

# HAND-EYE CALIBRATION USING MULTILINEAR CONSTRAINTS

*H. Stewenius & K. Åström*

Centre for Mathematical Sciences  
Lund Institute of Technology  
SE-221 00 Lund, Sweden  
{stewe, kalle}@maths.lth.se

## Abstract

*The hand-eye calibration problem is a fairly well studied problem. Most methods are related to those proposed by Tsai and Lenz. Here a novel method is introduced that uses multilinear constraints to eliminate the problem of solving for structure and motion. The method is based on tracked interest points. The method is non-iterative involving only low cost operations. The method has been tested on both simulated and real data.*

## 1. INTRODUCTION

This paper presents a novel method to solve the hand-eye calibration problem. In the hand-eye calibration problem a camera is fixed by a rigid transformation to a robot hand. The motion of the robot hand is assumed to be known. The goal is to compute the affine transformation from robot hand to camera as well as the internal parameters of the camera. The problem has been solved in many different ways. Tsai and Lenz solve the problem by solving for camera positions using a well known calibration target [1] and then factoring out the unknown transformation. They also solve the problem of unknown parameters in the robot [2].

Most other authors solves the problem in a similar way by solving for the unknown cameras and then factoring out the affine transformation, Andreff, Horaud and Espiau [3] uses elegant linear formalism to reduce the problem to a linear problem, Dornaika and Horaud [4] uses quaternions to linearize the problem.

Wei, Arbter and Hirzinger [5] use the robot motion to solve for the unknown calibration point and to calibrate the camera, hand-eye translation and lens distortion. They also consider motion planning for the calibration process. Dias et al [6] propose to use linear methods together with a Kalman or a recursive filter to increase stability. Zhuang and Qu [7] propose an improved Jacobian for use in a one stage iterative algorithm based on the reprojection errors. Malm and Heyden deviate strongly from the other methods in that they use a motion field based method instead of a point based

method [8].

In this paper it is proposed how to solve immediately and linearly for the rigid transformations and internal calibration using the multilinear constraints without having to deal with the structure and motion problem and thus reducing system dimensionality to only the desired variables and not to coordinates of all world points. The method is based on tracked interest points which can either come from one point tracked in all images or many points, each appearing in at least two images.

## 2. CAMERA EQUATIONS AND THE MULTILINEAR CONSTRAINT

Working in homogenous coordinates the standard camera projection equation is assumed, c.f. [9]

$$\lambda u = PU, \quad (1)$$

where  $u \in \mathbb{R}^3$  is the image point,  $P \in \mathbb{R}^{3 \times 4}$  the camera matrix,  $U \in \mathbb{R}^4$  the real world coordinates of the point and  $\lambda \in \mathbb{R}$  a constant.

The multilinear constraints [10, 11, 12] give a restriction cameras  $P(t_i)$  and images  $u(t_i)$  when observing one world point at different times  $t_i$ . In  $d$ -dimensional space the multilinear constraint is written

$$\text{rank}(M) \leq n + d + 1, \quad (2)$$

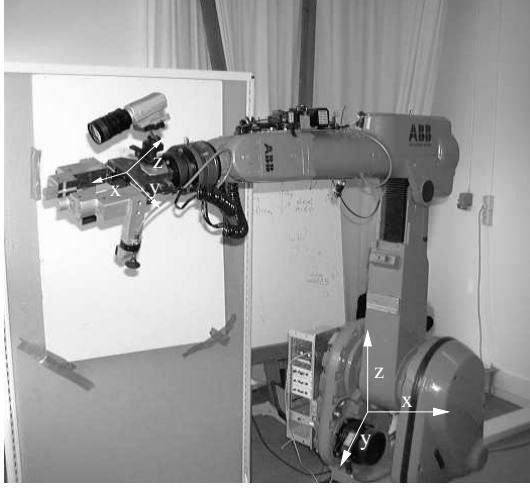
$$M = \begin{bmatrix} P(t_0) & u(t_0) & \mathbf{0} & \dots & \mathbf{0} \\ P(t_1) & \mathbf{0} & u(t_1) & \dots & \mathbf{0} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ P(t_n) & \mathbf{0} & \mathbf{0} & \dots & u(t_n) \end{bmatrix}. \quad (3)$$

