice that if RW coincided with CW, it would be trivial to compute H_{cg}, which in this case would be equal to $H_{gi}^{-1} H_{ci}^{-1}$. However, it is very difficult for the coordinate system on the calibration block to be set in a fixed and precisely known 3D relationship with respect to the robot coordinate system such that the positions of all the points on the calibration block are known relative to the robot.

a) Some basic background for a general rotation matrix and its real eigenvectors: Before describing the new technique, we introduce the representation for transformation (in particular, rotations) used below. It is well known [8] that any rigid body motion or Cartesian coordinate transformation can be modeled as a rotation by an angle θ around an axis through the origin with direction cosines n_1, n_2, n_3, followed by a translation T such that

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = R \begin{bmatrix} x \\ y \\ z \end{bmatrix} + T \quad \text{or} \quad \begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = H \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

where (x, y, z) and (x', y', z') are the coordinates of any point before and after the transformation. R is a 3×3 orthonormal matrix of the first kind (i.e., det $(R) = 1$)

$$R = \begin{bmatrix} n_1^2 + (1 - n_1^2)\cos\theta & n_1 n_2 (1 - \cos\theta) - n_3 \sin\theta & n_1 n_3 (1 - \cos\theta) + n_2 \sin\theta \\ n_1 n_2 (1 - \cos\theta) + n_3 \sin\theta & n_2^2 + (1 - n_2^2)\cos\theta & n_2 n_3 (1 - \cos\theta) - n_1 \sin\theta \\ n_1 n_3 (1 - \cos\theta) + n_2 \sin\theta & n_2 n_3 (1 - \cos\theta) + n_1 \sin\theta & n_3^2 + (1 - n_3^2)\cos\theta) \end{bmatrix}. \tag{8}$$

B. What Are the Observables and What is to be Computed?

1) The Observables or the Measurables: The observables are H_{ci} and H_{gi} for $i = 1, \cdots, N$. H_{ci} is obtained from computing the extrinsic calibration parameters (see Tsai 1986, [10], [11], [13]) using the image grabbed at the ith pause of robot movement. It defines the relative 3D rotation and translation from CW to C_i. For 36 calibration points, it takes about 20 ms to compute, and is accurate to one part in 4000. The other set of observables is the H_{gi}'s. Any robot that can supply the information of where the gripper is within the robot workstation is capable of delivering H_{gi}. This requires good robot calibration. Actually, even if H_{gi} may be bad, so long as H_{gij} is good, there is no problem. This is due to the fact that the computational procedure entails only H_{gij}, but not H_{gi}.

2) Elements to be Computed:

Intermediate: H_{gij}, H_{cij}

Since H_{gi} defines transformation from G_i to RW, and H_{gj}

H is the homogeneous transformation matrix (used in (1)–(5)), and is defined as

$$H = \begin{bmatrix} & R & & T \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

One of the eigenvector and eigenvalue of R must be the rotation axis and 1, respectively, since by definition, R is a rotation around axis $P_r \equiv [n_1\ n_2\ n_3]^T$, and obviously

$$RP_r = P_r$$

thus P_r is an eigenvector (or principal vector) and its corresponding eigenvalue is 1. From (8) and by definition, it is obvious that specifying P_r and θ completely specifies R. Therefore, it is quite convenient to represent R by P_r scaled some function of θ. We use a modified version of Rodrigues formula (see [4]), and define P_r as

$$P_r = 2 \sin \frac{\theta}{2} [n_1\ n_2\ n_3]^T, \qquad 0 \le \theta \le \pi. \tag{9}$$

Besides the advantages associated with quaternions or other vector representation of rotation matrix, one advantage for us is that some error formulas hold true even for noninfinitesimal perturbations. For example, Lemma V in Section III-A is exact. Also, the error formula in (29) is simpler. Another obvious advantage is that R is a simple function of P_r without any trigonometric functions

$$R = \left(1 - \frac{|P_r|^2}{2}\right) I + \frac{1}{2}(P_r P_r^T + \alpha \cdot \text{Skew}(P_r)) \quad (10)$$

where $\alpha = \sqrt{4 - |P_r|^2}$ and Skew (P_r) is defined in (11e). For the rest of the paper, the principal axis is defined as such, and all the computational procedures are given for P_r explicitly, and not for R.

b) Computational procedures and conditions for uniqueness: We first give the computational procedures and conditions of uniqueness before we derive them. The derivations and proofs follow from the eleven properties or lemmas in Section II-D. The actual proof for those eleven lemmas will be published in a later paper that will contain a fuller account of the work. The minimum number of stations is three, where station means the location where the robot pauses for doing camera extrinsic calibration. Using more than three stations improves the accuracy, as will be seen in Section III-A.

Some Definition of Notation:

θ_R: Angle of rotation for R.

P_{gij}: Principal axis or rotation axis for R_{gij} defined in (3), which is the 3D rotation from gripper coordinate frame G_i to G_j, as defined in (3). (11a)

P_{cij}: Rotation axis for R_{cij} in (4). (11b)

P_{cg}: Rotation axis for R_{cg}. (11c)

$$P'_{cg} = \frac{1}{2\cos\left(\frac{\theta_{R_{cg}}}{2}\right)} P_{cg} = \frac{1}{\sqrt{4 - |P_{cg}|^2}} P_{cg}. \quad (11d)$$

Skew (V): A skew-symmetric matrix generated by a 3D vector V such that

$$\text{Skew}\ (V) = \begin{bmatrix} 0 & -v_z & v_y \\ v_z & 0 & -v_x \\ -v_y & v_x & 0 \end{bmatrix}. \quad (11e)$$

N: Number of stations described in Section II-A.

Notice that the vectors defined above will also be used as a 3×1 column matrix. Also note that since P_{gij}, P_{cij}, and P_{cg} are rotation axes with a function of angle as their length, they completely specify R_{gij}, R_{cij}, and R_{cg}. That is why for the procedures in the following, the formula for P_{cg} is given, and not for R_{cg}.

3) Procedure for Computing R_{cg}:

Step 1: Compute P'_{cg}: For each pair of stations i, j such that the rotation angle R_{gij} or R_{cij} is as large as possible (Fig. 6 illustrates a good way to select the pairing. See also section on test results), set up a system of linear equations with P'_{cg} as the unknown

$$\text{Skew}\,(P_{gij} + P_{cij}) P'_{cg} = P_{cij} - P_{gij}. \quad (12)$$

Since Skew $(P_{gij} + P_{cij})$ is always singular, it takes at least two pairs of stations to solve for a unique solution for P'_{cg} using linear least squares technique.

Exception handling: If $P_{gi1j1} + P_{ci1j1}$ is colinear with $P_{gi2j2} + P_{ci2j2}$ while P_{gi1j1} is not colinear with P_{gi2j2}, then the rotation angle of R_{cg} must be 180° and the rotation axis the same as $P_{gi1j1} + P_{ci1j}$.

Step 2: Compute $\theta_{R_{gc}}$:

$$\theta_{R_{gc}} = 2 \tan^{-1} |P'_{cg}|. \quad (13)$$

Note: Step 2 is not quite necessary since P_{cg} in Step 3 is sufficient to represent rotation. However, (13) may be handy.

Step 3: Compute P_{cg}:

$$P_{cg} = \frac{2P'_{cg}}{\sqrt{1 + |P'_{cg}|^2}}. \quad (14)$$

4) Procedure for Computing T_{cg}: Given at least two pairs of stations i, j, set up a linear systems of three linear equations with T_{cg} as unknowns

$$(R_{gij} - I) T_{cg} = R_{cg} T_{cij} - T_{gij}. \quad (15)$$

For at least two pairs of stations, two sets of (15) are established and can be solved for the common unknowns T_{cg} using linear least squares solutions.

