2009 IEEE International Conference on Robotics and Automation
Kobe International Conference Center
Kobe, Japan, May 12-17, 2009

Automatic High-Precision Self-Calibration of Camera-Robot Systems

Andreas Jordt, Nils T Siebel and Gerald Sommer

Abstract— **In this article a new method is presented to obtain a full and precise calibration of camera-robot systems with eye-in-hand cameras. It achieves a simultaneous and numerically stable calibration of intrinsic and extrinsic camera parameters by analysing the image coordinates of a single point marker placed in the environment of the robot. The method works by first determining a rough initial estimate of the camera pose in the tool coordinate frame. This estimate is then used to generate a set of uniformly distributed calibration poses from which the object is visible. The measurements obtained in these poses are then used to obtain the exact parameters with CMA-ES (*Covariance Matrix Adaptation Evolution Strategy*), a derandomised variant of an evolution strategy optimiser. Minimal claims on the surrounding area and flexible handling of environmental and kinematical limitations make this method applicable to a range of robot setups and camera models. The algorithm runs autonomously without supervision and does not need manual adjustments. Our problem formulation is directly in the 3D space which helps in minimising the resulting calibration errors in the robot's task space. Both simulations and experimental results with a real robot show a very good convergence and high repeatability of calibration results without requiring user-supplied initial estimates of the calibration parameters.**

I. Introduction

A large part of current robotics research and development is on aspects that make robot systems more autonomous and versatile. Eye-in-hand systems, i.e. robots with a camera mounted to their end-effector, are often used to guide the end effector (e.g. a gripper) to ensure that grippers properly engage the intended targets. In order for such a system to work efficiently and with a high precision the system needs to know the exact pose of the camera in the robot tool frame as well as the camera's internal calibration parameters like the focal length and lense distortion.

Formulating these models and determining their parameters can be tedious, even though special calibration systems have been developed to make this easier. Re-calibration is one of the most important maintenance tasks for industrial robot systems. Parameters can change over time, e.g. if the camera is removed and re-assembled. The quality of this calibration process is highly relevant since inaccuracies limit the precision of the whole eye-in-hand system. Most of the research on autonomous hand-eye calibration has been done using calibration objects like checker boards, which provide a lot of information in one camera image. The dependence on a special calibration object, however, makes those algorithms less suitable for industrial purposes.

The algorithm presented in this article is able to automatically determine all intrinsic parameters, including lens distortion characteristics, as well as all extrinsic parameters, describing the relative pose of the camera with respect to the tool coordinate system. No special calibration pattern or high-precision calibration object is required; the method requires only one simple detectable marking in the robot environment. For data acquisition, the algorithm calculates a set of calibration movements. Based on the position of the marker in the images an optimisation problem is formulated and solved using the evolutionary optimiser CMA-ES (*Covariance Matrix Adaptation Evolution Strategy*) [1].

The remainder of the article is organised as follows. Section II describes related work. Section III gives a brief overview over our method, with details following in Sections IV (initial steps) and V (main calibration). The test setup and experimental results are located in Section VI, followed by conclusions in Section VII.

II. Related Work

Hand-eye calibration has been subject to research for about 30 years. The algorithms can be divided into 3 classes, depending on the type of calibration object used.

A. Methods using Calibration Boards

The most common calibration object is the calibration board. It allows to estimate the complete transformation between two camera poses by analysing the images acquired in these poses. Checker board patterns were used as calibration objects for hand eye calibration algorithms introduced 1988 by Tsai and Lenz [2], [3] and by Shiu and Ahmad [4]. Both calibration methods focus on calculating the transformation between wrist and camera without performing a camera calibration or robot calibration. They use local optimisation algorithms which usually necessitate good initial estimates. The method proposed by Shiu and Ahmad is based on the homogeneous [5] matrix equation $AX = XB$, where for each measurement A is the transition of the robot arm and B the motion captured by the camera [4]. X is the transformation from robot arm to the camera's optical centre which is to be determined. The algorithm by Weng *et al.* combines intrinsic and extrinsic calibration [6]. It works by first seeking the extrinsic parameter with minimum error for a standard camera model and then using these parameters as an initial estimate for refining intrinsic and extrinsic parameters. More recently Strobl and Hirzinger introduced a more effective way of solving $AX = XB$ based on a physical metric and a self-parameterising stochastical environment model, again in a calibration algorithm which needs a calibration board [7].

