Self Calibration of a Stereo Head Mounted onto a Robot Arm *

R. Horaud, F. Dornaika, B. Boufama, and R. Mohr

LIFIA & Inria Rhône-Alpes, 46, avenue Félix Viallet, 38031 Grenoble FRANCE

Abstract. In this paper we propose a new method for solving the hand-eye calibration problem and we show how this method can be used in conjunction with a reconstruction technique in order to estimate on-line the relationship between the frame in which the scene has been reconstructed (or calibration frame) and the frame attached to the robot hand. The method is particularly well suited for calibrating stereo heads with respect to the robot on which they are mounted. We discuss the advantage of on-line (self) versus off-line hand-eye and camera calibrations. We develop two solutions for solving for the hand-eye calibration problem, a closed-form solution and a non-linear least-squares solution. Finally we report on some experiments performed with a stereo head mounted onto a 6 degrees of freedom robot arm.

1 Introduction and motivation

Whenever a sensor is mounted onto a robot hand (or a gripper) it is important to know the relationship between the sensor frame and the hand frame. The problem of determining this relationship is refered to as the hand-eye calibration problem. In the particular case of the sensor being a single camera, the hand-eye calibration problem is equivalent to the problem of solving a homogeneous matrix equation of the form:

$$AX = XB \tag{1}$$

In this equation, X is the unknown hand-eye relationship, A is the camera motion, and B is the hand motion. Matrix B is generally provided by the direct kinematic model of the robot arm. The classical way of estimating A is to determine the pose of the camera (position and orientation) with respect to a fixed calibration object expressed in its own frame – the calibration frame. Let, for example, A_1 and A_2 be two matrices associated with two different camera positions. Then A is simply given by:

$$A = A_2 A_1^{-1} \tag{2}$$

In the past, some solutions were proposed for solving eq. (1), among others, by Tsai & Lenz [6] and Horaud & Dornaika [4]. While in most of the previous approaches standard linear algebra techniques are used, in [4] we noticed that there

* This work has been supported by the Esprit programme through the SECOND project (Esprit-BRA No. 6769).

Lecture Notes in Computer Science, Vol. 800
Jan-Olof Eklundh (Ed.)
Computer Vision – ECCV '94

are in fact two solution classes: (i) closed-form solutions if rotation is estimated first, independently of the translation and (ii) non-linear least-squares solutions if rotation and translation are estimated simultaneously. The latter class of solutions is numerically more robust than the former. Moreover, uniqueness analysis of the hand-eye geometry allows one to conclude that any Newton-like non-linear minimization method is likely to converge to the good solution.

In this paper we propose a new formulation for the hand-eye calibration problem. We show that this new formulation is somehow more general than the classical one since it can be used either off-line (as in the classical case) or on-line. On-line hand-eye calibration may well be viewed as a self calibration method since neither prior camera calibration nor a specific calibration object are needed. The self calibration method that we propose here has strong links with recently developped tools in camera self calibration and Euclidian reconstruction with uncalibrated cameras [2], [1], [3]. More specifically, we will make use of the fact that turning a projective reconstruction into an Euclidian one provides camera calibration as a side effect. Such an on-line camera calibration method will provide, together with on-line knowledge about the kinematic position of the robot arm, the bases for performing hand-eye self calibration.

In particular, with our new formulation, the problem of calibrating a stereo head mounted onto a robot yields an elegant solution. Moreover, self calibration is well suited for stereo head with variable geometry. Indeed, for such heads, the relationship between the head frame and the hand frame may vary and hence, off-line calibration is not very useful. Although there are many stereo head prototypes around, only a few of them are actually mounted onto a 6 degrees of freedom robot arm. The advantage of a robot-mounted stereo head is that the head has much more mobility and flexibility than if it lied onto a fixed platform. Therefore it seems reasonable to investigate ways to determine on-line the relationship between the head frame and the robot frame.

2 Problem formulation

We consider a classical pin-hole camera model. We recall that calibrating such a camera is equivalent to estimating the projective transformation between a 3-D frame and the 2-D image frame. Let M be a 3×4 matrix describing such a projective transformation. We have:

$$p = M\ P \tag{3}$$

where $p = (su\ sv\ s)^T$ is an image point with coordinates u and v, s is a scale factor, and $P = (x\ y\ z\ 1)^T$ is a 3-D point expressed in the frame in which the camera is to be calibrated.