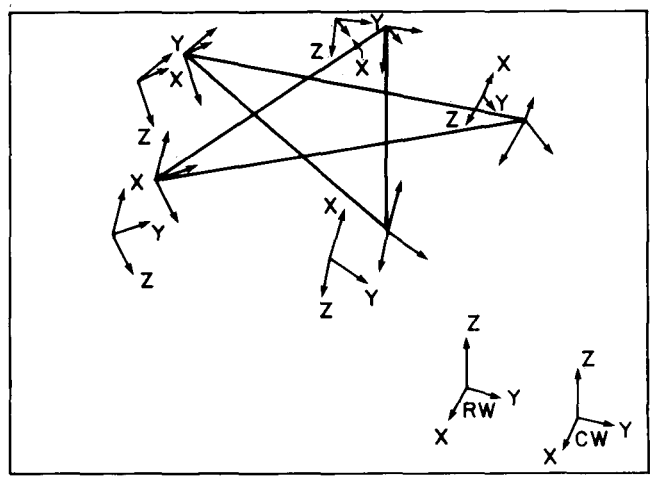


Fig. 6. Pairs of stations should be selected such that the interstation angle is as large as possible, and the angle between different interstation rotation axes are as large as possible. The bar between stations denotes a particular selection of a pair of stations.

C. Speed Performance

After the robot finishes the movement and grabbing the images, it takes only about $100 + 60N$ arithmetic operations to complete the computation. For a typical minicomputer, this only takes about 1/2 ms for ten stations. This complexity figure $(100 + 60N)$ can be derived as follows: The majority

of the computation is for solving the overdetermined linear least squares solutions of (12) and (15). It takes about $3 \times N \times 3^2$ to form the normal equation of either one of (11e) and (16), and $3^3 \times 2$ to solve the 3×3 normal equation. With a minimum of three stations and two interstation pairs, it takes about 1/10 ms. This is negligible compared with the robot movement and image acquisition and analysis; at the pause of each movement, it takes about 90 ms to grab an image, extract all the 36 feature point coordinates with high accuracy, and compute the extrinsic camera parameters defined in (4).

D. Derivations of Computational Procedures and Conditions of Uniqueness using Eleven Lemmas

In order to outline the derivations of the computational procedures without going into actual details, eleven lemmas will first be stated and the significance of each explained. Selected sets of the key lemmas will be proved. Then the proof for the computational procedure for R_{cg} and T_{cg} will be given, followed by the conditions for uniqueness which will be stated and proved.

Lemma I: R_{gij} and R_{cij} differ by a unitary similarity transformation

$$R_{gij} = R_{cg} R_{cij} R_{cg}^T. \tag{16}$$

Proof: This follows easily from the fact that H_{cg}, H_{gij}^{-1}, H_{cg}^{-1}, H_{cij} in Fig. 5 form a closed loop and thus their product equals identity.

Significance: As a result, the eigenvector matrix of R_{gij} can be transformed from that of R_{cij} using R_{cg}.

Lemma II: R_{cg} rotates the rotation axis of R_{cij} into that of R_{gij}, or

$$P_{gij} = R_{cg} P_{cij}. \tag{17}$$

Proof: This follows from expanding R_{cij} and R_{gij} in (16) by their associated eigenvector and eigenvalue matrices, and making use of the fact that P_{gij} and P_{cij} are the only real eigenvectors of R_{gij} and R_{cij}, respectively, and that the resultant rotation matrices on the left- and right-hand sides of (16) have a common real eigenvector.

Significance: Since, from Section II-B1, P_{cij} and P_{gij} can be readily available from the observables H_{cij} and H_{gij}, (17) establishes constraints on R_{cg} in order to solve for it. Lemma II also says that if we regard all P_{cij} and P_{gij} as two clusters of vectors or points, then R_{cg} transforms one cluster into another.

Lemma III: The rotation axis of R_{cg} is perpendicular to the vector joining the ends of the rotation axes for R_{cij} and R_{gij}, or

$$P_{cg} \perp (P_{gij} - P_{cij}). \tag{18}$$

Proof: This can be seen by observing Fig. 7, but algebraically, here is the proof (we will omit the subscript ij for clarity): The purpose is to show that $(P_g - P_c)^T P_c = 0$. By making use of (17) and the fact that $R_{cg} P_{cg} = P_{cg}$, we have

$$\begin{aligned}(P_g - P_c)^T P_{cg} &= (P_g - P_c)^T R_{cg}^T R_{cg} P_{cg} \\ &= (R_{cg} R_g - P_g)^T P_{cg} \\ &= [(R_{cg} - I) P_g]^T P_{cg} \\ &= P_g^T (R_{cg}^T - I) P_{cg} = 0.\end{aligned}$$

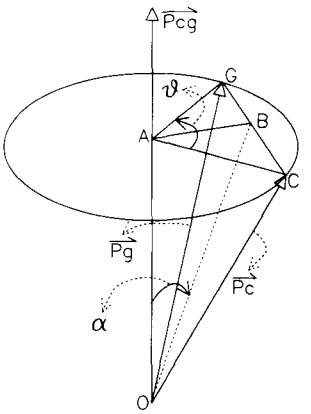


Fig. 7. Geometrical relationships between P_{cg}, P_c, and P_g. P_{cg} rotates P_c into P_g. The plane containing the circle is perpendicular to P_{cg}, and point B is the midpoint of point C and G.

Significance: This implies that for a given pair of distinct P_{gij} and P_{gij}, P_{cg} is confined to be in the bisecting plane of P_{cij} and P_{gij}. With two such pairs, the direction of P_{cg} can be determined. In fact, (18) implies that P_{cg} can be determined up to a scale factor s from

$$P_{cg} = s(P_{gi_1 j_1} - P_{ci_1 j_1}) \times (P_{gi_2 j_2} - P_{ci_2 j_2}). \tag{19}$$

However, we will not use Lemma III in this manner. Instead, Lemma III is used to build up the procedure for computing R_{cg} via Lemmas IV, V, and VI. The reason is that (19) is more error-sensitive and has unnecessary degeneracies due to the fact that angle is not considered jointly.

Lemma IV: $P_{gij} - P_{cij}$ is colinear with $(P_{gij} + P_{cij}) \times P_{cg}$.

Proof: This follows from the fact that $P_{gij} - P_{cij}$ is simultaneously orthogonal to P_{cg} (according to Lemma III) and to $P_{gij} + P_{cij}$ (this latter property can be easily proved).

Significance: This says that $P_{gij} - P_{cij} = s(P_{gij} + P_{cij}) \times P_{cg}$ for some scale factor s. Lemma V forces s to be 1. This lemma makes use of Lemma III, but the formula it generates is more accurate and robust than (18) (coming out of Lemma III) so far as computing P_{cg} is concerned, as will be seen.

Lemma V: $P_{gij} - P_{cij}$ and $(P_{gij} + P_{cij}) \times P'_{cg}$ have the same length, where P'_{cg} was defined in (11d).

