[17] G. S. Young, T. H. Hong, M. Herman, and J. C. S. Yang, "Kinematic calibration of an active camera system," in *Proc. IEEE Conf. Comput. Vision Pattern Recognition*, 1992, pp. 748–751.
[18] H. Zhuang and Z. S. Roth, "A linear solution to the kinematic parameter identification of robot manipulators," *IEEE Trans. Robot. Automat.*, vol. 9, pp. 174–185, Apr. 1993.

Active Self-Calibration of Robotic Eyes and Hand–Eye Relationships with Model Identification

Guo-Qing Wei, Klaus Arbter, and Gerd Hirzinger

***Abstract*— In this short paper, we first review research results of camera self-calibration achieved in photogrammetry, robotics and computer vision. Then we propose a method for self-calibration of robotic hand cameras by means of active motion. Through tracking a set of world points of unknown coordinates during robot motion, the internal parameters of the cameras (including distortions), the mounting parameters as well as the coordinates of the world points are estimated. The approach is fully autonomous, in that no initial guesses of the unknown parameters are to be provided from the outside by humans for the solution of a set of nonlinear equations. Sufficient conditions for a unique solution are derived in terms of controlled motion sequences. Methods to improve accuracy and robustness are proposed by means of best model identification and motion planning. Experimental results in both a simulated and a real environments are reported.**

***Index Terms*— Active motion, hand-cameras, hand–eye calibration, model identification, motion planning, self-calibration, unique solution.**

I. Introduction

In order to use cameras for estimating robot motion for object manipulation, it is usually necessary to do the following three calibrations: camera calibration, hand–eye calibration, and robot calibration. In this paper, we address the first two problems, assuming the third, i.e., robot calibration, has been done.

For camera calibration, the basic theory has been developed in the field of photogrammetry [17]. Calibration approaches used nowadays can be generally categorized into two classes: test-field calibration and self-calibration. Test-field calibration determines the camera internal and external parameters from images of a set of control points whose three-dimensional (3-D) coordinates are known in a world coordinate system [5], [17], [22], [26], [29]. Since it is difficult to fabricate and also maintain for a long period a highly accurate control field, the self-calibration methods were developed, which estimate not only the camera parameters but also the coordinates of the control points, based on multiple images of the same control field acquired at different camera stations. The collinearity constraint [11] and the coplanarity constraint [12], [17, pp. 259–260] were the most popular equations used in self-calibration, the later of which eliminates the world point coordinates. A major difficulty with the photogrammetric self-calibration approach is that good initial guesses of the unknown parameters have to be provided from the outside by humans for an iterative procedure to converge to the correct solution. In Computer Vision, Faugeras *et al.* [6] recently proposed a camera self-calibration approach which involves only camera internal parameters. "The method, however, was found to be noise sensitive and also computationally intensive, in spite of some improvements [14] in the formulation" (Luong and Faugeras [15]) The above self-calibration methods are passive in the sense that no knowledge about the camera motion is assumed. Active methods employ knowledge about camera motion, in terms of either a movable mechanical devices or a robot [1], [3], [16]. Among the self-calibration methods, either passive or active, developed in computer vision, almost all of them do not consider lens distortions (and some of the formulations are only applicable to the no-distortion case, e.g., [6], [16]). Besides, most of the above methods do not address the problem of how to automate the process of getting initial values of the unknown parameters in nonlinear iteration.

For hand–eye calibration, two basic hand–eye configurations were used in the robotics literature: static cameras and on-hand cameras. In the case of stationary cameras, hand–eye calibration was performed by moving the hand and tracking in the image, a single point (e.g., a light emitting diode, LED) on the gripper [2], [10], [13]. When the coordinates of the LED with respect to the hand coordinate system are known, hand–eye calibration is equivalent to camera calibration [10], [13], where the control points are generated by hand movements. These approaches, however, cannot deal with multiple points with unknown relative positions. The capability of utilizing multiple points is important in improving robustness and reducing the number of robot motions required. For the camera-on-hand configuration, earlier work on hand–eye calibration assume that the cameras have been calibrated in advance [21], [24]. By moving the robot hand to at least three stations, the hand–eye calibration problem was shown to be equivalent to solving equations of the form $A_iX = XB_i$ [21], [24], [25], [30]. In this approach, the robot motion matrix A_i is calculated from the known robotic kinematics; while the camera motion matrix B_i is determined by camera extrinsic calibration in terms of a known control field [24]. Recently, Zhuang *et al.* [31] calibrated a hand-camera, the hand–eye transformation, and the robot together by using a known control field, assuming that the image center and scale factor are known *a priori*. It was suggested in [31] that gauging devices be used to manually measure some parameters as the initial values in the solution of a set of nonlinear equations.

In this paper, we propose a complete autonomous approach for self-calibration of hand-cameras and hand–eye relationships.

The proposed approach has the following features

1) No metric control points are used. The coordinates of the calibration points are determined by the calibration itself. Besides, the number of object points can be as few as one *and* as many as one wants.
2) The initial values of all the unknowns for starting the iteration in the solution of a system of nonlinear equations for the calibration are found automatically, all in closed forms, by the calibration method itself.
3) Lens distortions are considered.
4) The method does not rely on any system knowledge or any pre-calibration or partial calibration of the camera's internal or external parameters.

As another salient feature, the method identifies the best lens distortion model and plans the robot motion such that both robustness and accuracy can be improved.

Manuscript received November 22, 1996. This paper was recommended for publication by Editor A. Goldenberg upon evaluation of the reviewers' comments.

The authors are with the Institute of Robotics and System Dynamics, German Aerospace Research Establishment, Oberpfaffenhofen 82234, Germany (e-mail: Guo-Qing.Wei@dlr.de, Klaus.Arbter@dlr.de, and Gerd.Hirzinger@dlr.de).

Publisher Item Identifier S 1042-296X(98)01432-3.

The paper is organized as follows. In Section II, we present the method of active camera calibration. In Section III, we address the problem of how to identify the best distortion model and to design robot motion. In Section IV, the method is tested and compared with a modified Tsai's algorithm. Conclusions are given in Section V.

II. Self-Calibration of a Hand–Eye System

Suppose a camera is rigidly mounted on a robot gripper. We denote the camera coordinate system by $\langle c\rangle$: $X_c - Y_c - Z_c$, the gripper coordinate system by $\langle g\rangle$: $X_g - Y_g - Z_g$. The transformation from the camera coordinate system to the hand (gripper) coordinate system is represented by the rotation R_{cg} and translation $\mathbf{t}_{cg}$ as

$$\begin{pmatrix} X_g \\ Y_g \\ Z_g \end{pmatrix} = R_{cg} \begin{pmatrix} X_c \\ Y_c \\ Z_c \end{pmatrix} + \mathbf{t}_{cg} \tag{1}$$

or in homogeneous form as

$$\begin{pmatrix} X_g \\ Y_g \\ Z_g \\ 1 \end{pmatrix} = H_{cg} \begin{pmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{pmatrix} \tag{2}$$

where

$$H_{cg} = \begin{bmatrix} & R_{cg} & & | & \mathbf{t}_{cg} \\ - & --- & - & | & -- \\ 0 & 0 & 0 & | & 1 \end{bmatrix} \triangleq H(R_{cg}, \mathbf{t}_{cg}). \tag{3}$$