We farther assume that the camera is rigidly mounted onto a robot gripper and that there is a cartesian frame associated with this gripper. Although it is impractical, it is theoretically possible to choose a calibration frame identical with the gripper frame: This means that the 3-D calibrating points are in fact

expressed in the gripper frame. Since the gripper is rigidly attached to the camera, the calibration thus obtained, i.e., matrix M, remains invariant with respect to robot motion.

In particular, we consider two different positions of the robot with respect to a fixed *true* calibration frame. Let Y_i $(i = 1, 2)$ be the transformation from the gripper frame to the true calibration frame , e.g., Figure 1. Obviously we have from eq. (3):

$$p = MY_1^{-1}\, Y_1 P = MY_2^{-1}\, Y_2 P = M\ P \tag{4}$$

In these equations $Y_1 P$ and $Y_2 P$ represent *the same* calibration point expressed in the (true) calibration frame. Moreover: $M_1 = MY_1^{-1}$ and $M_2 = MY_2^{-1}$ are the 3×4 projection matrices between the calibration frame and the camera in positions 1 and 2. With these notations we obtain immediately from eq. (4):

$$M_1 Y_1 = M_2 Y_2 \tag{5}$$

Recall that B is the gripper motion between two arm positions. We have (see Figure 1): $Y_2 = Y_1 B^{-1}$ and by substituting in eq. (5) and with the notation $Y = Y_1$ we finally obtain:

$$M_2 Y = M_1 Y B \tag{6}$$

This equation is the new formulation for the hand-eye calibration problem that does not make explicit the intrinsic and extrinsic camera parameters. The unknown Y is the transformation from the gripper frame to the calibration frame (or to any world frame in the case of on-line calibration) when the camera is in its first position, e.g., Figure 1. The projection matrices M_1 and M_2 may be obtained either off-line using a calibrating object or on-line using a method that will be briefly outlined in section 4.

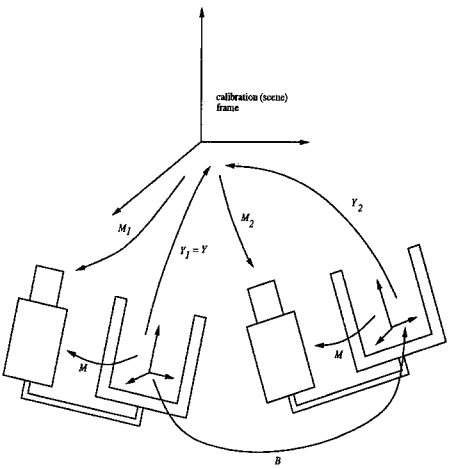

Fig. 1. This figure shows the relationship between a calibration (or scene) frame and two positions of the hand-eye device. The camera may well be calibrated with respect to either a gripper frame (M) or a scene frame (M_1 and M_2).

2.1 Relationship with the classical formulation

There is a very simple relationship between eq. (6) and eq. (1) that will be outlined in this section. It is well known that a projection matrix M_i decomposes into intrinsic and extrinsic camera parameters:

$$M_i = CA_i = \begin{pmatrix} \alpha_u & 0 & u_0 & 0 \\ 0 & \alpha_v & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} R_A^i & t_A^i \\ 0 & 1 \end{pmatrix} \tag{7}$$

Matrix C characterizes the camera and the projection, and matrix A_i characterizes the position and orientation of the camera with respect to the calibration frame. Here we assume that the intrinsic camera paramters do not change during the calibration process [3]. By substituting $M_i = CA_i$ in eq. (6) we obtain $CA_2Y = CA_1YB$ and it is straightforward to figure out that this equation reduces to: $A_2Y = A_1YB$

The relationship between X (in eq. (1)) and Y (in eq. (6)) is:

$$X = A_1Y \tag{8}$$

By substituting in the equation above and using eq. (2) we finally obtain eq. (1). The advantage of the new formulation over the latter equation is that one need not make explicit the intrinsic and extrinsic camera parameters. Many authors have noticed that, even in the case of a very precise camera calibration, the decomposition of the projective transformation into intrinsic and extrinsic parameters is numerically unstable [5].