The authors are with the Cognitive Systems Group, Institute of Computer Science, Christian-Albrechts-University of Kiel, Germany. E-Mail: {anj,nts,gs}@ks.informatik.uni-kiel.de

B. Methods using Single Point Markers

The second class of algorithms uses detectable single points with unknown relative positions as calibration objects. These methods are more versatile because the calibration pattern requires only little space and no high precision in manufacturing it. Unlike calibration boards, single markers do not deliver enough information in one image to calculate the corresponding camera transformation (the matrix B in Section II-A above). This leads to a new class of non-convex optimization problems that require global optimisation. Horaud and Dornaika presented a method introducing a new problem formulation as $MY = M'YB$ which regarded intrinsic and extrinsic parameters without the need to render the projection explictly during the optimisation process [8]. Wei and Arbter introduced an algorithm using the Gauss Markov Theorem to build and optimise environment models yielding very accurate results [9].

C. Methods using Tracked Local Image Features

The third class of calibration algorithms is characterised by the use of no specific calibration pattern at all. Instead, these methods extract from each camera image salient points, e.g. by their local properties, and track them from image to image. Andreff *et al.* developed an algorithm using correspondences of such points within the images to calculate the performed transformation [10]. Their algorithm is, however, strongly influenced by the inaccuracy of these point measurements and/or correspondence errors.

III. Overview of our Method

A. Main Features

One of the most important aspects in wrist mounted sensor calibration is the process of solving the equations derived from the acquired data. Optimisation in the high dimensional space of calibration parameters is done either by acquiring poses especially designed to keep the resulting objective function smooth or by generating good initial estimates. In these special cases the convergence of local optimisation methods is more easily ensured. A number of mathematical tools have been introduced to determine and ensure convergence and to ensure constraints, like having a positive focal length. The new technique introduced in this paper offers more versatility and freedom in two aspects:

- There are no limitations to the poses or paths taken by the robot to keep the calibration object in the camera's field of view. Existing algorithms use iterative movements to approach a desired pose to keep the calibration pattern visible. Our algorithm uses 6 very small initial robot movements which provides sufficient data to calculate an orientation, enabling us to focus the object for every reachable robot position. An arbitrary number of poses with a visible object can thereby easily be generated. In our tests the object was visible from all generated poses, albeit the algorithm does not require this.
- There are no limitations on the environment and camera/robot model. Several models for geometry and lens distortion have been tested; these can easily be exchanged. Thanks to the properties of the CMA-ES optimiser every model yields a unique and reproducible set of parameters. The modular concept is very open to adaptations since every part of the evaluation process can be easily exchanged.

B. The Algorithm

An important difference of our algorithm is its formulation of the environment model, which minimises the actual error in 3D space. The negative of this error measure is used in the calculation of the fitness function which is maximised in the evolutionary optimisation loop.

The algorithm can be separated into 3 sets of consecutive robot movements. Each set it followed by a calculation to determine how the following movements are to be executed depending on the measurements in the images taken at each pose. These are the 3 steps of the algorithm:

1) Three translations (20–50 mm) are done to estimate the orientation of the camera.
2) Three rotational movements (approx. 5 °) are executed to estimate the distance to the calibration object.
3) A set of calibration poses is generated and movements executed to provide the data used in the actual calibration process. This process yields the parameter vector containing the calibration results.

While the calculations in step 1 and 2 are based on simple equations, the data derived from the images of step 3 is used to formulate a more complex optimisation problem. Depending on the camera model the fitness function has a dimension up to 17, expressing the dependence of the actual geometrical error on these parameters. The CMA-ES optimisation algorithm we use (see Section V-D) is able to directly minimise this fitness function without the need to smooth it, invert it explicitly or simplify it in any way.