Assume there is a set of world points $\{P_i\}$, whose coordinates in an initial camera coordinate system $\langle c_0\rangle$ are represented by (X_{i0}, Y_{i0}, Z_{i0}), $i = 1, 2, \cdots, N$, where N is the number of points. If we move the robot hand to M different stations $\langle g_j\rangle$, $j = 0, 1, 2, \cdots, M$, where $\langle g_0\rangle$ stands for initial hand station, then the coordinates (X_{ij}, Y_{ij}, Z_{ij}) of the ith point at the jth camera station $\langle c_j\rangle$ are

$$\begin{pmatrix} X_{ij} \\ Y_{ij} \\ Z_{ij} \\ 1 \end{pmatrix} = H_{c0j} \begin{pmatrix} X_{i0} \\ Y_{i0} \\ Z_{i0} \\ 1 \end{pmatrix} \tag{4}$$

where H_{c0j} is the homogeneous transformation matrix from $\langle c_0\rangle$ to $\langle c_j\rangle$. If we use $H_{g0j} = H(R_{g0j}, \mathbf{t}_{g0j})$ to denote the robot motion matrix from $\langle g_0\rangle$ to $\langle g_j\rangle$, with R_{g0j} and $\mathbf{t}_{g0j}$ being the rotation and translation components, respectively, then H_{c0j} can be computed as

$$H_{c0j} = H_{cg}^{-1} H_{g0j} H_{cg}. \tag{5}$$

It can be easily shown that the rotation and translation components of H_{c0j} are

$$R_{c0j} = R_{cg}^T R_{g0j} R_{cg} \tag{6}$$

$$\mathbf{t}_{c0j} = R_{cg}^T (R_{cg} - I)\mathbf{t}_{cg} + R_{cg}^T \mathbf{t}_{g0j}. \tag{7}$$

Fig. 1 illustrates the chain of transformations.

Suppose (u_{ij}, v_{ij}) are the measured image coordinates of the ith world point P_i at the jth camera station $\langle c_j\rangle$. Then the following perspective equations can be obtained as the measurement equations:

$$\overline{u}_{ij} = f_x \frac{X_{ij}}{Z_{ij}}, \overline{v}_{ij} = f_y \frac{Y_{ij}}{Z_{ij}}; \qquad i = 1, \cdots, N; \quad j = 0, 1, \cdots, M \tag{8}$$

where f_x and f_y are the effective focal lengths in the x and y directions of the image plane, respectively; and $(\overline{u}_{ij}, \overline{v}_{ij})$ are the distortion-compensated frame buffer coordinates from the measured

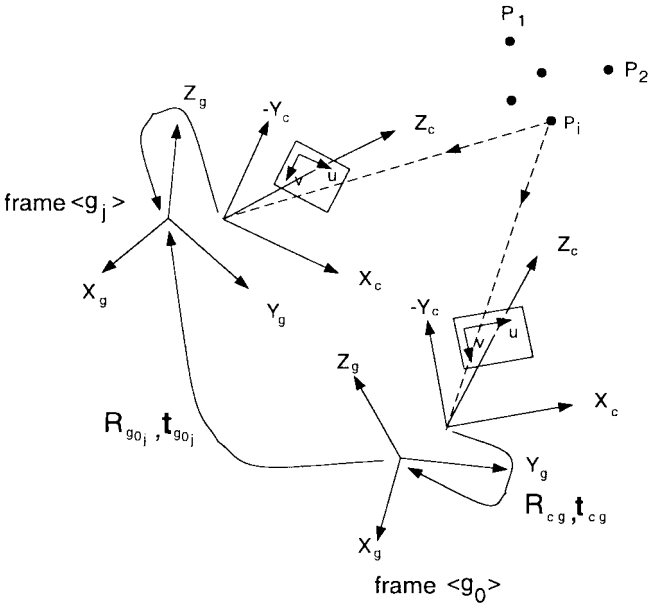


Fig. 1. The hand and eye coordinate systems and hand motion.

ones (u_{ij}, v_{ij}), according to a distortion model containing both radial and tangential components[1] [4]:

$$\overline{u}_i = u_i - u_0 + p_1(3x^2 + y^2/k^2) + 2p_2 y^2 + x(1 + k_1 r^2) \tag{9}$$

$$\overline{v}_i = v_i - v_0 + 2p_1 xy + p_2(x^2 k^2 + 3y^2) + y(1 + k_1 r^2) \tag{10}$$

with

$$x = u_i - u_0;\ y = v_i - v_0 \tag{11}$$

and

$$r^2 = k^2 x^2 + y^2 \tag{12}$$

where we have, for simplicity, omitted the subscript j; the parameters (u_0, v_0) are the coordinates of the image center; p_1, p_2, and k_1 are the tangential and radial distortion coefficients, respectively; $k = f_y/f_x$ is the image plane scale factor, which is the ratio of the between-pixel distances d_x and d_y in the x- and y- directions of the camera CCD plane.

By inserting (3)–(5) into (8), we can now state the camera self-calibration problem as: determine from the measurement equations (8), the world coordinates (X_{i0}, Y_{i0}, X_{i0}), $i = 1, \cdots, N$, the camera internal parameters $u_0, v_0, f_x, f_y, p_1, p_2, k_1$, and the hand–eye configuration parameters (external parameters) R_{cg} and $\mathbf{t}_{cg}$, based on the robot motion parameters $\{H_{g0j}\}$ and the image coordinate measurements $\{(u_{ij}, v_{ij})\}$. Here, we use the roll–pitch–yaw angles (α, β, γ) to parameterize the hand–eye rotation matrix R_{cg}.

In the above formulation, assumptions are made of the known robot motions $\{H_{g0j}\}$. This is an assumption adopted in most previous work on hand–eye calibration, e.g., Tsai and Lenz [24], Shiu and Ahmad [21], to name a few. This assumption is not so hard to meet for most industrial robots, since we do not require a high *absolute* positioning precision. Rather, it is the relative motion that is used in the calibration. Often, it is helpful to use the *measured* amounts of relative motions (from joint angles) rather than the *commanded* ones in the calibration equations, because of error accumulation in the execution of a specified motion sequence.

[1]Notice that the distortion model is expressed directly in the frame buffer instead of in the CCD plane, so that a scale factor k is involved, as compared with that in [4].

Because of the nonlinearity of the measurement equations, we adopt the active motion principle to find the initial values of the unknowns to be used in nonlinear iteration, all in closed forms as follows.

Suppose the stations $\langle g_j \rangle$, $j = 1, 2, \cdots, M_t$ are obtained by M_t pure translational motions of the hand started from $\langle g_0 \rangle$. As a first order approximation, we assume, for the time being, that the distortion coefficients are all zero, i.e., $p_1 = 0$, $p_2 = 0$, and $k_1 = 0$. Under these assumptions, the perspective equations (8) can be reduced as in (13) and (14), shown at the bottom of the page, where $R_{cg}(m, n)$ and $t_{g0j,n}$ are the (m, n)th and nth components of R_{cg} and $\mathbf{t}_{g0j}$, respectively. If, in the above equations, we consider only one point, say P_1, then the obtained sub-system (i.e., for $i = 1$, $j = 1, 2, \cdots, M_t$) can be viewed as the projective equations of a set of *virtual* world points $\mathbf{v}_j = (t_{g0j,1}, t_{g0j,2}, t_{g0j,3})^T$, $j = 0, 1, \cdots, M_t$, with the virtual world coordinate system located at $\mathbf{t}_v = (X_{10}, Y_{10}, Z_{10})^T$ and having the orientation $R_v = R_{cg}^T$ with respect to the camera coordinate system. This insights allows us to employ the method of perspective transformation matrix [5] to find the internal parameters u_0, v_0, f_x, f_y, the external parameters R_{cg}, and the world coordinates (X_{10}, Y_{10}, Z_{10}) in closed forms. [The perspective transformation matrix here refers to the 3×4 matrix which transforms the 3-D virtual points $\{\mathbf{v}_j\}$ to the two-dimensional (2-D) frame buffer coordinates $\{(u_{1j}, v_{1j})\}$.] The obtained values for u_0, v_0, f_x, f_y and R_{cg} are then substituted into (13) and (14), for $i = 2, 3, \cdots, N$. The resulting equations can be rearranged in a linear form on (X_{i0}, Y_{i0}, Z_{i0}) for $i = 2, 3, \cdots, N$, by multiplying both sides with the denominator. From the obtained linear system, the world coordinates can be easily solved for in closed forms. So far, we have obtained u_0, v_0, f_x, f_y, R_{cg}, and the world coordinates $(X_{i0}, Y_{i0}, Z_{i0})^T$. Suppose then the motion stations $\langle g_j \rangle$, $j = M_t + 1, \cdots, M$ contain nonzero rotational components, which we call *compound motions*. By substituting all the obtained parameters into (4)–(8) for $j = M_t + 1, \cdots, M$, the measurement (8) can be similarly rearranged in a linear form on $\mathbf{t}_{cg}$ and can be easily solved for the hand–eye translation parameters. The complete procedure of estimating the unknown parameters above also leads to the following sufficient conditions for a unique solution.