2.2 The case of a stereo head

As alredy mentioned, another advantage of our formulation with respect to the classical one is that it allows an elegant extension to the case of a stereo head mounted onto the robot arm:

- With the classical formulation two independent equations need to be solved, that is, $AX = XB$ for the left camera, and $A'X' = X'B$ for the right camera.
- With the new formulation both cameras contribute to the same unknown. Indeed we have $M_2Y = M_1YB$ for the left camera and $M_2'Y = M_1'YB$ for the right camera. Hence the left camera and right camera calibrations are fused into a unique calibration problem. Notice that this may be easily generalised to any number of cameras rigidly mounted onto the robot arm.

3 Problem solution

In this section we show that the new formulation has a mathematical structure that allows one to solve the problem either in closed form or by using Newton-like non-linear least-squares minimization methods.

Notice that a projection matrix M_i can be written as:

$$M_i = \begin{pmatrix} N_i & n_i \end{pmatrix}$$

where N_i is a 3×3 matrix and n_i is a 3-vector. It is well known that N_i has rank 3. This can be easily observed from the decomposition of such a matrix into intrinsic and extrinsic parameters. With this notation eq. (6) may be decomposed into a matrix equation[2]:

$$N_2 R_Y = N_1 R_Y R_B \tag{9}$$

and a vector equation:

$$N_2 t_Y + n_2 = N_1 R_Y t_B + N_1 t_Y + n_1 \tag{10}$$

Introducing the notation: $N = N_1^{-1} N_2$, eq. (9) becomes:

$$N R_Y = R_Y R_B \tag{11}$$

Two properties of N may be easily derived: N is the product of three rotation matrices, it is therefore a rotation itself and since R_Y is an orthogonal matrix, the above equation defines a similarity transformation. It follows that N has the same eigenvalues as R_B. In particular R_B has an eigenvalue equal to 1 and let n_B be the eigenvector associated with this eigenvalue.

If we denote by n_N the eigenvector of N associated with the unit eigenvalue, then we obtain:

$$N R_Y n_B = R_Y R_B n_B = R_Y n_B \tag{12}$$

and hence we have:

$$n_N = R_Y n_B \tag{13}$$

By premultiplying eq. (10) with N_1^{-1} we obtain:

$$(N - I) t_Y = R_Y t_B - t_N \tag{14}$$

with: $t_N = N_1^{-1}(n_2 - n_1)$.

To summarize, the new formulation decomposes into eqs. (13) and (14) which are of the form:

$$v' = Rv \tag{15}$$

$$(K - I)t = Rp - p' \tag{16}$$

where R and t are the parameters to be estimated (rotation and translation), v', v, p', p are 3-vectors, K is a 3×3 rotation matrix and I is the 3×3 identity matrix.

Eqs. (15) and (16) are associated with one motion of the hand-eye device. In order to estimate R and t at least two such motions are necessary. In the general case of n motions one may cast the problem of solving $2n$ such equations into the problem of minimizing two positive error functions:

$$f_1(R) = \sum_{i=1}^{n} \|v_i' - Rv_i\|^2 \tag{17}$$

and

$$f_2(R, t) = \sum_{i=1}^{n} \|Rp_i - (K_i - I)t - p_i'\|^2 \tag{18}$$

Therefore, two approaches are possible:

[2] R_B and t_B are the rotation matrix and translation vector associated with the rigid displacement B.

1. *R then t.* Rotation is estimated first by minimizing f_1. This minimization problem has a simple closed-form solution [4]. Once the optimal rotation is determined, the minimization of f_2 over the translational parameters is a linear least-squared problem.
2. *R and t.* Rotation and translation are estimated simultaneously by minimizing $f_1 + f_2$. This minimization problem is non-linear but it provides the most stable solution [4].

4 Camera self calibration

In this section we describe a method for estimating a set of n projection matrices with a camera mounted onto a robot arm. Camera self calibration is the task of computing these projection matrices by observing an unknown scene and not a calibration pattern. We consider k points of the scene $P_1, \ldots P_j, \ldots P_k$ and let p_{ij} denote the projection of P_j onto the i^{th} image, that is, when the camera and the gripper are in position i. With the same notations as in section 2 one may write:

$$p_{ij} = M_i P_j \quad (i = 1 \ldots n,\ j = 1 \ldots k) \tag{19}$$

Therefore each scene point P_j is observed through its projections $p_{1j}, \ldots p_{nj}$ which in practice have to be tracked in the image sequence.