IV. Initial Estimates and Pose Generation

The initial translations and rotations are performed to retrieve basic information about the environment, minimising the inherent assumptions of the method. This data is used to generate a set of robot poses in which the calibration mark is visible from the gripper mounted camera. Rotations are needed to gather information about translational differences between gripper and optical centre of the camera, while the orientation estimation requires translational movements [2].

Unlike the method in [9] the initial environment model calculation is based on the observation of a single marker and allows to compute an a priori error maximum. Because of this, the resulting model can be trusted to yield a pattern-facing pose for every position, and does *not* need iterative refinement and intermediate steps approaching far off poses.

A. Preliminaries

Let $c \in \mathbb{R}^4$ be the homogeneous [5] position of the calibration pattern in the world coordinate system and let $I \subset \mathbb{N}$ be the indices to the set of poses. For each $i \in I$ let $G_i \in \mathbb{R}^{4\times4}$ be the corresponding gripper pose

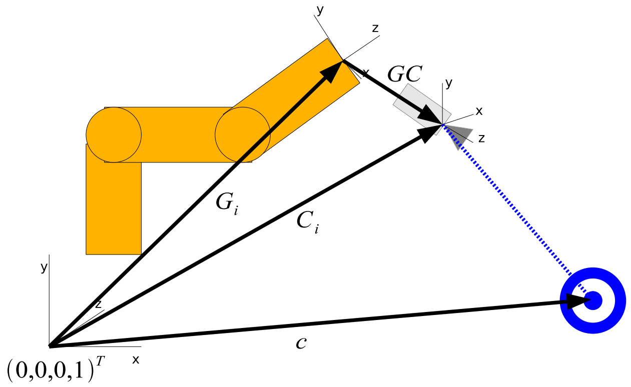

Fig. 1. The calibration pattern (bottom right) can be located by its position in the world coordinate system and as a point on the line of sight

as a homogeneous matrix in the world coordinate system[1]. Further, let $C_i \in \mathbb{R}^{4\times4}$ be the camera pose for a given index $i \in I$. The origin is the optical centre of the camera and the z-axis describes the line of sight. Since the camera is mounted rigidly to the gripper, the transformation $GC \in \mathbb{R}^{4\times4}$ from gripper to camera is constant:

$$GC = G_i^{-1} \cdot C_i \quad \forall i \in I. \tag{1}$$

In the following sections homogeneous transformation matrices are decomposed into their rotational and translational part. The indices r and t are used to denote this:

$$M = M_t \cdot M_r. \tag{2}$$

To simplify the equations homogeneous vector elements are referred to by indices. For $v \in \mathbb{R}^4, v = (x, y, z, s)^T$ we write $(v)_x := x$, $(v)_y := y$ and $(v)_z := z$.

B. Initial Translations

The first step of the algorithm is an estimation of the camera orientation in the (robot) world coordinate system, which is equivalent to estimating the transformation from the gripper to the camera, since the transformation from the robot base to the gripper coordinate system is known (up to the robot's precision).

Three equidistant translations are performed along the x-, y- and z-axis of the robot basis, resulting in camera poses C_1, C_2 and C_3. The image movements observed by the camera are defined as $t_1, t_2, t_3 \in \mathbb{R}$. Let $\delta_t := |t_1| = |t_2| = |t_3|$, the distance of each movement. Ignoring for a moment the projection to the image plane, the observed movement of the calibration object in camera coordinates can be used to describe the camera orientation:

$$t_1 = (C_0^{-1} \cdot c) - (C_1^{-1} \cdot c), \tag{3}$$

which leads to

$$t_1 = (GC^{-1} \cdot G_{r,0}^{-1} \cdot G_{t,0}^{-1} \cdot c) - (GC^{-1} \cdot G_{r,1}^{-1} \cdot G_{t,1}^{-1} \cdot c). \tag{4}$$

Since there is no rotation performed $(G_{r,0} = G_{r,1})$:

$$t_1 = GC_r^{-1} \cdot G_{r,0}^{-1} \cdot \begin{pmatrix} \delta_t \\ 0 \\ 0 \\ 0 \end{pmatrix}. \tag{5}$$