Lemma 1: If a robot undergoes: a) 5 translational motions ($M_t = 5$), among which no more than 3 of the translation vectors with respect to the initial hand system are coplanar and b) 2 compound motions whose axes of rotation (with respect to the initial hand system) do not coincide with $\mathbf{t}_{cg}$, then the solutions for all the unknown parameters are unique.

Proof: See [27].

If the image data are free of noise and there exist no lens distortions, the values of the unknown parameters estimated above are exact, since we have introduced no approximations in the estimation. In general cases, however, when noises and distortions are both present, the obtained values can only be regarded as good initial guesses because the estimation errors accumulate in a sequential way. To refine the estimates, the Newton–Raphson method is used to adjust all the parameters (including the distortion coefficients) simultaneously, by local linearization of the original measurement (8) about the initial values. It can be easily demonstrated that the normal equations of the locally-linearized measurement equations take the following form:

$$\begin{pmatrix} A_1 & \mathbf{0} & \cdots & \mathbf{0} & B_1 \\ \mathbf{0} & A_2 & \cdots & \mathbf{0} & B_2 \\ \cdot & \cdot & & \cdot & \cdot \\ \cdot & \cdot & & \cdot & \cdot \\ \cdot & \cdot & & \cdot & \cdot \\ \mathbf{0} & \mathbf{0} & \cdots & A_N & B_N \\ B_1^T & B_2^T & \cdots & B_N^T & B_0 \end{pmatrix} \begin{pmatrix} \delta\mathbf{x}_1 \\ \delta\mathbf{x}_2 \\ \cdot \\ \cdot \\ \cdot \\ \delta\mathbf{x}_N \\ \delta\mathbf{x}_0 \end{pmatrix} = \begin{pmatrix} \mathbf{e}_1 \\ \mathbf{e}_2 \\ \cdot \\ \cdot \\ \cdot \\ \mathbf{e}_N \\ \mathbf{e}_0 \end{pmatrix} \tag{15}$$

where $\delta\mathbf{x}_i$ is a 3×1 increment vector for the world coordinates, $i = 1, \cdots, N$, $\delta\mathbf{x}_0$ a $m \times 1$ increment vector for the internal and external parameters, with m being the number of internal and external parameters; $\mathbf{e}_i$'s are the residual errors; A_i, B_i, and B_0 are matrices of size 3×3, $3 \times m$ and $m \times m$, respectively; The bordered block-diagonal structure of the normal matrix in (15) allows us to use a partitioning scheme proposed by Brown [17] to solve the normal equations. This avoids the inversion of a very large matrix when the number of world points is large. The computational burden in solving the normal equation by partitioning is made only linearly proportional to the number of world points. It should be noted that the Levenberg–Marquardt algorithm [19], which has been widely used in nonlinear optimizations, takes no account of the structure of the normal matrix. Its computational cost increases quadratically with the dimensions of the problem, which can be very large in problems like uncalibrated 3-D reconstruction [18] or self-calibration. We give, in the Appendix, details of the matrix reduction scheme and derive the Gauss–Markov theorem [20] for the estimation of parameter variances in the reduction case.

III. Model Identification and Motion Planning

In this section, we shall deal with two issues related to improvement of robustness and accuracy of the proposed method.

A. Model Identification

Model identification is concerned with the choice of the best model in describing a problem. Since the calibration equations derived in the last section are based on physical processes, the number of parameters employed should be minimum, except in the distortion model, which may be camera dependent. We suppose here that the distortion models of (9) and (10) cover the possible distortions, so that there is only a possible over-parameterization in the model. Overparameterization may cause the variances of some of the estimated parameters to increase, especially when there are few measurements. Although the computed parameters *as a whole* in the case of over-parameterization may still be useful in performing the correct transformation from the sensor space to the world space, the individual parameters when used alone for other purposes are not as reliable, because of the correlations in the estimates. To cope with this problem, we use the statistic inference method to deduce whether some specific (or all) distortion components should be excluded from the final calibration procedure, thus increasing the reliability of the estimated parameters.

The student-distribution (t-distribution) [8], [20] could be used to test whether a certain variable takes on a presumed value under a selected significance level $(1 - \alpha)$, based on the estimated variance of the variable [7]. A problem with the t-test is that it provides

$$u_{ij} - u_0 = f_x \frac{R_{cg}(1,1)t_{g0j,1} + R_{cg}(2,1)t_{g0j,2} + R_{cg}(3,1)t_{g0j,3} + X_{i0}}{R_{cg}(1,3)t_{g0j,1} + R_{cg}(2,3)t_{g0j,2} + R_{cg}(3,3)t_{g0j,3} + Z_{i0}} \tag{13}$$

$$u_{ij} - u_0 = f_x \frac{R_{cg}(1,2)t_{g0j,1} + R_{cg}(2,2)t_{g0j,2}R_{cg}(3,2)t_{g0j,3} + Y_{i0}}{R_{cg}(1,3)t_{g0j,1} + R_{cg}(2,3)t_{g0j,2} + R_{cg}(3,3)t_{g0j,3} + Z_{i0}}, \quad i = 1, \cdots, N;\ j = 0, 1, \cdots, M_t \tag{14}$$

reliable results only when the variables to be tested are not correlated. We propose here to use the F-distribution (Fisher's Distribution)[2] to overcome this drawback. To sketch its use in camera model identification, two *null hypotheses* are made: $H_0^{\langle 1\rangle}$: $p_1 = 0$, $p_2 = 0$, $k_1 = 0$, and $H_0^{\langle 2\rangle}$: $p_1 = 0$; $p_2 = 0$, which correspond to two practical situations of no-distortion and pure radial distortion. The *alternate hypothesis* H_1 is that none of the distortion parameters are vanishing, i.e., both tangential and radial distortions are present. To test the null-hypotheses, the calibration is run three times under the three hypotheses $H_0^{\langle 1\rangle}$, $H_0^{\langle 2\rangle}$, and H_1, respectively. Denote the respective sums of squares of residuals of the measurement equations by $E(H_0^{\langle 1\rangle})$, $E(H_0^{\langle 2\rangle})$, and $E(H_1)$. It can then be shown [20] that

$$F^{\langle 1\rangle} = \frac{[E(H_0^{\langle 1\rangle}) - E(H_1)]}{3} \div \frac{E(H_1)}{n-m} \tag{16}$$

and

$$F^{\langle 2\rangle} = \frac{[E(H_0^{\langle 2\rangle}) - E(H_1)]}{2} \div \frac{E(H_1)}{n-m} \tag{17}$$

are approximately F-distributed with (3, $n-m$) and (2, $n-m$) degrees of freedom if the respective hypotheses $H_0^{\langle 1\rangle}$ or $H_0^{\langle 2\rangle}$ are true; where n is the number of equations used in the calibration; m is the total number of unknown variables.