Proof: Again, in the following proof and in Fig. 7, we will drop the subscript ij for P_{gij} and P_{cij}. Let the angle between the vectors represented by P'_{cg} and $P_g + P_c$ be α. Then by definition

$$|(P_g + P_c) \times P'_{cg}| = |P_g + P_c| |P'_{cg}| \sin \alpha.$$

Substituting (9) and (11d) into the above gives

$$\begin{aligned}|(P_g + P_c) \times P'_{cg}| &= |P_g + P_c| 2 \sin \frac{\theta}{2} \\ &\quad \cdot \left(4 - 4 \sin^2 \frac{\theta}{2}\right)^{-1/2} \sin \alpha \\ &= |P_g + P_c| \tan \frac{\theta}{2} \sin \alpha \\ &= 2|\overline{OB}| \sin \alpha \tan \frac{\theta}{2} \\ &= 2|\overline{AB}| \tan \frac{\theta}{2} = 2|\overline{CB}| \\ &= |\overline{CG}| = |P_g - P_c|.\end{aligned}$$

In the above derivations, A, B, C, G, O are points in Fig. 7, and $|\overline{OB}|$ means length of the vector extending from point O to point B, etc. Also, in the above derivations, several geometric and trigonometric relationships in Fig. 7 are used. One is $|P_g + P_c| = 2|\overline{OB}|$. Another is $|\overline{AB}| = |\overline{OB}| \sin \alpha$. Still another is $|\overline{CB}| = |\overline{AB}| \tan \theta/2$. These properties follow easily from Fig. 7. Thus we have shown that

$$|(P_g + P_c) \times P'_{cg}| = |P_g - P_c|.$$

Significance: Given Lemmas IV and V, Lemma VI is readily derived, which easily leads to the computational procedure for P_{cg} in (12).

Lemma VI:

$$(P_{gij} + P_{cij}) \times P'_{cg} = P_{cij} - P_{gij}. \tag{20}$$

Proof: This is a direct consequence of Lemmas IV and V.

Significance: Although (18) provides a constraint on the direction of P_{cg} for any pair of stations ij, (20) provides a stronger constraint since it constrains $\theta_{R_{cg}}$ as well as P_{cg}.

Lemma VII: Skew $(P_{gij} + P_{cij})$ is singular and has rank 2.

Significance: Skew $(P_{gij} + P_{cij})$ is the coefficient matrix for the systems of linear equations in (12) used to solve for P'_{cg}. Therefore, Lemma VII implies that it is impossible to compute R_{cg} with only two stations.

Lemma VIII:

$$(R_{gij} - I)T_{cg} = R_{cg}T_{cij} - T_{gij}. \tag{21}$$

Proof: This follows from the same derivation as Lemma I.

Significance: Lemma VIII establishes the equation in (15) used to solve for T_{cg}.

Lemma IX: $R_{gij} - I$ is singular and has rank 2.

Significance: $R_{gij} - I$ is the coefficient matrix for the systems of linear equations in (15) used to solve for T_{cg}. Therefore, Lemma IX implies that it is impossible to compute T_{cg} with only two stations.

Lemma X: If $\theta_{R_{cg}} \neq \pi$, or equivalently, $|P_{cg}| \neq \pm 2$, then

$$\begin{bmatrix} \text{Skew } (P_{gi_1j_1} + P_{ci_1j_1}) \\ \text{Skew } (P_{gi_2j_2} - P_{ci_2j_2}) \end{bmatrix} \tag{22}$$

has full column rank if and only if $P_{gi_1j_1}$ and $P_{gi_2j_2}$ have different directions (or equivalently, $P_{ci_1j_1}$ and $P_{ci_2j_2}$ have different directions).

Significance: Expression (22) is just the compound matrix of two Skew $(P_{gij} + P_{cij})$ in Lemma VII, and therefore is the coefficient matrix for solving R_{cg} given two pairs of P_{gij} and P_{cij}. Thus Lemma X ensures that given a minimum of three stations, the solution for R_{cg} is unique.

Lemma XI:

$$\begin{bmatrix} R_{gi_1j_1} - I \\ R_{gi_2j_2} - I \end{bmatrix} \tag{23}$$

has full column rank if and only if $P_{gi_1j_1}$ and $P_{gi_2j_2}$ have different directions (or equivalently, $P_{ci_1j_1}$ and $P_{ci_2j_2}$ have different directions).

Significance: Expression (23) is just the compound matrix of two $R_{gij} - I$ in Lemma IX, and therefore is the coefficient matrix for solving T_{cg} given two pairs of P_{gij} and P_{cij}. Thus Lemma XI ensures that given a minimum of three stations, the solution for T_{cg} is unique.

Proof of the Computational Procedure for R_{cg} in (12)–(14): Equation (12) follows from Lemma VI by considering the fact that for any two 3×1 vectors a and b

$$a \times b = \text{Skew}(a)\, b \tag{24}$$

where a and b on the left denotes vectors while a and b on the right are 3×1 column matrices. Equations (13) and (14) simply follow from the definitions of P'_{cg} in (11d).

Proof of the Computational Procedure for T_{cg} in (15): This follows simply from Lemma VIII.

Minimum Number of Stations: Three. This follows from Lemmas VII, IX, X, and XI. Equivalently, the minimum number of pairs of stations needed is two.

Conditions of Uniqueness: For a minimum of three stations (or two pairs of stations), the necessary and sufficient condition for a unique solution for R_{cg} and T_{cg} is that the interstation rotation axes are not colinear for different pairs of stations.

Proof: This follows from Lemmas X and XI. Note that when the sum of rotation axes $(P_{gij} + P_{cij})$ are colinear while P_{gij} is different for different interstation rotations, then the solution is still unique except that (12) cannot be used. In this case, angle (R_{cg}) is simply 180° and the rotation axis is the same as $P_{cij} + P_{gij}$.

III. Accuracy Issues

In the following, error analysis will first be given. Then, as a result of error analysis, critical factors dominating the error, and steps for improving accuracy will be described.

A. Error Analysis

The purposes of error analysis are as follows:

1) It reveals what the critical factors influencing the accuracy are.

2) It gives rise to various means for improving accuracy.

3) It is essential for accuracy prediction, which is important to model-driven 3D vision planning.

4) It helps to determine whether one has properly implemented the algorithm. If the error is much larger than what the error formula predicts, something in the setup, programs, or system are not in the right order.

In this section, we first give a list of definitions, followed by a list of lemmas used for deriving the final error formula for R_{cg} and T_{cg}. Then the error formula for R_{cij}, T_{cij}, and R_{gij}, T_{gij} due to error of R_{ci}, T_{ci} and R_{gi}, T_{gi} will be given, followed by the error formula of R_{cg} and T_{cg}. Critical factors affecting the accuracy will be discussed in the next section, and test results will follow thereafter.

Definitions:

RMS: Root mean square (or average of the sum of squares).

$\sigma(V)$: RMS of the magnitude of error corrupting a 3D vector V. $\sigma(V)$ and σ_V are equivalent.

$\sigma(R)$: RMS of the magnitude of error of P_R (rotation axis scaled by the rotation angle, see (6), (11a)–(11c)). $\sigma(R)$ and σ_R are equivalent.

Err (V): Maximum magnitude of error corrupting a 3D vector V.

Err (R): Maximum magnitude of error for P_R.

List of Lemmas Leading to the Final Error Formula

Lemma I:

$$P_{\Delta R \cdot R} = P_{\Delta R} + P_R$$

where ΔR is a small perturbation rotation matrix.

Note: Lemma I says that for small error perturbation of rotation, the rotation axes are additive.