This derivation also applies to t_2 and t_3, which leads to the matrix equation:

$$GC_r = G_{r,0}^{-1} \cdot \begin{pmatrix} \frac{1}{\delta_t} & 0 & 0 & 0 \\ 0 & \frac{1}{\delta_t} & 0 & 0 \\ 0 & 0 & \frac{1}{\delta_t} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} & t_1^T & & \\ & t_2^T & & \\ & t_3^T & & \\ 0 & 0 & 0 & 1 \end{pmatrix}. \tag{6}$$

It is therefore possible to estimate the orientation of the camera by using these initial translations. Standard cameras only deliver 2D projections of the 3D movements t_1, t_2 and t_3. However, we can reconstruct the missing information due to the properties of the performed movements, assuming the projection is given by a parallel projection P_λ and a scale vector $\lambda \in \mathbb{R}^2$:

$$P_\lambda := \mathbb{R}^4 \to \mathbb{R}^2, \begin{pmatrix} x \\ y \\ z \\ s \end{pmatrix} \mapsto \begin{pmatrix} x \cdot \lambda_x \\ y \cdot \lambda_y \end{pmatrix}, \ \lambda = \begin{pmatrix} \lambda_x \\ \lambda_y \end{pmatrix} \tag{7}$$

The error caused by this assumption can be neglected, since the influence of the z value is in most parts substituted by the arbitrary scale vector. The projection model implies that $\exists \lambda_x, \lambda_y, (t_1)_z, (t_2)_z, (t_3)_z \in \mathbb{R}$:

$$t_1 = \begin{pmatrix} \frac{P(t_1)_x}{\lambda_x} \\ \frac{P(t_1)_y}{\lambda_y} \\ (t_1)_z \\ 1 \end{pmatrix} \wedge t_2 = \begin{pmatrix} \frac{P(t_2)_x}{\lambda_x} \\ \frac{P(t_2)_y}{\lambda_y} \\ (t_2)_z \\ 1 \end{pmatrix} \wedge t_3 = \begin{pmatrix} \frac{P(t_3)_x}{\lambda_x} \\ \frac{P(t_3)_y}{\lambda_y} \\ (t_3)_z \\ 1 \end{pmatrix}, \tag{8}$$

which, using (6), gives λ_x and λ_y:

$$\lambda_x = \left\| \begin{pmatrix} -\frac{P(t_1)_x}{\delta_t} \\ -\frac{P(t_2)_x}{\delta_t} \\ -\frac{P(t_3)_x}{\delta_t} \\ 0 \end{pmatrix} \right\|_2, \ \lambda_y = \left\| \begin{pmatrix} -\frac{P(t_1)_y}{\delta_t} \\ -\frac{P(t_2)_y}{\delta_t} \\ -\frac{P(t_3)_y}{\delta_t} \\ 0 \end{pmatrix} \right\|_2. \tag{9}$$

Since the product of the two rightmost matrices in (6) is orthonormal and rows 1, 2, 4 and the determinant ($= 1$) are known, the unknown 3^{rd} row is given by the cross product:

$$\begin{pmatrix} (t_1)_z \\ (t_2)_z \\ (t_3)_z \end{pmatrix} = \begin{pmatrix} -\frac{P(t_1)_x}{\delta_t \cdot \lambda_x} \\ -\frac{P(t_2)_x}{\delta_t \cdot \lambda_x} \\ -\frac{P(t_3)_x}{\delta_t \cdot \lambda_x} \end{pmatrix} \times \begin{pmatrix} -\frac{P(t_1)_y}{\delta_t \cdot \lambda_y} \\ -\frac{P(t_2)_y}{\delta_t \cdot \lambda_y} \\ -\frac{P(t_3)_y}{\delta_t \cdot \lambda_y} \end{pmatrix}. \tag{10}$$

This way the z-components lost during the projection can be recovered and GC_r can be calculated.

C. Initial Rotations

Now that the orientation of the camera $C_{0,r}$ is estimated it is possible to calculate the distance between the gripper and the calibration object by performing three rotations. Each is a rotation around a point on the camera's line of sight.