By selecting a significance level $(1-\alpha)$, e.g., $\alpha = 0.01$, we test if

$$F^{\langle 1\rangle} \leq F_{1-\alpha/2}(3,\, m-n) \tag{18}$$

$$F^{\langle 2\rangle} \leq F_{1-\alpha/2}(2,\, m-n) \tag{19}$$

where $F_{1-\alpha/2}(m_1, m_2) : P\{F < F_{1-\alpha/2}(m_1, m_2)\} = 1-\alpha$ is the value of a $F(m_1, m_2)$-distributed variable, below which the probability is $1-\alpha$. If (18) is satisfied, then $H_0^{\langle 1\rangle}$ is true; if (19) is satisfied, $H_0^{\langle 2\rangle}$ is true; if none of them are satisfied, then H_1 is true. After the correct model is identified, the calibration is run with the decided model. Experiments show that with the proposed model identification, the variances of some parameters can be reduced by as much as a factor of 3.

B. Motion Planning

In Section II, we did not address the problem of how to move the robot optimally, such that the variances of the estimated parameters be minimized (with respect to motions). Formal experimentation design methods [8] prove to be impractical for problems of high dimensions like ours. Instead, we propose a heuristic, other than theoretically sound method, which is easy to implement and suboptimal.

Our basic idea of motion planning is to do first an initial calibration without planning. The motions of the robot in the initial calibration can be simply chosen as translations along and rotations around the coordinate axes of the gripper. Through the initial calibration we obtain knowledge about the camera parameters and the world coordinates. This knowledge, though not very exact, provides the crucial information for motion planning.

1) Effect of Robot Motion Uncertainties: The effect of robot motion uncertainties on calibration depends on both hand–eye configuration and the object depths in the camera. It is very difficult to analytically write down such dependencies for general cases and to use experimentation design methods [8], if not computationally prohibitive, to reduce the influences of robot motion errors. To circumvent this difficulty, we analyze the *equivalent* image coordinate disturbances, caused by motion errors, of the feature points. For translational motion errors, the equivalent 3-D coordinate disturbances, being equal to the motion errors in magnitude, are projected onto the image plane through the perspective scaling of the object depth. Thus, the equivalent image coordinate disturbances become smaller when the camera is placed farther way from the object. For rotational motion errors, the equivalent 3-D coordinate errors are proportional to the object depth in the hand system. However, the 3-D errors, when projected onto the image plane, will be scaled by the object depth in the camera. Thus, the equivalent image coordinate disturbances cannot become infinitely large. It was shown in [27] that for a simplified hand–eye configuration (parallel coordinate axes), the minimum image disturbance is zero, and the maximum disturbance is bounded, for both translational and rotational errors. Since the image feature extraction errors depend on object depth too, which are difficult to be quantified, we are, in practice, still unable to find the optimal depth in the presence of all the error sources. The above analysis, however, provides an intuitive judgment of the effect of robot motion errors.

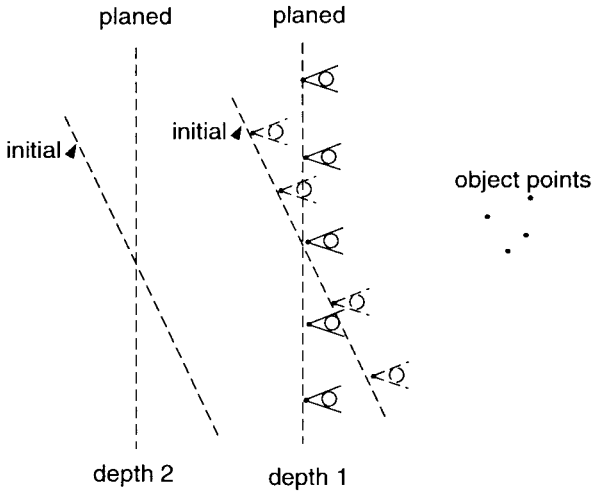


Fig. 2. Planning the translation of a camera.

2) Maximize the Motion Information: In the following we shall plan the translational and rotational motions of the robot at a given depth. Due to space limitations, We shall only outline the ideas behind the planning without giving detailed formulas for the computation of the planed motions.

a) Translational motion planning: From Section II, we know that translational motions contribute mainly to the determination of camera internal parameters and the hand–eye rotation parameters. Based on the initially calibrated parameters, we can plan the camera translations such that all stations maintain the specified depths, see Fig. 2. This keeps the image disturbances at a given depth to be roughly at the same level, as described previously. Most importantly, at each chosen depth level, we can design the translational motions such that the object points distribute *homogeneously* across the image plane. This would have not been possible if we knew nothing about the values of the camera parameters and the object coordinates (thanks to the initial calibration). Distributing the image points across the whole image plane, especially having enough points near the image boundary, contributes significantly to the reduction of the variances of camera internal parameters, especially the distortion parameters.

b) Rotation planing: In order to reduce the uncertainties in the hand–eye translation parameters $\mathbf{t}_{cg}$ and the coordinates of the world points, rotational components in the compound motions should be made as large as possible. Again, based on the initial calibration, we can maximize the rotation angle (which may be limited by working

[2]The F-distribution of (m_1, m_2) degrees of freedom (DOF), denoted by $F(m_1, m_2)$, is the ratio of two χ^2 (chi-square)-distributed variables of m_1 and m_2 DOF's divided by their respective DOFs; while a χ^2-distributed variable of k DOF's is the sum of k squared Gaussians of zero mean and unit variance.

TABLE I
RELIABILITY OF THE GAUSS–MARKOV THEOREM

Gauss-Markov	variance	σ_X	σ_Y	σ_Z	σ_α	σ_β	σ_γ	$\sigma_{t_{cg,1}}$
	Model M_0	0.11	0.11	0.10	0.00021	0.00050	0.00064	0.16
	Model M_1	0.11	0.14	0.12	0.00025	0.00058	0.00084	0.16
	Model M_2	0.36	0.31	0.13	0.00065	0.00173	0.00173	0.16
	variance	$\sigma_{t_{cg,2}}$	$\sigma_{t_{cg,3}}$	σ_{u_0}	σ_{v_0}	σ_{f_x}	σ_{f_y}	
	Model M_0	0.16	0.14	0.28	0.37	0.50	0.42	
	Model M_1	0.16	0.14	0.35	0.47	0.55	0.43	
	Model M_2	0.16	0.13	0.90	1.03	0.99	1.03	
Monte-Carlo	variance	σ_X	σ_Y	σ_Z	σ_α	σ_β	σ_γ	$\sigma_{t_{cg,1}}$
	Model M_0	0.15	0.16	0.096	0.00022	0.00053	0.00064	0.16
	Model M_1	0.15	0.16	0.096	0.00024	0.00058	0.00075	0.16
	Model M_2	0.15	0.16	0.096	0.00062	0.00164	0.00162	0.16
	variance	$\sigma_{t_{cg,2}}$	$\sigma_{t_{cg,3}}$	σ_{u_0}	σ_{v_0}	σ_{f_x}	σ_{f_y}	
	Model M_0	0.16	0.13	0.30	0.32	0.29	0.32	
	Model M_1	0.16	0.13	0.33	0.38	0.30	0.31	
	Model M_2	0.16	0.13	0.87	0.81	0.30	0.32	

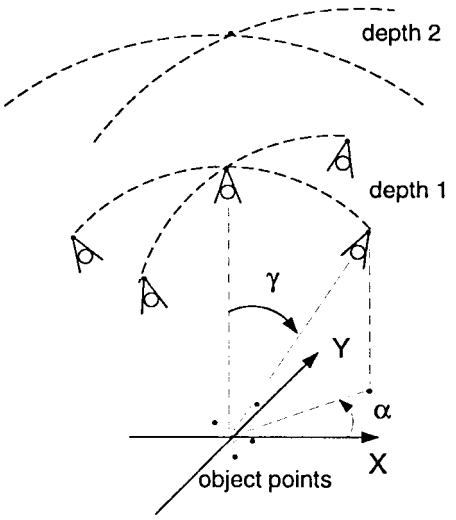


Fig. 3. Planning the rotation of a camera.

space or robot pose) while keeping the object depth unchanged and without having the object points stray out of view. This is realized by setting the camera stations on the surface of a sphere of radius equal to the chosen depth and having its origin located at the center of gravity of the object points, as illustrated in Fig. 3, where two angles α and γ control the amount of rotations.