Lemma II:

$$\sigma(R_1 R_2) = \sqrt{(\sigma_{R1}^2 + \sigma_{R2}^2)}.$$

Lemma III:

$$\text{Err } (R_1 R_2) = \text{Err } (R_1) + \text{Err } (R_2).$$

Lemma IV:

$$\sigma(\Sigma V_i) = \sqrt{\sum_i \sigma_{Vi}^2}$$

where V_i' are a number of 3D vectors with RMSE σ_{V_i}.

Lemma V:

$$\sigma(R \cdot V) = \sqrt{\frac{2}{3}(\sigma_R |V|)^2 + \sigma_v^2}$$

where R is a rotation matrix with RMSE σ_R and V is a 3D vector with RMSE σ_V.

Lemma VI:

$$\text{Err } \left(\sum V_i\right) = \sum_i \text{Err } (V_i).$$

Lemma VII:

$$\text{Err } (RV) = \text{Err } (R)|V| + \text{Err } (V).$$

The proof for the above Lemmas will be published in a later paper. The following formulas can be easily derived from the above lemmas and the relationships between R_{cij} and R_{ci}, R_{cj} and between T_{cij} and R_{ci}, T_{ci} using (12).

Error of R_{cij} due to Error of R_{ci} and R_{cj}

$$\sigma_{R_{cij}} = \sqrt{\sigma_{R_{ci}}^2 + \sigma_{R_{cj}}^2} \tag{25a}$$

$$\text{Err } (R_{cij}) = \text{Err } (R_{ci}) + \text{Err } (R_{cj}). \tag{25b}$$

Similar formulas hold for R_g.

Error of T_{cij} due to Error of R_{ci} and T_{ci}

$$\sigma_{T_{cij}} = \sqrt{\frac{2}{3}(\sigma_{R_{ci}}^2 + \sigma_{R_{cj}}^2)|T_{ci}|^2 + \sigma_{T_{ci}}^2 + \sigma_{T_{cj}}^2} \tag{26a}$$

$$= \sqrt{\frac{4}{3}\sigma_{R_c}^2 |T_{c_1}|^2 + 2\sigma_{T_c}^2}. \tag{26b}$$

Equation (26b) is a simpler version of (26a) with $\sigma_{R_{ci}}$ and $\sigma_{R_{cj}}$ replaced by σ_{R_c}, etc.

$$\text{Err } (T_{cij}) = [\text{Err } (R_{ci}) + \text{Err } (R_{cj})]|T_{ci}| + \text{Err } (T_{ci}) + \text{Err } (T_{cj}). \tag{26c}$$

Error of T_{gij} due to Error of R_{gi} and T_{gi}

$$\sigma_{T_{gij}} = \sqrt{\frac{2}{3}\sigma_{R_{gj}}^2 |T_{gi} - T_{gj}|^2 + \sigma_{T_{g1}}^2 + \sigma_{T_{g2}}^2} \tag{27a}$$

$$= \sqrt{\frac{2}{3}\sigma_{R_g}^2 |T_{g1} - T_{g2}|^2 + 2\sigma_{T_g}^2} \tag{27b}$$

$$\text{Err } (T_{gij}) = \text{Err } (R_{g2})|T_{g1} - T_{g2}| + \text{Err } (T_{g1}) + \text{Err } (T_{g2}). \tag{27c}$$

Error of R_{cg}

Three-Station Case:

$$\sigma_{R_{cg}} = \frac{1}{\sin[\angle(P_{g12}, P_{g23}]} \cdot \sqrt{\frac{2}{3}(\sigma_{R_{g12}}^2 + \sigma_{R_{c12}}^2)\left[\frac{1}{(\theta_{R_{g12}})^2} + \frac{1}{(\theta_{R_{g23}})^2}\right]} \tag{28a}$$

$$\text{Err } (R_{cg}) = \frac{1}{\sin[\angle(P_{g12}, P_{g23})]} \cdot \sqrt{\left(\text{Err } (R_{g12}) + \text{Err } (R_{c12})\right)\left(\frac{1}{|\theta_{R_{g12}}|} + \frac{1}{|\theta_{R_{g23}}|}\right)} \tag{28b}$$

where $\angle(P_{g12}, P_{g23})$ means the angle between P_{g12} and P_{g23}

$$\theta_{R_{g12}}/2 = 2|P_{g12}| = \text{angle } (R_{g12})/2.$$

Note that $\theta_{R_{gij}} = \theta_{R_{cij}}$ for all ij.

Since it is always easy to have $\theta_{R_{g12}}$ close to $\theta_{R_{g23}}$ (see the arrangement in Section IV), (28a) reduces to

$$\sigma_{R_{cg}} = \sqrt{\frac{4}{3}(\sigma_{R_{g12}}^2 + \sigma_{R_{c12}}^2)} \cdot \frac{1}{\sin[\angle(P_{g12}, P_{g23})]|\theta_{R_{g12}}|} \tag{28c}$$

$$\text{Err } (R_{cg}) = \frac{2}{\sin[\angle(P_{g12}, P_{g23})]} \cdot \frac{\text{Err } (R_{g12}) + \text{Err } (R_{c12})}{|\theta_{R_{g12}}|}. \tag{28d}$$

Redundant Stations:

$$\sigma_{R_{cg}} = \sqrt{(\sigma_{R_{g12}}^2 + \sigma_{R_{c12}}^2)\left(\frac{1}{\lambda_1^2 + \lambda_2^2} + \frac{1}{\lambda_2^2 + \lambda_3^2} + \frac{1}{\lambda_3^2 + \lambda_1^2}\right)} \tag{29}$$

where λ_i's are the singular values of a matrix with the rows being the interstation rotation eigenvectors (using definition in (9)) for the camera. A few facts are worth noting (details and proofs to be published). First is that as the interstation rotation angle increases, λ_i increases linearly, making $\sigma_{R_{cg}}$ inversely proportional to the interstation rotation angle. The second and most important fact is that as the number of interstation rotations N increases λ_i increases by the square root of N, making $\sigma_{R_{cg}}$ inversely proportional to $\sqrt{N}$. Similarly

$$\text{Err}\ (R_{cg}) = (\text{Err}\ (R_{g12}) + \text{Err}\ (R_{c12}) \cdot \sqrt{\frac{1}{\lambda_1^2+\lambda_2^2} + \frac{1}{\lambda_2^2+\lambda_3^2} + \frac{1}{\lambda_3^2+\lambda_1^2}}. \quad (30)$$

Error of T_{cg}

Three-Station Case:

$$\sigma_{T_{cg}} \le \frac{1}{\sqrt{2\left(\sin^2\frac{\theta_{R_{g12}}}{2} + \sin^2\frac{\theta_{R_{g23}}}{2}\right)}} \cdot \left[\frac{16}{9}\frac{\sigma_{R_{g12}}^2 + \sigma_{R_{c12}}^2}{\sin^2[\angle(P_{g12}, P_{g23})]}|T_{c1}|^2 \frac{(\theta_{R_{c12}}^2 + \theta_{R_{c23}}^2)^2}{\theta_{R_{c12}}^2\theta_{R_{c23}}^2} + \text{cond}\ (A)\left(\frac{8}{3}\sigma_{R_c}^2|T_{c1}|^2 + 4\sigma_{T_c}^2 + \frac{2}{3}\sigma_{R_g}^2 \cdot (|T_{g1} - T_{g2}|^2|T_{g2} - T_{g3}|^2) + 4\sigma_{T_g}^2\right)\right]^{1/2} \quad (31)$$

where

$$A = \begin{bmatrix} R_{g12} - I \\ R_{g23} - I \end{bmatrix}$$

and is the coefficient matrix for solving T_{cg} in (15); and cond (A) is the condition number of A and is defined as

$$\text{cond}\ (A) = \| A \| \cdot \| A^{-1} \|.$$

To simplify the formula, we can regard $\theta_{R_{cij}}$ as being close in magnitude, and $|T_{gi} - T_{gj}|$ as being close in magnitude, making (31) somewhat simpler:

$$\sigma_{T_{cg}} \le \frac{1}{2\sin\frac{\theta_{R_{g12}}}{2}}\left[\frac{16}{9}\frac{\sigma_{R_{g12}}^2 + \sigma_{R_{c12}}^2}{\sin^2[\angle(P_{g12}, P_{g23})]}|T_{c1}|^2 + \text{cond}\,(A)\left(\frac{8}{3}\sigma_{R_c}^2|T_{c1}|^2 + 4\sigma_{T_c}^2 + \frac{4}{3}\sigma_{R_g}^2 \cdot |T_{g1} - T_{g2}|^2 + 4\sigma_{T_g}^2\right)\right]^{1/2}. \quad (32)$$