[1] The world coordinate system can be defined arbitrarily as long as it is fixed in relation to the robot coordinate system. In this article we equate the world coordinate system with the robot base coordinate system.

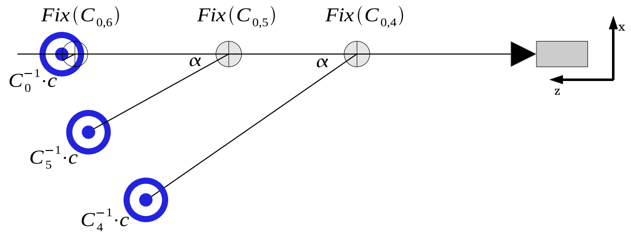

Fig. 2. The distance between the calibration object and the camera is calculated by rotations around a centre placed on the estimated line of sight.

Let $p_0, p_1, p_2 \in \mathbb{R}^4$ be these points and t_y the gripper's rotation axis. The corresponding gripper poses are defined as G_4,G_5 and G_6. Let $t \in \mathbb{R}^4$ be the 3D motion of the calibration pattern in the camera coordinate system while rotating around p_0 such that:

$$t = (C_0^{-1} \cdot c) - (C_4^{-1} \cdot c). \tag{11}$$

The motion t determines $C_{0,t}$ since there is one transformation GC_t to every motion t observed in the camera coordinate system. The movement performed by the gripper $G_{0,4}$ is known.

$$(C_4^{-1} \cdot c) = GC \cdot G_{0,4} \cdot GC^{-1} \cdot (C_0^{-1} \cdot c) \tag{12}$$

Define $R \in \mathbb{R}^{4 \times 4}$ as

$$R := GC_r \cdot G_{0,4} \cdot GC_r^{-1}. \tag{13}$$

Then solving the equation

$$x + t = GC_t \cdot R \cdot GC_t^{-1} \cdot x, x \in \mathbb{R}^4 \tag{14}$$

yields the missing elements of GC and the position of the calibration pattern c.

In contrast to the 3D motion the data provided by the camera is not sufficient to solve the equation (14). We know, however, that if $c = p_0$:

$$G_0^{-1} \cdot c = G_4{-}1 \cdot c \tag{15}$$

because of the definition of G_4. This implies:

$$C_0^{-1} \cdot c = C_4^{-1} \cdot c. \tag{16}$$

Any other p_0 leads to a visible motion in the camera image[2]. The direction and length of the observed motion depends on the distance between c and p_0 and the coordinate system as well as the projection.

Let $d \in \mathbb{R}$ be the initial distance estimation:

$$p_0 = C_0 \cdot \begin{pmatrix} 0 \\ 0 \\ d \\ 1 \end{pmatrix}. \tag{17}$$

Given an error $\varepsilon \in \mathbb{R}$ a rotation by $\alpha \in [0, \pi/2]$ around p_0 leads to an observable motion:

$$C_4^{-1} \cdot c = \begin{pmatrix} \sin(\alpha) \cdot \varepsilon \\ 0 \\ \cos(\alpha) \cdot \varepsilon + d \\ 1 \end{pmatrix}. \tag{18}$$

The next distance estimator d' is evaluated likewise using the same angle α for the rotation. The x-component of the position of the calibration object in the camera coordinate systems C_4, C_5 and the distance estimator error ε are linearly dependent. Disregarding the lens distortion, any pinhole or parallel projection output can be used to find a

$$p_2 = C_0 \cdot (0, 0, d'', 1)^T, d'' \in \mathbb{R} \tag{19}$$

such that $\varepsilon = 0$, $p_2 = c$ and

$$C_6^{-1} \cdot c = C_4^{-1} \cdot c, \tag{20}$$

which in our setup yields a very accurate initial estimate of GC, although a less precise estimate would suffice.