IV. EXPERIMENTS

We implemented our self-calibration method in both a simulated and a real robot environment. In simulations, ground truth were used to test the accuracy. We also implemented a modified Tsai's algorithm, so that comparisons can be made between our method and Tsai's method.

A. Simulations

1) Setup: We simulated a robot end-effector mounted with two hand cameras, with their internal and external parameters chosen similar to those in a real environment. The camera base line is 54 mm; the effective focal length 500 pixels. The distance between the origin of each camera coordinate system and that of the end-effector system is 68 mm. Two world points, 16 mm apart and 180 mm in front of the cameras, are used as the calibration points. (The knowledge about the object size and position is not used in the calibration.) Throughout the simulations, we assume the motion uncertainties of the robot to be: for translation, $\sigma_T = 0.05$ mm, and for rotation, $\sigma_R = 0.03^\circ$, all along the Y_g axis.

2) How Reliable is the Gauss–Markov Theorem?: The Gauss–Markov theorem provides accuracy assessment in the absence of ground truth. It was originally developed for linear problems. Its validity in nonlinear problems, such as camera calibration, depends on how well the calibration equation can be locally linearized at the point of solution. We test its validity in terms of Monte-Carlo simulation (1000 runnings of the calibration). Besides the motion uncertainties of the robot as specified above, Gaussian noises were added to both the u and v image coordinates, with $\sigma_u = \sigma_v = 0.5$ pixels. Three lens distortion models were simulated, which are: M_0—no distortion; M_1—radial distortion; and M_2—radial + tangential distortions. We first check the mean values of the estimated parameters, since zero-mean error is the essential assumptions under which the Gauss–Markov theorem can be used to estimate the variances. It was found that 1) for correct parameterization, i.e., the use of the correct distortion model to the data, the estimation errors are approximately zero-mean; 2) for over-parameterization, most of the errors are zero-mean, except for the image center u_0, v_0 and the focal length f_x and f_y, whose estimates are slightly biased. Then, we compare the variances estimated by the Gauss–Markov theorem to those estimated by the Monte Carlo simulation. Table I shows an example, in which the three models M_0, M_1, and M_2 were applied to the image data generated by model M_0. From the table, the following can be observed. First, in the case of correct parameterization, the Gauss–Markov theorem does provide reliable estimates of the parameter variances; while in the case of over-parameterization, it overestimates the variances of the world points and the effective focal length. Second, over-parameterization significantly increases the variances of the hand–eye rotation parameters and that of the image center.

Another situation worth mentioning is that of underparameterization, e.g., the use of a no-distortion model to the radial distortion data. It was found that the parameters estimated in the case of underparameterization are heavily biased. Thus, the Gauss–Markov theorem in this case tells nothing about the accuracy of the estimates since no information about mean values is provided by the theorem.

3) Model Identification: We have seen that the use of an incorrect model has led to increased parameter uncertainties (in the form of either biased estimates or increased variances). By applying the statistical inference method described in Section III, we are able to identify the correct lens-distortion model under moderate noise

TABLE II
MOTION PLANNING IMPROVES CALIBRATION ACCURACY

variance	σ_X (mm)	σ_Y (mm)	σ_Z (mm)	σ_α (°)	σ_β (°)	σ_γ (°)	$\sigma_{t_{cg,1}}$ (mm)
initial calibration	0.32	0.29	0.24	0.00055	0.00150	0.00168	0.30
improved calibration	0.11	0.11	0.10	0.00021	0.00050	0.00064	0.16
variance	$\sigma_{t_{cg,2}}$ (mm)	$\sigma_{t_{cg,3}}$ (mm)	σ_{u_0} (pixels)	σ_{v_0} (pixels)	σ_{f_x} (pixels)	σ_{f_y} (pixels)	
initial calibration	0.30	0.32	0.80	0.99	1.05	1.16	
improved calibration	0.16	0.12	0.28	0.37	0.50	0.42	

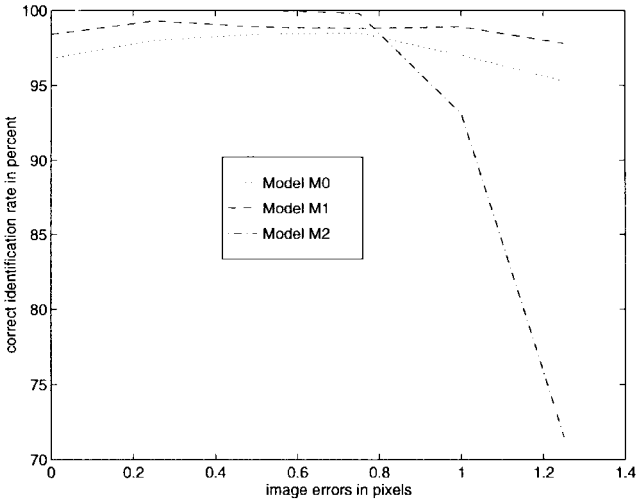


Fig. 4. The rate of correct model identification by the F-distribution.

levels. We tested the identification of the three distortion models under various image noises (up to 1.25 pixels). (Robot motion uncertainties were always assumed.) In the simulation, the maximum radial distortion in model M_1 is 6.6 pixels; while the maximum radial and tangential distortions in model M_2 are 12.0 and 2.3 pixels, respectively. A significance level of 0.99 ($\alpha = 0.01$) is chosen for testing the F-distribution.[3] Fig. 4 shows the identification results. It can be seen that for noise levels within 20% of the distortion components, the rate of correct identification is higher than 97%. The statistics were all gathered from 1000 Monte Carlo simulations. For noise levels which are too large, it is imaginable that the method cannot distinguish between systematic errors and random errors.

4) Initial Calibration versus Improved Calibration: After the identification of the correct lens-distortion model, further improvements in accuracy can be achieved by motion planning. The following gives an example for the M_0 model. An initial calibration was first performed by using data obtained from simple translation along and rotation around the coordinate axes of the end-effector. A total of 143 stations were generated. Gaussian image noises were added, with $\sigma_u = \sigma_v = 0.5$ pixels. After the initial calibration, the obtained parameters are used to plan the robot motion using the method in Section III-B for the final calibration. The variances in both the initial and the planned calibrations are listed in Table II. It can be seen from the table that a reduction of variance by as much as a factor of 3 has been achieved.

5) Other Implementation Details:

a) Joint calibration or separate calibration: The two cameras were calibrated separately, although it is possible to calibrate the two cameras jointly because of the *same* object points observed. In the latter case, correspondences of image points should be established across different cameras. The advantage of joint calibration is that it may further reduce the parameters' variances. It was found, however, that joint camera calibration have some numerical instabilities.

b) Number of iterations and speed: In most cases, the number of iterations are within ten. With the reduction scheme described in the Appendix, considerable run time can be saved. For the case of two object points in our experiments, a saving of about 50% is achieved. In the case of more object points, more savings are expected.

c) Motion planing: In planning camera motions, the following factors are considered:

1) The number of depth levels for a calibration: it should be at least 2; we use 3 to 5.
2) The number of translational motion stations at each depth level: experience shows that at least 25 image points should be obtained at each depth level.
3) The maximum rotation angles in our experiments (in both simulated and real cases) are (refer to Fig. 3), for γ: $\gamma_{\max} = 15°$, increment $\Delta\gamma = 5°$; and for α: $\alpha_{\max} = 360°$, increment $\Delta\alpha = 45°$.