For each measurement, i.e., for each image point, eq. (19) can be written as a set of two constraints:

$$\begin{cases} u_{ij} = \dfrac{m_{11}^{(i)} x_i + m_{12}^{(i)} y_i + m_{13}^{(i)} z_i + m_{14}^{(i)}}{m_{31}^{(i)} x_i + m_{32}^{(i)} y_i + m_{33}^{(i)} z_i + m_{34}^{(i)}} \\ v_{ij} = \dfrac{m_{21}^{(i)} x_i + m_{22}^{(i)} y_i + m_{23}^{(i)} z_i + m_{24}^{(i)}}{m_{31}^{(i)} x_i + m_{32}^{(i)} y_i + m_{33}^{(i)} z_i + m_{34}^{(i)}} \end{cases} \tag{20}$$

Since we have k points and n images we obtain $2 \times n \times k$ such constraints. Each projection matrix is defined up to a scale factor, so by setting $m_{34}^{(i)} = 1$, we are left with $11 \times n$ unknowns associated with the projection matrices and $3 \times k$ unknowns associated with the coordinates of the scene points. For example for 10 images and 50 scene points we have $2 \times 10 \times 50 = 1000$ constraints and $11 \times 10 + 3 \times 50 = 260$ unknowns. So if n and k are large enough we obtain more constraints than unknowns and hence, the problem may be solved by seeking a minimum of the following error function:

$$f(M_1, \ldots M_i, \ldots M_n, P_1, \ldots P_j, \ldots P_k) =$$
$$\sum_{ij} \left(u_{ij} - \frac{m_{11}^{(i)} x_i + m_{12}^{(i)} y_i + m_{13}^{(i)} z_i + m_{14}^{(i)}}{m_{31}^{(i)} x_i + m_{32}^{(i)} y_i + m_{33}^{(i)} z_i + m_{34}^{(i)}} \right)^2 +$$
$$\sum_{ij} \left(v_{ij} - \frac{m_{21}^{(i)} x_i + m_{22}^{(i)} y_i + m_{23}^{(i)} z_i + m_{24}^{(i)}}{m_{31}^{(i)} x_i + m_{32}^{(i)} y_i + m_{33}^{(i)} z_i + m_{34}^{(i)}} \right)^2$$

Several authors implemented solutions for solving this non-linear least-squares minimization problem [1], [3]. Whenever such a solution is found, it is defined

up to a collineation W (a 4×4 invertible matrix). Indeed, for any such matrix W we have (see also eq. (4)): $p_{ij} = M_i W^{-1} W P_j$. One way to fix this collineation is to select 5 algebraically free points which can be used to form a projective basis associated with the scene. The coordinates of these 5 points may be assigned the canonical ones [2]: $(0\ 0\ 0\ 1)^T$ $(1\ 0\ 0\ 1)^T$ $(0\ 1\ 0\ 1)^T$ $(0\ 0\ 1\ 1)^T$ $(1\ 1\ 1\ 1)^T$.

Thus, one obtains by a non-linear least-squares minimization technique a projective reconstruction of the scene, that is, the coordinates of the scene points are expressed with respect to the projective basis just mentioned. The projection matrices are also defined up to a collineation W^{-1} and therefore they are not very useful in general, and in particular for calibrating our stereo head with respect to the robot arm. Therefore, one has to turn the projective data (the scene points and the projective matrices) into Euclidian data. There are several methods to do it but this is beyond the scope of this article. Let us mention that the simplest way to think of this mapping is to assign cartesian coordinates to the 5 points forming the projective basis. Thus, this cartesian frame becomes in fact the scene (or the calibration) frame. The procedure described in this section may well be applied to both cameras composing the stereo head.

5 Experiments and discussion

In order to perform on-line (self) hand-eye calibration, we gathered 9 image pairs with a stereo head mounted onto a robot hand. Three of these images corresponding to the left camera are shown on Figure 2. 28 corners were detected in the first left image and tracked along the sequence.