Similarly

$$\text{Err}\,(T_{cg}) \le \frac{1}{\sqrt{2\left(\sin^2\frac{\theta_{R_{g12}}}{2} + \sin^2\frac{\theta_{R_{g23}}}{2}\right)}}\left\{\frac{4}{\sin[\angle(P_{g12}, P_{g23})]}(\text{Err}\,(R_{g12}) + \text{Err}\,(R_{c12})|T_{ci}| + \text{cond}\,(A)[4\,\text{Err}\,(R_c) + 2\,\text{Err}\,(R_{g2})|T_{g1} - T_{g2}| + 4\,\text{Err}\,(T_g)]\right\}^{1/2}. \quad (33)$$

The effect of the number of stations on the error is the same as that for R_{cg}. This is verified by the test results in Section IV.

B. Critical Factors Affecting the Accuracy and Steps in Improving Accuracy

By observing the accuracy formulas for R_{cg} and T_{cg} in Section III-A, the following observations can be made:

Observation 1: The RMS error of rotation from gripper to camera $\sigma_{R_{cg}}$ is inversely proportional to the sine of the angle between the interstation rotation axes.

By observing (28a), it is seen that $\sigma_{R_{cg}}$ is inversely proportional to sin $(\angle(P_{g12}, P_{g23}))$ (which is equal to sin $(\angle(P_{c12}, P_{c23}))$). This is reasonable since, from Lemma II in Section II-D, R_{cg} rotates P_{cij} into P_{gij}. With a minimum of two pairs of ij's, (17) is used to solve for R_{cg}. When $\angle(P_{g12}, P_{g23})$ becomes smaller, P_{g12} becomes closer to P_{g23}, making (17) for each ij more similar to each other, thus causing the equation to be closer to singularity. Alternatively, one can see that the coefficient matrix for solving P'_{cg} (see Lemma VII in Section II-D) becomes singular as $P_{gi_1j_1}$ approaches $P_{gi_2j_2}$ and $P_{ci_1j_1}$ approaches $P_{ci_2j_2}$. In fact, it can be shown that the row vectors of the coefficient matrix in Lemma X lie in two planes, with $P_{gi_1j_1} + P_{ci_1j_1}$ and $P_{gi_2j_2} + P_{ci_2j_2}$ being the normal vectors of the two planes. Thus the greater the difference between $P_{ci_1j_1}$ and $P_{ci_2j_2}$, the closer the two planes are to being orthogonal, making the coefficient matrix more linearly independent.

Observation 2: The rotation and translation error are both inversely proportional to the interstation rotation angle. That is

$$\sigma_{R_{cg}} \propto -1\,\theta_{R_{gij}} \quad \text{and} \quad \sigma_{T_{cg}} \propto -1\,\theta_{R_{gij}}.$$

This can be seen from (28c), (28d), and (31). This is reasonable since R_{cg} is determined solely from P_{cij} and P_{gij}, and the greater $\theta_{R_{cij}}$ and $\theta_{R_{gij}}$ are, the smaller the effect of a small perturbation (with given size $\sigma_{R_{c12}}$, $\sigma_{R_{g12}}$) is on the result.

Observation 3: The distance between the camera lens center and the calibration block has a dominant effect on the translation error.

This comes from (31)–(33). In fact, any of the terms in (31) or (32) involving $|T_{ci}|$ or $|T_{g1} - T_{g2}|$ generates much more error than all other terms in most of the practical setup. For example, if $|T_{c1}|$ is 5 in and the error of interstation rotations is 3 mrad (these are practical figures that one would encounter), then any term in (31) or (32) involving $\sigma_{R_c}|T_{c1}|$ would generate 15-mil error, which is much bigger than those other terms involving σ_{T_c}, which is the error of translation as a result of extrinsic camera calibration. The term involving σ_{T_g}, however, has the potential of being very big, since this depends on the positional accuracy of the robot, which can be bad.

Observation 4: The distance between the robot gripper coordinate centers at different stations is also a critical factor in forming the error of translation. But the distances between different camera stations are not important.

This again comes from (31)–(33). The situation is similar to that described in Observation 3. Notice that $|T_{g1} - T_{g2}|$ is not the distance between gripper tips at different stations. It is the amount of movement of the *robot gripper coordinate center.*

Observation 5: The error of rotation is linearly proportional to the error of orientation of each station relative to the base. The error of translation is approximately linearly proportional to this error of orientation unless the error of robot translational positioning accuracy is big.

This comes from (28c), (28d), and (31)–(33).

It is convenient to define two types of critical factors. One is first-degree, and the other second-degree. The first-degree factor is more dominant in most cases, but sometimes, some second-degree factor ($\sigma_{T_{gi}}$) can be so bad that it becomes dominant.

First-Degree Critical Factors:

1) The angle between different interstation rotation axes (e.g., $\angle(P_{g12}, P_{g23})$). Note: $\angle(P_{g12}, P_{g23}) = \angle(P_{c12}, P_{c23})$.
2) The rotation angle of interstation rotation $\theta_{R_{gij}} (= \theta_{R_{cij}})$.
3) The distance between the camera lens center and the calibration block $|T_{ci}|$, and the distance between the robot-arm coordinate centers at different stations $|T_{g1} - T_{g2}|$.
4) The error of rotation of each station relative to base $\sigma_{R_{gi}}$, $\sigma_{R_{ci}}$, or error of interstation rotation $\sigma_{R_{gij}}$, $\sigma_{R_{cij}}$.

Second-Degree Critical Factor:

1) Error of translation of each station relative to base $\sigma_{T_{ci}}$, $\sigma_{T_{gi}}$.

C. Steps to Improve Accuracy

1) Adopt the setup to be described in Section IV in order to achieve maximum angles between different interstation rotation axes, no matter how many stations are used.

2) Maximize the rotation angle for interstation rotations. This again can be done using the setup mentioned earlier.

3) Minimize the distance between the camera lens center and the calibration block. This requires a small calibration block and suitable optics for short-range viewing.

4) Minimize the distance between the robot arm coordinate centers at different stations. This requires some planning and is robot-dependent.

5) Use redundant stations. The setup described in Section IV is ideal for using as many stations as you wish. Since the extrinsic calibration plus feature extraction can be done within 90 ms when 36 points are used, using more frames poses no problem. The error due to nonsystematic sources will be reduced by a factor of $\sqrt{N}$ where N is the number of stations. (See results in Section IV.)