D. Pose Generation

Based on the acquired measurements a set of poses of arbitrary size can be generated. A set of many poses facing the calibration pattern can be distributed uniformly in the robot environment. (Unreachable poses will be reported by the inverse kinematics and discarded.) This way, the acquired data is based on the maximum entropy, providing a well defined optimum to the extrinsic parameters of the resulting equation system. In order to improve the calculation of the intrinsic camera parameters an adaptive random value is added to the orientation of each pose and thereby distributing the observation points of the marker uniformly in the image plane. This creates an optimal set of data to calculate intrinsic camera and lens distortion values in a stable manner. At least 16 poses are required to get a unique optimum but a minimum of 30 poses is recommended to guarantee a precise result.

V. Main Calibration and Optimisation

A. Camera Parameter Vector Θ

The acquired images are combined in a fitness function representing the actual geometrical environment. The parameter vector passed to the fitness function has 11 dimensions plus the dimensions needed to describe the lens distortion functions δ_u and δ_v. The function yields the actual geometrical error for every parameter set represented by the parameter vector Θ:

$$\Theta := (x, y, z, \gamma_x, \gamma_y, \gamma_z, \beta_x, \beta_y, c_x, c_y, c_z, \delta_u, \delta_v)^T \tag{21}$$

The parameters are:

- x, y, z to describe the translation represented by GC.
- γ_x, γ_y, γ_z to describe the rotation represented by GC.
- β_x and β_y, aperture ratio in x- and y-direction.
- c_x, c_y and c_z, the position of the calibration pattern.
- δ_u and δ_v, the lens distortion functions.

[2] This statement is true for every projection but the trivial one, which can be neglected.

The position of the calibration marker is not necessarily a part of the optimisation problem. For every parameter vector Θ', defined as Θ without c_x, c_y and c_z, a position can be calculated that yields the lowest possible error with respect to the given estimation Θ'. This calculation is done in $O(|I|^2)$ and gives a high computational cost to the otherwise very effective $O(|I|)$ fitness function that works without rendering the calibration marker position. This aspect increases the problem dimension by 3 but it preserves the versatility of the algorithm by keeping it scalable.

B. *Environment Model*

The error value given by the fitness function is the mean negative error calculated over all images. Let $i \in I$ be the image at hand. Evaluating Θ for image i is done by calculating the image coordinates according to Θ, where the calibration pattern centre is expected and comparing this to the observed image coordinates. The Gripper pose G_i is known for every image. GC^* and c^* are provided by the current guess Θ^*. In the camera coordinate system the calibration pattern is supposed to be at position p_i^*,

$$p_i^* = (C_i^*)^{-1} \cdot c^* = G_i^{-1} \cdot (GC^*)^{-1} \cdot c^*. \tag{22}$$

Hence image coordinates implied by Θ^* are given by

$$u_i^* = \frac{(p_i^*)_x}{(p_i^*)_z} + \delta_u \left(\frac{(p_i^*)_x}{(p_i^*)_z}, \frac{(p_i^*)_y}{(p_i^*)_z} \right) \tag{23}$$

and

$$v_i^* = \frac{(p_i^*)_y}{(p_i^*)_z} + \delta_v \left(\frac{(p_i^*)_x}{(p_i^*)_z}, \frac{(p_i^*)_y}{(p_i^*)_z} \right) \tag{24}$$

The error e_i for an image i is then given by

$$e_i(\Theta^*) = \sqrt{(u_i - u_i^*)^2 + (v_i - v_i^*)^2}, \tag{25}$$

where (u_i, v_i) are the image coordinates the pattern was actually observed at. The overall fitness of a parameter vector Θ is therefore given by the negative RMS error

$$F(\Theta) = -\sqrt{\sum_{i \in I} e_i(\Theta)^2}. \tag{26}$$

C. *Lens distortion*

The calibration algorithm does not depend on any particular lens distortion algorithm but is able to apply every distortion model given by $\delta_u : \mathbb{R}^2 \to \mathbb{R}$ and $\delta_v : \mathbb{R}^2 \to \mathbb{R}$.

A list of distortion models have been tested, including the one by Heikkilä and Silvén [11] and different types of polynomial distortion models. The distortion model described by Weng *et al.* [6] is the most versatile and gave very stable results. Therefore it was used in the tests shown in Section VI. It is defined by the distortion functions

$$\delta_u(u, v) = (g_1 + g_3)u^2 + g_4 uv + g_1 v^2 + k_1 u(u^2 + v^2) \tag{27}$$

and

$$\delta_v(u, v) = g_2 u^2 + g_3 uv + (g_2 + g_4)v^2 + k_1 v(u^2 + v^2) \tag{28}$$

where g_1, g_2, g_3, g_4 and k_1 are the distortion parameters.