The same process (corner detection and tracking) was performed with the right image sequence. Notice that only the reference points need be matched between the first left and right images. This is to ensure that the "left" and "right" points are reconstructed with respect to the same scene reference frame. Hence, the non-linear reconstruction algorithm described in section 4 is run twice, first with the left image sequence and second, with the right image sequence. Therefore, two series of projection matrices are provided, one for the left camera and the other for the right camera.

In order to be able to evaluate on-line calibration on a quantitative basis we calibrated off-line, i.e., [4] and we compared the two calibration data sets. We noticed a discrepancy in translation which may be explained by the relatively small camera (or robot) motions during tracking. In the case of off-line calibration the camera motions were quite large. It is wortwhile to notice that in all these experiments (off- and on-line) the robot itself was poorly calibrated. Errors of about 10mm in robot motion were often noticed. Another important feature that may explain the difference between the two calibration processes is the number of points. Indeed, the calibration pattern used off-line has 460 points while the on-line process used only 28 points.

Euclidian reconstruction from uncalibrated cameras is a very recent research topic in computer vision. The experiments that we described in this paper and that we continue to perform allow the validation of such reconstruction tech-

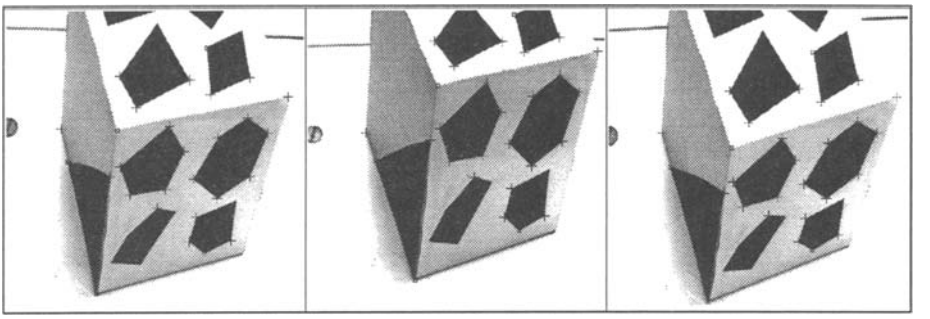

Fig. 2. Three among the 9 images gathered with the left camera (the first, the fourth, and the seventh one). The tracked points are also shown. The 5 reference points are marked with a small square.

niques. It is one thing to see a reconstruction displayed onto a screen and another thing to have it work in a real environment. Therefore we believe that experiments such as those briefly described in this paper are an excellent testbed for any reconstruction method. Indeed, the result of on-line calibration can be easily compared with the result obtained off-line, within a more classical context. However, the latter may well be viewed as the ground-truth and used to validate, through hand-eye calibration, the whole reconstruction process. Ground-truth data are very often missing in computer vision research.

References

1. B. Boufama, R. Mohr, and F. Veillon. Euclidian constraints for uncalibrated reconstruction. In *Proceedings Fourth International Conference on Computer Vision*, pages 466–470, Berlin, Germany, May 1993,. IEEE Computer Society Press, Los Alamitos, Ca.
2. O. D. Faugeras. What can be seen in three dimensions with an uncalibrated stereo rig. In G. Sandini, editor, *Computer Vision - ECCV 92, Proceedings Second European Conference on Computer Vision, Santa Margherita Ligure, May 1992*, pages 563–578. Springer Verlag, May 1992.
3. R. I. Hartley. Euclidian reconstruction from uncalibrated views. In *ESPRIT-ARPA-NSF Workshop on Applications of Invariance in Computer Vision II*, pages 187–201, Ponta Delgada, Azores, October 1993.
4. R. Horaud and F. Dornaika. Hand-eye calibration. In *Proc. Workshop on Computer Vision for Space Applications*, pages 369–379, Antibes, France, September 1993.
5. T. Q. Phong, R. Horaud, A. Yassine, and D. T. Pham. Object pose from 2-D to 3-D point and line correspondences. Technical Report RT 95, LIFIA–IMAG, February 1993. Submitted to the *International Journal on Computer Vision.*
6. R.Y. Tsai and R.K. Lenz. A new technique for fully autonomous and efficient 3D robotics hand/eye calibration. *IEEE Journal of Robotics and Automation*, 5(3):345–358, June 1989.