6) Use camera calibration algorithm setup that yields high accuracy to improve error on translation and rotation of each individual station.

7) Try to precalibrate the robot itself so that the position and orientation of each station is known more accurately. If this is difficult, then at least try to make interstation translation and rotation more accurate, if possible. That is, the robot system may not be able to tell the user the absolute location and orientation of its gripper coordinate frame, it may, however, be able to better tell the amount of relative movement from station to station.

IV. Simulation and Real Experiment Results

A. Simulation Experiments

1) The Station Generation Process: It is important to use a process for simulating the position and orientation of gripper and camera stations that is realistic and easy for controlling the critical parameters of Section III-B in order to see their effect on the final accuracy. It should allow all critical parameters to be in optimum conditions simultaneously. It also serves as a means of planning robot motion in order to generate stations in the real experiments. Fig. 6 illustrates the results of using our process for generating a five-station configuration. The bright coordinate frames are the camera coordinate frames C_i while the darker frames are for the robot gripper coordinate frames G_i. The bars in Fig. 6 indicate the selections of interstation pairs. The station generation process is described as follows: first, set up a calibration block world coordinate frame CW and a robot world coordinate frame RW as in Fig. 6. Next, directly above CW, place a pair of coordinate frames C_0 and G_0 for camera and gripper that maintains a distance of $|T_c|$ from CW (notice that $|T_c|$ is one among the critical factors in Section III-B) with the z axis of C_0 pointing right at CW. C_0 and G_0 are actually not used for computing the results R_{cg} and T_{cg}, but rather for generating other stations. Next, select a number N to be the total number of stations to be generated. Then, generate N stations for camera and gripper by rotating C_0 and G_0 around N axes uniformly distributed with $360/N$ degrees apart, centered at CW and parallel to the xy plane of CW. The interstation pairs are chosen using a star-drawing technique (see Fig. 6). This gives a systematic way of generating an arbitrary number of stations while at the same time allowing one to easily vary the critical parameters for testing error sensitivity.

2) The Control of Critical Parameters as Simulation Input Parameters: All of the critical parameters (first and second degree) listed in Section III-B can be simulated with easy control. The control of each critical parameter is listed in the following:

Interstation rotation angle: This is controlled by varying the rotation angle used in rotating C_0 and G_0 to each individual station.

Angle between different interstation rotation axes: This is controlled actually by varying the number of stations generated. For each case, choose only the first three traversed by doing star drawing. Obviously, the larger the number of overall stations is, the narrower the angle between successive interstation rotation axes is.

Distance between camera and calibration block: This

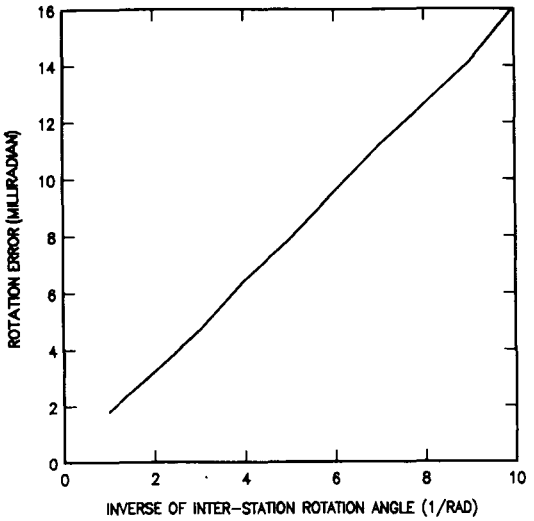


Fig. 8. RMS rotation error as a function of inverse of interstation rotation angle.

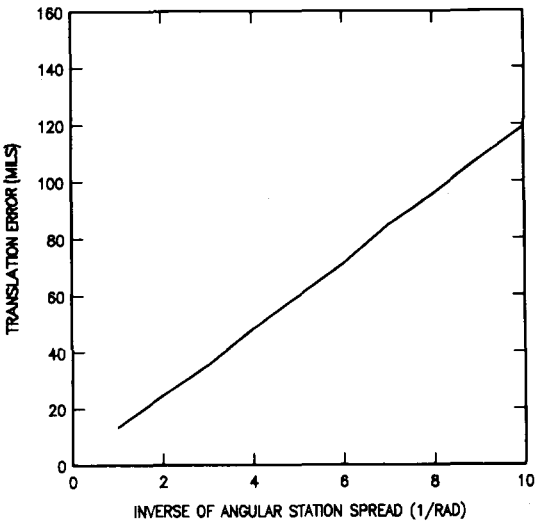


Fig. 9. RMS translation error as a function of inverse of interstation rotation angle.

is controlled by varying the distance $|T_c|$ in the above generation procedure for placing C_0 and G_0.

Number of stations: This is the parameter in the process that is totally arbitrary, except that if it is even, the star drawing is not as straightforward. We always use an odd number of stations.

Rotation and translation error for each station ($\sigma_{R_{ci}}$, $\sigma_{T_{gi}}$, $\sigma_{R_{gi}}$, and $\sigma_{T_{gi}}$): This is an extrinsic calibration error, and can be simulated by perturbing the ideal homogeneous transformation matrices for each station. From our simulation tests, they agree quite well with Section III-B.

Fixed setup parameters: In order to simulate the actual physical setup, all the setup parameters are selected to be almost the same as those used in the real experiments to be described, except that due to the x axis problem with our robot (to be described later), the station generation process used in the real experiment is modified. In the following, the setup parameters are set as follows:

$$|T_c| = 6.65\,\text{in} \quad |T_{cg}| = 9.5\,\text{in}$$
$$N = 3 \quad \sigma_{T_{ci}} = 3\,\text{mil}$$
$$\sigma_{T_{gi}} = 5\,\text{mil} \quad \sigma_{R_{ci}} = \sigma_{R_{gi}} = 1.5\,\text{mrad}.$$
$$\angle(P_{g12}, P_{g23}) = \pi/3$$

In the following, we show the simulation results of four critical parameters on the error of rotation R_{cg} and translation T_{cg}. These four critical parameters are tested separately. For the testing of each parameter, all the other setup and critical parameters are set as above, while the very parameter under test will be allowed to vary over a given range. 1000 tests are done for each case, and statistics are gathered. The results are shown in the following figures.

3) Simulation Results:

Effect of the size of interstation rotation angle on accuracy: Section III-B described the relationships between the size of the interstation rotation angle and the R_{cg} as well as T_{cg} accuracy, and gave procedures for improving the accuracy. Extensive simulation has been done and the results are consolidated into Figs. 8 and 9 (one for rotation error and the other translation). The curve is linear up to statistical sampling tolerance. It agrees quite well with Observation 2 in Section III-B, and it confirms the recommendation made in Section III-C.

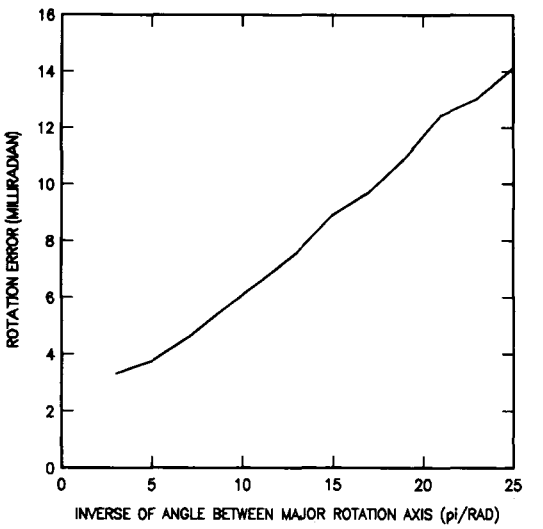


Fig. 10. RMS rotation error as a function of inverse of angle between two different interstation rotation axes.