The maximum amount of translations in the camera's $X_c - Y_c$ plane is determined automatically by the initially calibrated internal parameters, object coordinates, and the image size.

B. Comparison with Tsai's Method

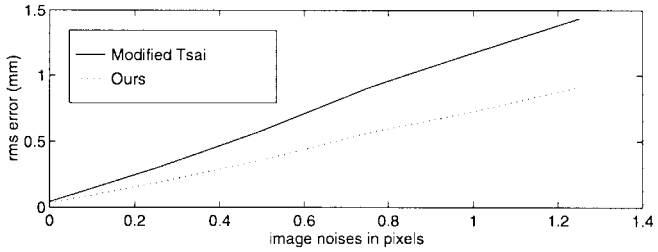
Tsai [22] and Tsai and Lenz [23] have proposed efficient methods for camera calibration in the case of radial distortion. Though metric calibration fields were used in their work, the method can be adopted in our framework for camera calibration and hand–eye calibration, since the robot translations generate metric virtual points up to an unknown translation. The difference here is that their method can only consider one object point due to the unknown relative positions between multiple points. In implementing their methods, we made the extensions as follows. First, the initial image center is assumed to be at the apparent image center. Then the RAC (radial alignment constraint) [22], [23] is used to estimate part of the external parameters. The obtained parameters are then used in the perspective equations for global optimization, which estimates the image center, focal lengths, the radial distortion parameter, and external parameters *simultaneously*. This implementation provides more accurate results than does the original method. The original method estimates the parameters in two sequential stages, which has been found to be error-prone [26]. Besides, the method in [23] for estimating image center relies on the existence of significant lens distortions. The modified method overcomes these drawbacks. After the cameras are calibrated, the hand–eye translation parameters are computed from robot rotations by using (8).

We used the same calibration data generated by model M_1 (radial distortion) to run both Tsai's method and ours. Noises of different levels were added to the image coordinates. Instead of checking each individual parameters obtained by the two methods, we use the

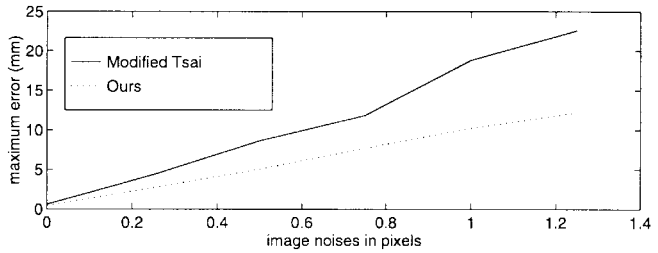
[3] For $\alpha = 0.01$, $m > 120$, $F_{1-\alpha/2}(2, m) = 4.97$, $F_{1-\alpha/2}(3, m) = 4.12$ [8].

TABLE III
CAMERA PARAMETERS' VARIANCES IN REAL EXPERIMENTS

variance	σ_X	σ_Y	σ_Z	σ_α	σ_β	σ_γ	$\sigma_{t_{cg,1}}$	$\sigma_{t_{cg,2}}$
initial calibration	1.44	1.58	0.65	0.0019	0.0072	0.0087	0.21	0.20
improved calibration	0.41	0.31	0.31	0.00060	0.0020	0.0017	0.24	0.19
variance	$\sigma_{t_{cg,3}}$	σ_{u_0}	σ_{v_0}	σ_{f_x}	σ_{f_y}	σ_{p_1}	σ_{p_2}	σ_{k_1}
initial calibration	0.55	3.13	6.13	3.95	4.86	3.55e-6	5.78e-6	4.01e-8
improved calibration	0.31	0.89	1.35	1.03	1.51	1.31e-6	1.57e-6	1.02e-8



(a)



(b)

Fig. 5. Comparison of our method with Tsai's method in 3-D reconstruction: (a) rms error and (b) maximum error.

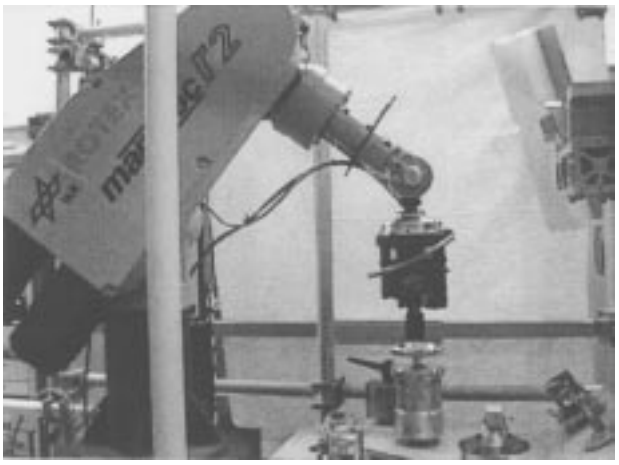


Fig. 6. The experimental environment.

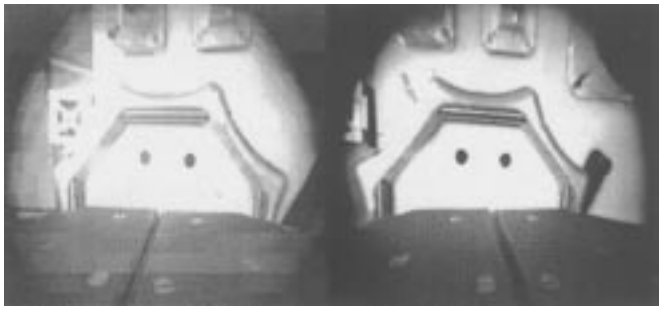


Fig. 7. The images of the ORU from the hand cameras. The two black blobs are used as calibration points.

calibrated parameters to make 3-D reconstruction in the end-effector coordinate system. A total of 172 test points were used, and for each noise level, error statistics for 3-D reconstruction were obtained from 1000 runnings of the calibrations. The rms errors and maximum errors of the two methods are shown in Fig. 5. It can be seen from the figure that our method outperforms Tsai's method in accuracy; for instance, at the noise level $\sigma_u = \sigma_v = 0.5$ pixels, the rms and maximum errors are, for Tsai's method, 0.58 and 8.6 mm; for our method, 0.36 and 5.0 mm, respectively. Computationally, our method is more expensive despite the use of the reduction scheme, since it involves the estimation of multiple object coordinates.

C. Real Experiments

The calibration methods have been implemented in a real robot environment. The robot is Manutec R2, with a repeatability of 0.05 mm for translational motion (unknown accuracy for rotation); The gripper is a multisensory one developed at the authors' institution (with 2 tiny Teli CS6100P cameras in the front) [9]. The focal length, resolution, and visual angle of the cameras are 3.8 mm, 460 (H) $\times$ 420 (V) TV lines, and 90°, respectively. Fig. 6 shows the experimental environment, in which the object beneath the gripper, called the Orbital Replaceable Unit (ORU), is used for camera calibration. Fig. 7 shows a view of the ORU from the stereo hand cameras, where the two black blobs artificially marked on the object are used to ease feature detection. A Datacube MaxVideo 200 image processing system is used to track the feature points in real time during robot motion. The tracking algorithm is started with blob searching, which consists of a) local thresholding with the threshold adapted to local intensity statistics; b) ellipse edge-fitting and residual testing; and c) computation of the center of gravity of the binarized blobs. After the blobs are found, they are tracked by local binarization and computation of region of interest in the next cycle. The center of gravity of the binarized blob is used as the image coordinates in calibration. All the tracking computations are realized by the DataCube hardware. (We have recently eased the tracking task by using colored blobs for real-time color segmentation [28].) The calibration distance of the objects to the camera is approximately 185 mm. Model identification indicates that the cameras contain both radial and tangential distortions. Motion planing is carried out based on an initial calibration. The estimated parameter variances by the method in the Appendix for one of the cameras in both the initial and the improved calibrations are shown in Table III. The improvement by the planned calibration over the initial one can be obviously seen. The image coordinate variance is estimated to be 0.66 pixels.