Considering this in space gives

$$\text{rank} \begin{bmatrix} P(t_0) & u(t_0) & \mathbf{0} \\ P(t_1) & \mathbf{0} & u(t_1) \end{bmatrix} \leq 5. \quad (4)$$

This implies

$$\det \begin{bmatrix} P(t_0) & u(t_0) & \mathbf{0} \\ P(t_1) & \mathbf{0} & u(t_1) \end{bmatrix} = 0. \quad (5)$$



**Fig. 1.** Robot with coordinate system

### 3. CALIBRATION USING MULTILINEAR CONSTRAINTS

Using  $K$  as the internal calibration of the camera ( $3 \times 3$ ),  $R_{HE}$  and  $T_{HE}$  as the rotation respectively the translation of the camera in the robot hand coordinate system. The hand has rotation  $R_H$  and translation  $T_H$  in the world coordinate system and the camera motion can be expressed as

$$P(t) = KR_{HE} [R_H(t) \quad -T_{HE} - R_H(t)T_H(t)]. \quad (6)$$

With  $A = KR_{HE}$  the above is rewritten

$$P(t) = A [R_H(t) \quad -T_{HE} - R_H(t)T_H(t)]. \quad (7)$$

It is important to note that (5) gives a constraint for each pair of images of the same point and that constraints coming from different points may be used together without increasing overall dimensionality.

#### 3.1. Solving for Camera Rotation and Internal Calibration Using Translations

With pure translations we can with a suitable choice of robot coordinate system assume that  $R_H(t) = I$  and (7) is simplified to

$$P(t) = A [I \quad -T_{HE} - T_H(t)]. \quad (8)$$

Inserting this into (5) gives

$$\det \begin{bmatrix} A & A(-T_{HE} - T_H(t_i)) & u(t_i) & \mathbf{0} \\ A & A(-T_{HE} - T_H(t_j)) & \mathbf{0} & u(t_j) \end{bmatrix} = 0 \quad (9)$$

which is equivalent to

$$\det \begin{bmatrix} A & A(-T_{HE} - T_H(t_i)) & u(t_i) & \mathbf{0} \\ \mathbf{0} & A(-T_H(t_j) + T_H(t_i)) & -u(t_i) & u(t_j) \end{bmatrix} = 0. \quad (10)$$

Simplifying with  $\Delta T_H = T_H(t_j) - T_H(t_i)$ , the above is rewritten

$$\det A \det [-A\Delta T_H \quad -u(t_i) \quad u(t_j)] = 0 \quad (11)$$

$\Leftrightarrow$

$$-\det A \det [-A\Delta T_H \quad u(t_i) \quad u(t_j)] = 0 \quad (12)$$

$\Leftrightarrow$

$$\det [A\Delta T_H \quad u(t_i) \quad u(t_j)] = 0 \quad (13)$$

since  $\det(A) \neq 0$ . Decomposing  $A$  into elements, this is written

$$\det \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \Delta T_H \quad u(t_i) \quad u(t_j) = 0 \quad (14)$$