Effect of angle between interstation rotation axes on accuracy: The situation is similar to that above except that $\angle(P_{g12}, P_{P_{g23}})$ is allowed to vary while $\theta_{R_{g12}}$ and $\theta_{R_{g23}}$ are fixed. Fig. 10 shows the average error of rotation as a function of $\angle(P_{g12}, P_{P_{g23}})$. It is again linear, as predicted in Lemma I.

Effect of camera-to-calibration-plate distance on accuracy: According to (31), the translation error has a dominant effect on $\sigma_{T_{cg}}$ unless $\sigma_{T_{ci}}$ or $\sigma_{T_{gi}}$ are enormous. Fig. 11 reflects this quite well. The RMS error of T_{cg} is plotted against this parameter. The curve is generally linear, but around the origin, it bends somewhat, due to the fact that when $|T_c|$ is small, its effect is no longer dominant and the effect of $\sigma_{T_{gi}}$ shows up.

Effect of number of interstation pairs on accuracy: Figs.

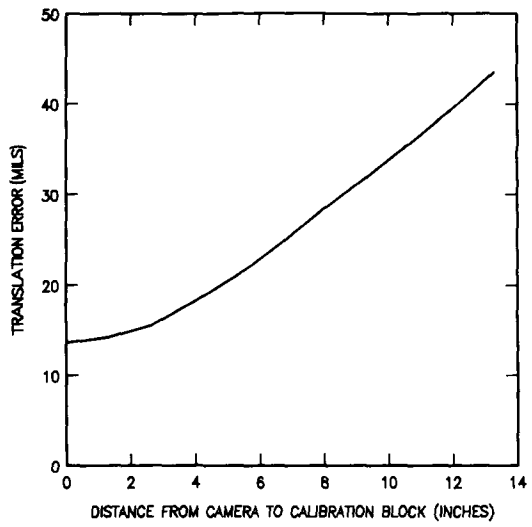


Fig. 11. RMS translation error as a function of the inverse of distance between camera and calibration plate.

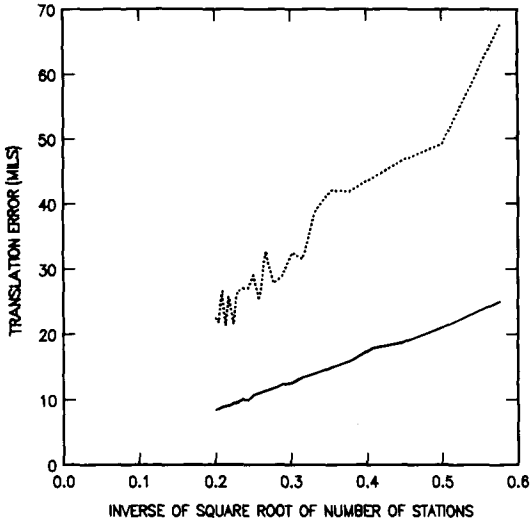


Fig. 12. Translation error as a function of inverse of square root of the number of interstation pairs. The solid line is RMSE and dashed line is maximum error.

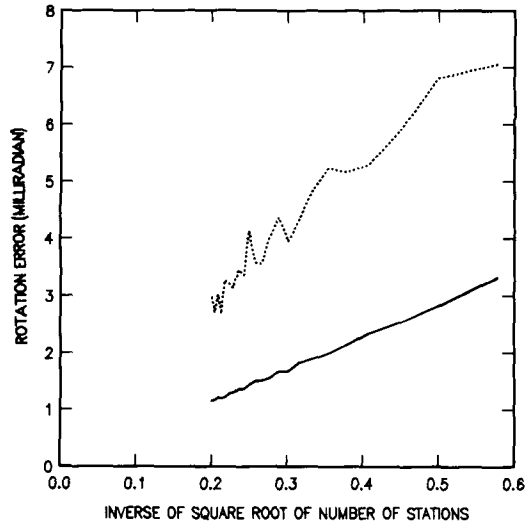


Fig. 13. Rotation error as a function of inverse of square root of the number of interstation paris. The solid line is RMSE and dashed line is maximum error.

12 and 13 show the error of translation and rotation as a function of the inverse of square root of the number of interstation pairs. The solid line shows the RMS error, while the dashed line is the maximum error out of one thousand tests. As expected, the RMS error increases linearly as the inverse of $\sqrt{N}$. Since the proposed technique is quite efficient, and the station pose planning and robot motion are automatic, increasing the number of stations is quite feasible, and pays off well.

B. Real Experiments

1) Setup Description: Fig. 3 shows the physical setup we used. A Javelin CCDE 480×388 camera is fastened to the last joint of an IBM Clean Room Robot (CRR). The CRR has two manipulators, each with seven degrees of freedom (including gripper opening). We only use one of the manipulators. The CRR is an electric box frame Cartesian robot. There are three linear joints (x, y, z) and three rotary joints (roll, pitch, and yaw) for each manipulator. The work volume is about 6 ft by 4 ft by 2 ft and the repeatability for linear joint is about 4 mil, and that for the rotary joints 1 mrad. The accuracy is calibrated to a limited extent. The scale and offset for each rotary joint are calibrated to 3-mrad accuracy. The rotation axes for the three rotary joints are supposed to be intersecting at the same point (origin of RW coordinate frame), but we did not calibrate that. The x axis has some problems: For our robot, the z beam sags, causing the movement in the x axis to be like that of a pendulum. This effect is not fully calibrated yet, but we suspect that it generates about 20-mil translation and 15-mrad rotation within a work range of 15 in. Due to this problem, we are forced to modify the station generation procedure used in the simulation in order to avoid using the x axis. Either with or without moving the x axis, the station placement and manipulator motion planning is automatic, and the number of stations can be arbitrary without manual intervention.

The calibration block is a clear glass plate with the center 1 in by 1 in area filled with 36 black discs printed on it using step-and-repeat photographic emulsion (see Fig. 14). The discs are 5000 μm apart with 2000-μm radius (accurate to 1 μm). The calibration is back-lighted and sits in the middle of the work space.

2) Accuracy Assessment: The accuracy of our hand/eye calibration results is assessed by how accurately we can predict the placement of a camera in 3D world with any arbitrary manipulator movement. As was indicated in Section I-D, one of the main reasons why robot hand/eye calibration is important is that the robot needs to know not only where the gripper is, but also where the camera is in the work space, so that the measurement taken by vision can be related to the robot. Being able to determine where the camera is in the work space for an arbitrary manipulator movement is thus the primary goal. This is tested in the following steps:

Step 1: Move the manipulator to $2N$ different positions where N is greater than 2. For each station i, compute the camera to calibration block homogeneous transformation H_{ci} using extrinsic calibration. This takes about 90 ms per station.

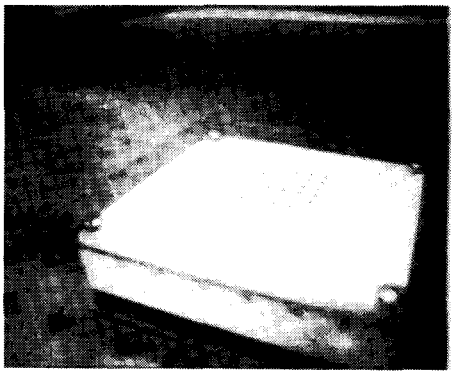


Fig. 14. Calibration block is a clear glass plate with 36 discs printed on it using photographic emulsion. The accuracy is 4 μm.