As another test of the calibration accuracy, the distance between the two blobs was measured at various positions and orientations of the robot. From 200 measurements, the rms error and maximum error for our method are found to be 0.19 and 0.61 mm, respectively; while for Tsai's method, the rms and maximum errors are 0.30 and

0.90 mm, respectively. We also tested the accuracy in 3-D space by making 3-D reconstruction at one station and predicting the 3-D coordinates according to robot motions. The distances between the predicted and the computed 3-D points are computed. The rms and maximum errors in the 3-D case are, for our method, 0.74 and 2.11 mm; for Tsai's method, 1.05 and 3.25 mm, respectively; The average depth of the object points in the camera for the test is about 135 mm. Notice that these errors have included the robot motion errors.

V. Conclusions

In this paper, we have presented a fully automatic method for calibrating cameras mounted on a robot hand. By tracking a set of image points (as few as one point) in the camera, the camera's internal parameters (including distortion coefficients) and the mounting parameters can be calibrated. Metric reconstruction of the object points are produced at the same time. The method is active not only in the sense that the camera is able to move, but also in that the motion can be designed to raise robustness of the results. The model identification method has been found to contribute to the accuracy improvements, too. The use of the reduction scheme has substantially speeded up the calibration procedure, especially when more object points are involved. Experiments show that our method is more accurate and can be used in real robotic applications.

Appendix
The Gauss–Markov Theorem and the Reduction Scheme

In this Appendix, we first outline the Gauss–Markov Theorem in the usual sense for variance estimation. Then, we describe a reduction scheme for solving the normal equations of bordered block-diagonal structure and derive the corresponding Gauss–Markov Theorem.

A. Gauss–Markov Theorem

Suppose y is the dependent variable and $\mathbf{a} = (a_1, a_2, \cdots, a_m)^T$ the m independent variables in a linear regression model $y = \mathbf{x}^T\mathbf{a}$, where $\mathbf{x} = (x_1, x_2, \cdots, x_m)^T$ are the unknown model parameters; Suppose that the observations y_i under the independent variables $\mathbf{a}_i$, $i = 1, 2, \cdots, n$, are uncorrelated, and that the observation errors are zero-mean Gaussians of the same variance. Then, in the minimum variance estimation of the model parameters $\mathbf{x}$, the covariance matrix of $\mathbf{x}$ can be estimated as

$$\mathbf{V}_{\mathbf{x}} = (A^T A)^{-1}\sigma_y^2 \tag{20}$$

where A is the stacked matrix of $\mathbf{a}_i^T$'s. The matrix $(A^T A)^{-1}$ is the normal matrix in the normal equations; and σ_y^2 is the variance estimate of the observations

$$\sigma_y^2 = \left(\sum_i e_i^2\right)/(n-m) \tag{21}$$

where e_i's are the residuals of the observation equations. From $\mathbf{V}_{\mathbf{x}}$, the variances and the correlation coefficients of the estimates can be computed.

B. Reduction Scheme

To solve normal equations of the form (15), we can first express $\delta\mathbf{x}_i$ in (15) as

$$\delta\mathbf{x}_i = A_i^{-1}(\mathbf{e}_i - B_i\delta\mathbf{x}_0); \qquad i = 1, 2, \cdots, N. \tag{22}$$

Then we insert the above expression into the last block equation in (15) to get the solution for $\delta\mathbf{x}_0$ as

$$\delta\mathbf{x}_0 = \left(B_0 - \sum_{i=1}^{N} B_i^T A_i^{-1}\right)^{-1}\left(\mathbf{e}_0 - \sum_{i=1}^{N} B_i^T A_i^{-1}\mathbf{e}_i\right). \tag{23}$$

After $\delta\mathbf{x}_0$ has been determined, $\delta\mathbf{x}_i$ are computed from (22). The whole procedure involves only one matrix inversion of size $m \times m$ and N matrix inversions of size 3×3.

C. Gauss–Markov Theorem in the Reduction Case

Here we just give the formulas for the variance estimation of $\delta\mathbf{x}_i$ and $\delta\mathbf{x}_0$, without going into the details of the derivations.

Define

$$D = \left(B_0 - \sum_{i=1}^{N} B_i^T A_i^{-1}\right)^{-1}. \tag{24}$$

The covariance matrix for $\delta\mathbf{x}_0$ is

$$\mathbf{V}_{\delta\mathbf{x}_0} = D\sigma_e^2 \tag{25}$$

where σ_e^2 is the estimated variance of the measurement based on (21). The covariance matrix for $\delta\mathbf{x}_i$ is

$$\mathbf{V}_{\delta\mathbf{x}_i} = (A_i^{-1} + A_i^{-1}B_i D B_i^T A_i^{-1})\sigma_e^2. \tag{26}$$

The covariance between $\delta\mathbf{x}_i$ and $\delta\mathbf{x}_i$ is

$$\mathrm{Cov}\,(\mathbf{x}_i, \mathbf{x}_j) = A_i^{-1}B_i D B_j^T A_j^{-1}\sigma_e^2 \tag{27}$$

while the covariance between $\delta\mathbf{x}_i$ and $\delta\mathbf{x}_0$ is

$$\mathrm{Cov}\,(\mathbf{x}_i, \mathbf{x}_0) = A_i^{-1}B_i D\sigma_e^2. \tag{28}$$

The above equations are used to estimate the variances of the parameters after convergence is reached.

Acknowledgment

The authors are very grateful to the anonymous reviewers for their valuable comments.

References

[1] A. Basu, "Active calibration of cameras: Theory and implementation," *IEEE Trans. Syst., Man, Cybern.*, vol. 25, pp. 256–265, 1995.
[2] D. J. Bennett, D. Geiger, and J. M. Hollerbach, "Autonomous robot calibration for hand–eye coordination," *Int. J. Robot. Res.*, vol. 10, no. 5, pp. 550–559, 1991.
[3] F. Du and M. Brady, "Self-calibration of the intrinsic parameters of cameras for active vision system," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 1993, pp. 477–482.
[4] I. W. Faig, "Calibration of close-range photogrammetric systems: Mathematical formulation," *Photogramm. Eng. Remote Sens.*, vol. 41, pp. 1479–1486, 1975.
[5] O. D. Faugeras and G. Toscani, "The calibration problem for stereo," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 1986, pp. 15–20.
[6] O. D. Faugeras, O.-T. Luong, and S. J. Maybank, "Camera self-calibration: Theory and experiments," in *Proc. Europ. Conf. Computer Vision*, 1992, pp. 321–334.
[7] W. Förstner, "Reliability analysis of parameter estimation in linear models with application to mensuration problems in computer vision," in *Proc. Sec. Int. Workshop on Robust Computer Vision*, Institut für Photogrammetrie, Unversität Bonn, Germany, Mar. 1992, pp. 1–110.
[8] D. M. Himmelblau, *Process Analysis by Statistical Methods.* New York: Wiley, 1970.
[9] G. Hirzinger, B. Brunner, J. Dietrich, and J. Heindl, "Sensor-based space robotics-ROTEX and its telerobotic features," *IEEE Trans. Robot. Automat.*, vol. 9, pp. 649–663, Oct. 1993.
[10] A. Izaguirre, P. Pu, and J. Summer, "A new development in camera calibration: Calibrating a pair of mobile cameras " *Int. J. Robot. Res.*, vol. 6, pp. 104–115, 1987.
[11] J. F. Kenefick, M. S. Gyer, and B. F. Harp, "Analytical self-calibration," *Photometric Eng.*, pp. 1117–1126, 1972.
[12] O. Kölbl, "Selbstkalibrierung von Aufnahmekammern," *Bildmessung und Luftbildwessen*, vol. 1, pp. 31–37, 1972.
[13] P. L. Liang, Y. L. Chang, and S. Hackwood, "Adaptive self-calibration of robot systems," *IEEE Trans. Syst., Man, Cybern.*, vol. 19, no. 4, 1989.