$\Leftrightarrow$

$$\sum_{k=1, l=1}^{3,3} a_{kl} \det [I(i, j)T_H \quad u(t_i) \quad u(t_j)] = 0 \quad (15)$$

where  $I(i, j)$  is a  $3 \times 3$  matrix with a 1 on place  $(i, j)$  and zeroes elsewhere. Let  $B(t_i, t_j)$  and  $\tilde{A}$  be column vectors with elements defined by

$$B_{3(k-1)+l}(t_i, t_j) = \det [I(k, l)T_H \quad u(t_i) \quad u(t_j)], \quad (16)$$

$$\tilde{A}_{3(k-1)+l} = a_{kl}. \quad (17)$$

Equation (15) can now be rewritten

$$B(t_i, t_j)^T \tilde{A} = 0. \quad (18)$$

This implies that

$$\tilde{A}^T (B(t_i, t_j)^T B(t_i, t_j)) \tilde{A} = 0, \quad (19)$$

which has to be fulfilled for all pairs of measurements  $(t_i, t_j)$ , that is

$$\sum_{i,j} \tilde{A}^T (B(t_i, t_j)^T B(t_i, t_j)) \tilde{A} = 0 \quad (20)$$

$\Leftrightarrow$

$$\tilde{A}^T \left( \sum_{i,j} B(t_i, t_j)^T B(t_i, t_j) \right) \tilde{A} = 0 \quad (21)$$

$\Leftrightarrow$

$$\begin{cases} \tilde{A}^T Q \tilde{A} = 0 \\ Q = \sum_{i,j} B(t_i, t_j)^T B(t_i, t_j) \end{cases} \quad (22)$$

As  $A$  has an arbitrary scale and  $\|A\|_F \neq 0$  it can be assumed that  $\|A\|_F = 1$ , where  $\|A\|_F = \|\tilde{A}\|_2 = \sqrt{\sum_{k,l} A_{kl}^2}$ . The problem is now overdetermined. A solution to this minimization problem is given by SVD[13] as  $Q$  is a symmetric matrix. The factorisation of  $A$  into an upper triangular matrix  $K$  and a rotation matrix  $R_{HE}$  is in principle a QR-factorisation [13].

### 3.2. Computing the Translation of a Camera Known in Both Orientation and Internal Parameters

From the previous step, the matrix  $A$  is known. By a change of the image coordinate system, it can without loss of generality be assumed that  $A = I$ . The camera matrix  $P(t)$  ( $3 \times 4$ ) can then be written

$$P(t) = [R_H(t) \quad -T_{HE} - R_H(t)T_H(t)] \quad (23)$$

and the constraint (5) is simplified to

$$\det \begin{bmatrix} R_H(t_i) & -T_{HE} - R_H(t_i)T_H(t_i) & u(t_i) & \mathbf{0} \\ R_H(t_j) & -T_{HE} - R_H(t_j)T_H(t_j) & \mathbf{0} & u(t_j) \end{bmatrix} = 0 \quad (24)$$

$$\Leftrightarrow C(t_i, t_j)^T T_{HE} + D(t_i, t_j) = 0 \quad (25)$$

where

$$D(t_i, t_j) = \det \begin{bmatrix} R_H(t_i) & R_H(t_i)T_H(t_i) & u(t_i) & \mathbf{0} \\ R_H(t_j) & R_H(t_j)T_H(t_j) & \mathbf{0} & u(t_j) \end{bmatrix} \quad (26)$$

is a scalar and  $C(t_i, t_j)$  is a column matrix with elements

$$c_k(t_i, t_j) = \det \begin{bmatrix} R_H(t_i) & I(k) & u(t_i) & \mathbf{0} \\ R_H(t_j) & I(k) & \mathbf{0} & u(t_j) \end{bmatrix} \quad (27)$$

where  $I(i)$  is column  $i$  of the  $3 \times 3$  identity matrix. Equation (25) implies that

$$(C(t_i, t_j)^T T_{HE} + D(t_i, t_j))^2 = 0 \quad (28)$$

$$\Leftrightarrow D^2(t_i, t_j) + 2D(t_i, t_j)C(t_i, t_j)^T T_{HE} + T_{HE}^T C(t_i, t_j)C(t_i, t_j)^T T_{HE} = 0 \quad (29)$$