The robot gripper position and orientation relative to robot world, which is H_{gi}, is also recorded.

Step 2: Compute H_{cg} using procedures in Section II-B2b, using data from stations 1 through N.

Step 3: For each station k (k from 1 to N) compute the homogeneous matrix H_{RC} (homogeneous transformation from robot world frame RW to calibration block world frame CW) by

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

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

Step 4: Let stations $N + 1$ through $2N$ be called verification stations. For each of the verification stations, predict the position and orientation of the camera relative to the robot world base coordinate RW by $H_{cg}^{-1} H_{gk}^{-1}$ where k is the station index, and H_{gk} is computed from robot joint coordinates (see Section II-B). Compare this predicted position and orientation with $H_{ck} H_{RC}$, 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	
N	Rotation Error (mrad)	Translation Error (mil)
4	4.568	23.238
6	3.304	19.078
8	3.264	26.712
10	2.888	14.642
12	2.782	12.516

Since there is no absolute H_{cg} ground truth to compare with, the accuracy has to be assessed as the error of new camera pose prediction, as described earlier. The effect of N is indeed very significant. We have a program that automatically plans the movement of the manipulator for an arbitrary number of stations, and since the algorithm proposed in this paper is quite efficient, increasing the number of frames is quite easy. Also observe that the error of the predicted camera pose includes both the error of the calibrated hand/eye relationship and the robot's positioning error. Notice from the table that for 10 stations, the translation error is about 14 mil. But the robot's positioning accuracy is worse than 10 mil. This means that the eye-to-hand relationship is calibrated to better than 10 mil. Using the error formula in (32) scaled by $\sqrt{10/3}$, the error fo T_{cg} is predicted to be 10.66 mil, agreeing well with the real experiment data. The rotation error is about 2.88 mrad. Notice that the error of rotary joint is about 2.5 mrad. Therefore, the actual error of R_{cg} should also be of this order of magnitude. This agrees very well with the prediction by (28), which gives 2.557 mrad. Notice that the error in the table is not strictly monotonic with respect to the inverse of the square root of number of stations. This is due to the fact that the simulation curves presented earlier were averaged over 1000 tests, while here, for each N, there is only one test. Also, since the robot error itself gets into it, it is more unpredictable, while the simulation curves only show the H_{cg} error. Nevertheless, the error generally decreases nicely as the number of stations increases.

V. Conclusion

This paper introduced a high-speed, high-accuracy, versatile, simple, and fully autonomous technique for 3D robotics hand/eye calibration. It is high speed since it takes only about $100 + 64N$ arithmetic operations to compute the hand/eye relationship after the robot finishes the movement, and incurs only additional 64 arithmetic operations for each additional station. This makes the current algorithm the fastest compared with the state of the art. The speed performance is especially attractive to those applications where the hand/eye configuration needs to be changed frequently. For example, the robot may pick up the camera to perform some task, and then put it right back to a holder. Since the grasping cannot be precise, hand/eye calibration must be performed frequently. It is also important to those tasks where hand/eye relationships need be changed frequently due to different task requirements. As for the accuracy, no other reported hand/eye calibration technique does any better. The results in our real experiments could be further improved if we changed the optics and the size of calibration block, as well as the mounting position, so that all of the critical factors described in the accuracy analysis section would be taken into consideration.

References

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

[2] K. S. Fu, R. C. Gonzalez, and C. S. G. Lee, *Robotics: Control, Sensing, Vision and Intelligence.* New York, NY: McGraw-Hill, 1987.

[3] A. Izaguirre, J. Summers, P. Pu, "A new development in camera calibration, calibrating a pair of mobile TV cameras," to appear in *Int. Robotics Res.*, 1987.

[4] J. L. Jenkins and J. D. Turner, *Optimal Spacecraft Rotational Manervers.* Amsterdam, The Netherlands: Elsevier, 1986.

[5] R. Lenz and R. Y. Tsai, "Calibrating a Cartesian robot with eye-on-hand configuration independent of eye-to-hand relationship," in *Proc. IEEE Computer Vision and Pattern Recognition Cont.* (Ann Arbor, MI, June 5–9, 1988).

[6] ——, "Techniques for calibration of the scale factor and image center for high accuracy 3D machine vision metrology," in *Proc. IEEE Int. Conf. on Robotic and Automation* (Raleigh, NC). Also to appear in *IEEE Trans. Pattern Anal. Mach. Intel.*

[7] G. Puskorius and I. Feldkamp, "Calibration of robot vision," in *Proc. IEEE Int. Conf. on Robotic and Automation* (Raleigh, NC, 1987).

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

[9] Y. Shiu and S. Ahmad, in *Proc. IEEE Int. Conf. on Robotic and Automation* (Raleigh, NC, 1987).

[10] 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 presented at the 1986 IEEE International Conference on Computer Vision and Pattern Recognition, Miami, FL, June 22-26.

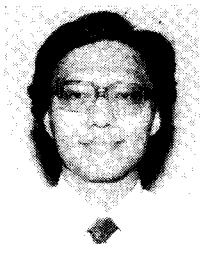
[11] R. Tsai and R. Lenz, "Review of the two-stage camera calibration technique plus some new implementation tips and new techniques for center and scale calibration," presented at the Second Topical Meeting on Machine Vision, Optical Society of America, Lake Tahoe, Mar. 18-20, 1987.

[12] ——, "A new technique for fully autonomous and efficient 3D robotics hand-eye calibration," in *4th Int. Symp. on Robotics Research* (Santa Cruz, CA, Aug. 9-14, 1987).

[13] ——, "Review of RAC-based camera calibration vision," Society of Manufacturing Engineers, Nov. 1988.

[14] ——, "Real time versatile robotics hand/eye calibration using 3D machine vision," presented at the Int. Conf. on Robotics and Automation, Philadelphia, PA. Apr. 24-29, 1988.

[15] ——, "Three-dimensional mechanical part measurement using a vision/robot system," IBM Res. Rep. RC 10506, May 8, 1984.



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

He was employed by Bell-Northern Research/INTS-Telecommunications, Montreal, 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 3D time-varying scene analyses. 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 T. J. Watson Research Center, Yorktown Heights, NY, where he has been initiating and leading projects in the area of 3D machine vision. He 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 is the Technical Editor of Robotics Vision and Inspection System for IEEE TRANSACTIONS ON ROBOTICS AND AUTOMATION. He has served as chairman of the Computer Vision committee for the IEEE Society of Robotics and Automation for 1988. He served on the program committee for 1989 IEEE Int. Conf. CVPR, and has served as session chairman for 1986 IEEE Intern. Conf. on CVPR, 1987 and 1988 IEEE International Conference on Robotics and Automation. He is on the editorial board for *Robotics Review* (MIT Press). His major research interests include 3D robotics and geometric vision, sensor planning, modeling and calibration, 3D stereo measurement, inspection, automated assembly, etc.

Reimar K. Lenz was born in Aachen, W. Germany, on January 10, 1956. He received the B.S. degree from the TU Stuttgart in 1977, the M.S. and Ph.D. degree from TU 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, 64 kBit/s TV) and at the IBM T. J. Watson Research Center, Yorktown Heights, NY, during 1986 and 1987 (Manufacturing Research, Robot Vision, Real-Time Vision). His current research interests are camera calibration and pattern recognition.