[14] O. T. Luong and O. D. Faugers, "Self-calibration of a camera using multiple images," in *Proc. Int. Conf. Pattern Recognition*, 1992, pp. 9–12.
[15] ____, "An optimization framework for efficient self-calibration and motion estimation," in *Proc. Int. Conf. Pattern Recognition*, 1994, pp. 248–252.
[16] S. D. Ma, "A self-calibration technique for active vision system," *IEEE Trans. Robot. Automat.*, vol. 12, Feb. 1996.
[17] *Manual of Photogrammetry*, 4th ed. Bethesda, MD: Amer. Soc. Photogrammetry, 1980.
[18] R. Mohr, L. Quan, and F. Veillon, "Relative 3D reconstruction using multiple uncalibrated images," *Int. J. Robot. Res.*, vol. 14, no. 6, pp. 619–632, 1995.
[19] W. H. Press *et al., Numerical Recipes in C: The Art of Scientific Computing*., 2nd ed. Cambridge, U.K.: Cambridge Univ. Press, 1992.
[20] C. R. Rao, *Linear Statistical Inference and Its Applications*, 2nd ed. New York: Wiley, 1973.
[21] Y. C. Shiu and S. Ahmad, "Calibration of wrist-mounted robotic sensors by solving homogeneous transformation equations of the form $AX = XB$," *IEEE Trans. Robot. Automat.*, vol. 5, pp. 16–29, Feb. 1989.
[22] R. Y. Tsai, "A versatile camera calibration technique for high-accuracy 3-D machine vision metrology using off-the-shelf TV cameras and lenses," *IEEE J. Robot. Automat.*, vol. RA-3, pp. 323–344, 1987.
[23] R. Lenz and R. Y. Tsai, "Techniques for calibration of the scale factor and image center for high accuracy 3-D machine vision metrology," *IEEE Trans. Pattern Anal. and Machine Intell.*, vol. 10, no. 5, pp. 713–720, 1988.
[24] R. Y. Tsai and R. K. Lenz, "A new technique for fully autonomous and efficient 3-D robotics hand/eye calibration," *IEEE Trans. Robot. Automat.*, vol. 5, pp. 345–358, June 1989.
[25] C. C. Wang, "Extrinsic calibration of a vision sensor mounted on a robot," *IEEE Trans. Robot. Automat.*, vol. 8, pp. 161–175, Apr. 1992.
[26] G.-Q. Wei and S. D. Ma, "Implicit and explicit camera calibration: Theory and experiments," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 16, no. 5, pp. 469–480, 1994.
[27] G.-Q. Wei, K. Arbter, and G. Hirzinger, "Active self-calibration of robotic eyes, hand–eye relationships and the application to sensor-based robot positioning," German Aerospace Research Establishment, Oberfaffenhofen, Tech. Rep., Jan. 1995.
[28] ____, "Real-time visual servoing for laparoscopic surgury," *IEEE Eng. Med. Biol.*, vol. 16, pp. 40–45, Jan./Feb. 1997.
[29] J. Weng, P. Cohen, and M. Herniou, "Calibration of stereo cameras using a nonlinear distortion model,"*IEEE Trans. Pattern Anal. Machine Intell.*, vol. 14, 1992.
[30] H. Zhuang and Y. C. Shiu, "A noise-tolerant algorithm for robotic hand–eye calibration with or without sensor orientation measurement," *IEEE Trans. Syst., Man, Cybern.*, vol. 23, pp. 1168–1175, July/Aug. 1993.
[31] H. Zhuang, K. Wang, and Z. S. Roth, "Simultaneous calibration of a robot and a hand-mounted camera," *IEEE Trans. Robot. Automat.*, vol. 11, pp. 649–660, Oct. 1995.

Analysis of Probabilistic Roadmaps for Path Planning

Lydia E. Kavraki, Mihail N. Kolountzakis, and Jean-Claude Latombe

***Abstract*— We provide an analysis of a recent path planning method which uses probabilistic roadmaps. This method has proven very successful in practice, but the theoretical understanding of its performance is still limited. Assuming that a path γ exists between two configurations a and b of the robot, we study the dependence of the failure probability to connect a and b on:**

1) **the length of γ;**
2) **the distance function of γ from the obstacles;**
3) **the number of nodes N of the probabilistic roadmap constructed.**

Importantly, our results do not depend strongly on local irregularities of the configuration space, as was the case with previous analysis. These results are illustrated with a simple but illuminating example. In this example, we provide estimates for N, the principal parameter of the method, in order to achieve failure probability within prescribed bounds. We also compare, through this example, the different approaches to the analysis of the planning method.

***Index Terms*— Probabilistic roadmaps, randomized algorithms, robot path planning.**

I. Introduction

Motion planning has been an active area of research during the last two decades [18]. The problem has gained increasing attention because of the larger number of potential applications (e.g. robotics, manufacturing, computer-assisted surgery, molecular biology). Several recent papers describe path planners that can deal with robots that have more than 3 or 4 degrees of freedom (DOF) and move in realistic environments (for a survey see [12]). Because of the high computational complexity of path planning, practical planners employ different kinds of heuristics to guide the search of the robot from its initial to its final position [1]–[3], [5], [8]–[11], [13], [16], [20], [21].

This short paper considers the success of a class of probabilistic algorithms for path planning [11]–[14], [20], [21] and tries to establish a framework for the theoretical understanding of their results. Our ultimate goal is to further enhance the performance of these methods by evaluating the role of their parameters and by finding ways to choose good values for these parameters. We will restrict ourselves to the description of the planner in [12]–[14] for a concise presentation of the algorithm and our results. We hereafter refer to this planner as probabilistic roadmap planner (PRM).

PRM builds probabilistic roadmaps in the free configuration space (C-space) [19] of the robot by generating and interconnecting a large number of (mostly) random configurations of the robot. These roadmaps are used for answering path planning queries: given an initial and final configuration of the robot, PRM connects them to the roadmap by simple paths (e.g. straight segments) and then searches

Manuscript received January 31, 1996; revised November 12, 1996. This work was supported in part by ARPA Grant N00014-94-1-0721 and NSF Grant DMS-9304580. This paper was recommended for publication by Associate Editor M. Peshkin and Editor S. Salcudean upon evaluation of the reviewers' comments. This paper was presented in part at the 1996 International Conference on Robotics and Automation.

L. E. Kavraki is with the Department of Computer Science, Rice University, Houston, TX 77005 USA.

M. N. Kolountzakis is with the Department of Mathematics, University of Illinois at Urbana-Champaign, Urbana, IL 61801 USA.

J.C. Latombe is with the Robotics Laboratory, Department of Computer Science, Stanford University, Stanford, CA 94305 USA.

Publisher Item Identifier S 1042-296X(98)01433-5.