which has to be fulfilled for a pairs of measurements

$$\sum_{i,j} (D^2(t_i, t_j) + 2D(t_i, t_j)C(t_i, t_j)^T T_{HE} + T_{HE}^T C(t_i, t_j)C(t_i, t_j)^T T_{HE}) = 0 \quad (30)$$

$$\Leftrightarrow \left( \sum_{i,j} D^2(t_i, t_j) \right) + 2 \left( \sum_{i,j} D(t_i, t_j)C(t_i, t_j)^T T_{HE} \right) + T_{HE}^T \left( \sum_{i,j} C(t_i, t_j)C(t_i, t_j)^T \right) T_{HE} = 0 \quad (31)$$

which is a second order polynomial for which the minimizing  $T_{HE}$  is the least squares solution to (25) for all observed image pairs.

### 3.3. Reprojection Based Method

In the situation of only one tracked point the setup is very similar to a special case of [5]. Using stage one of the method proposed here it is possible to compute  $A$  and from

this value to compute a point in space. Assuming a value for  $T_{HE}$  it is then possible to use the known motion of the robot hand to compute an image of the point that can be compared with the actual image of the point. The computed image should be equal to the real image point and the reprojection error is defined as the sum of these errors. Given this error function it is possible to find  $T_{HE}$  by conventional optimization. This alternative approach can be used to compute  $T_{HE}$  and will be used for verification as the laboratory experiments were made using only one point.

## 4. TESTS, VALIDATION AND SPEED

### 4.1. Tests in the Robot Lab

A modified ABB IRB2003 robot which is capable of moving in all 6 degrees of freedom was used to verify the methods on real data. The camera was mounted using a ball head camera holder on the gripper (or hand) of the robot so that the orientation of the camera could be changed with respect to the orientation of the robot hand coordinate system, cf. Figure 1. It is difficult to verify measurements made on real data. Visual inspection of the Robot tells us that the answer we get is realistic; it also gives a small reprojection error if used to solve for a point. All tests in the Robot lab have been made by tracking a single point as this was easier to handle.

Using the ‘‘Camera Calibration Toolbox for Matlab’’ by Bouguet implementing the calibration method of [14] it is possible to get an estimate for the camera parameters using a calibration plane

$$\begin{bmatrix} f_1 \\ f_2 \end{bmatrix} = \begin{bmatrix} 1001.3 \pm 5.8 \\ 1041.3 \pm 6.0 \end{bmatrix}, \quad (32)$$

$$\begin{bmatrix} x_0 \\ y_0 \end{bmatrix} = \begin{bmatrix} 317.8 \pm 7.3 \\ 240.5 \pm 6.6 \end{bmatrix}. \quad (33)$$

Tracking one point in 37 observations and using the method proposed here gives

$$K = \begin{bmatrix} 1000.6 & -0.1 & 326.0 \\ 0 & 1036.6 & 232.1 \\ 0 & 0 & 1 \end{bmatrix}. \quad (34)$$

It is hard to say if the estimated values for  $T_{HE}$  are correct or not, but a visual inspection of the camera mount tells us that the values are realistic and on a correct scale. Using a sequence with 72 camera positions and images a value for  $T_{HE}$  was calculated.

An attempt at validation of the method was done by selecting 30 random observations and computing  $T_{HE}$ , this was repeated 100 times. For  $T_{HE}$  we get

$$\text{mean}(T_{HE}) = \begin{bmatrix} -103.8 \\ -117.3 \\ -221.9 \end{bmatrix}, \quad (35)$$

$$\sigma(T_{HE}) = \begin{bmatrix} 4.6 \\ 6.8 \\ 4.1 \end{bmatrix}. \quad (36)$$

using the reprojection method  $T_{HE}$  was calculated to

$$T_{HE} = \begin{bmatrix} -101.6 \\ -106.6 \\ -221.2 \end{bmatrix}. \quad (37)$$

Matlab code for this implementation can be found on the authors home-page [15].

## 4.2. Speed

These costs are computed assuming, that we have a set of tracked points and corresponding robot movements. Tracked points can be obtained by using the KLT-tracker [16]. Estimates for flops are based on the numbers from Matlab. The cost of the first step of finding the camera is 330 flops per point pair and 15795 flops for SVD and RQ factoring. For the second stage there is a need of solving an over determined system in the overhead and the total load for 72 observations of one point is 1.4Mflops corresponding to 555flops per observation. For both steps together the load for under 100 observations is below 2mega-flops. Modern CPUs measure performance in gigaflops per second.

## 5. REFERENCES

- [1] R. K. Lenz and R. Y. Tsai, "A new technique for fully autonomous and efficient 3d robotics hand/eye calibration," *IEEE Transactions on Robotics and Automation*, vol. 5, no. 3, pp. 345–358, 1989.
- [2] R. K. Lenz and R. Y. Tsai, "Calibrating a cartesian robot with eye-on-hand configuration independent of eye-to-hand relationship," *IEEE Transactions on Robotics and Automation*, vol. 11, no. 9, pp. 916–928, 1989.
- [3] N. Andreff, R. Horaud, and B. Espiau, "On-line hand-eye calibration," in *Proceedings. Second International Conference on 3-D Digital Imaging and Modeling*, 1999, pp. 430–436.
- [4] F. Dornaika and R. Horaud, "Simultaneous robot-world and hand-eye calibration," *IEEE Transactions on Robotics and Automation*, vol. 14, no. 4, pp. 617–622, 1998.
- [5] Guo-Qing Wei, K. Arbter, and G. Hirzinger, "Active self-calibration of robotic eyes and hand-eye relationships with model identification," *IEEE Transactions on Robotics and Automation*, vol. 14, no. 1, pp. 158–166, 1998.
- [6] J. Dias, A. de Almeida, H. Araujo, and J. Batista, "Improving camera calibration by using multiple frames in hand-eye robotic systems," in *IEEE/RSJ International Workshop on Intelligent Robots and Systems*, 1991, vol. 1, pp. 285–291.
- [7] Hanqi Zhuang and Zhihua Qu, "A new identification jacobian for robotic hand/eye calibration," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 24, no. 8, pp. 1284–1287, 1994.
- [8] H. Malm and A. Heyden, "Hand-eye calibration from image derivatives," in *Proc. 6th European Conf. on Computer Vision, Dublin, Ireland*, June 2000, vol. 2, pp. 493–507.
- [9] R. I. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, Cambridge University Press, 2000.
- [10] A. Heyden and K. Åström, "A canonical framework for sequences of images," in *Proc. IEEE workshop on Representation of Visual Scenes, MIT, Cambridge, MA, USA*, 1995, pp. 45–52.
- [11] O. Faugeras and B. Mourrain, "On the geometry and algebra of the point and line correspondences between  $n$  images," in *Proc. 5th Int. Conf. on Computer Vision, MIT, Boston, MA*, 1995, pp. 951–956.
- [12] B. Triggs, "Matching constraints and the joint image," in *Proc. 5th Int. Conf. on Computer Vision, MIT, Boston, MA*, 1995, pp. 338–343.
- [13] G. H. Golub and C. F. Van Loan, *Matrix Computations*, Johns Hopkins University Press, 1996.
- [14] Z. Zhang, "Flexible camera calibration by viewing a plane from unknown orientations," in *Proc. 7th Int. Conf. on Computer Vision, Kerkyra, Greece*, 1999.
- [15] H. Stewenius, "Homepage with code for hand-eye calibration using multilinear constraints," <http://www.maths.lth.se/~stewe/>.
- [16] J. Shi and C. Tomasi, "Good features to track," in *Proc. Conf. Computer Vision and Pattern Recognition*, 1994.
- [17] H. Stewenius, "Studies in mobile camera calibration," Licentiate thesis, Centre for Mathematical Sciences, Lund Institute of Technology, Sweden, 2003.