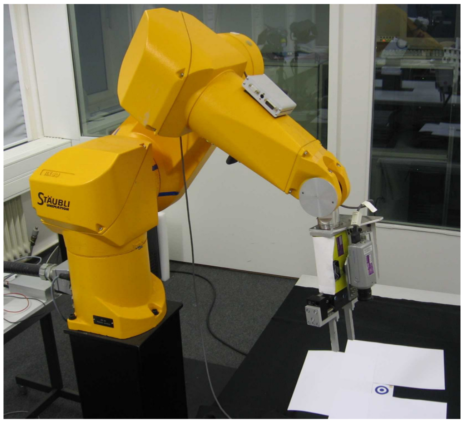

Fig. 3. Unimation Stäubli RX90 with camera and calibration marker

D. *The Use of CMA-ES and Initial Calibration Values*

The CMA-ES algorithm (*Covariance Matrix Adaptation-Evolution Strategy*) is an evolutionary algorithm for global non-linear optimisation [1]. As an evolutionary algorithm, its optimisation loop keeps populations of candidate solutions (individuals), selects the best (fittest) ones and mutates them to produce the next generation, thereby moving in the search space. As an evolution strategy (ES) algorithm, it adapts its own strategy (search) parameters to the structure of the search space. This algorithm uses a second order approach by estimating a covariance matrix of samples. This matrix is an approximation of the inverse Hessian matrix as used in traditional optimisation, however, CMA-ES does not require the knowledge (or even existence) of a derivative.

Using CMA-ES enables our algorithm to directly minimise the error in 3D space without the need to smooth the fitness function. We use CMA-ES with a population size of 30 individuals and a maximum number of 700 generations. The mean calculation time of the optimisation is less than 20 seconds. The initial values for all calibration values are set to 0. CMA-ES initialises the first generation around this value. Simple physical constraints are enforced during the optimisation, e.g. the focal length must be positive, $f > 0$.

VI. Experiments

A. *Test Setup*

In the test setup a Sony DFW-X710 camera was mounted rigidly to the wrist of a Unimation Stäubli RX90 robot arm, see Figure 3. A single calibration marker was placed on the ground in front of the robot. Hand-eye calibration test methods are difficult to validate since it is very difficult to obtain high-precision ground truth. Therefore, our algorithm was tested both in the real world and in a simulator. The simulator uses a full model of the environment in OpenGL and a simulated camera.

B. *Results from Simulations*

Tests in a simulated 3D environment were performed to evaluate the convergence properties of the method and its accuracy. The results achieved in the test runs showed a

TABLE I

MEAN RMS ERROR IN THE INTRINSIC AND EXTRINSIC CAMERA PARAMETERS

Parameter	x	y	z	yaw	pitch	roll	β_x	β_y	g_1	g_2	g_3	g_4	k_1
RMS	0.063	0.017	0.021	0.0014°	0.027°	0.025°	0.0054	0.0069	0.00009	0.00007	0.00008	0.00010	0.00011

very reliable convergence. All 4000 tests runs for this article converged to the same optimum, so that the same results are achieved every time the algorithm is run. The test results displayed in Table I have been generated using a simulated Stäubli RX90, a camera aperture angle of 45° and a camera resolution of 1024×1024 pixels.

Additionally, a test with artificial image noise has been performed in order to compare the algorithm introduced in this paper to other calibration methods. In [9] Wei *et al.* compare their calibration method to the classical Tsai and Lenz algorithm. Using the same setup and also image noise of $\sigma_x = \sigma_y = 0.5$ px, the mean overall translation errors are:

Our Method	Wei and Arbter	Tsai and Lenz
0.28 mm	0.36 mm	0.58 mm

It can be seen that in the simulation, where precision can be most easily compared, our algorithm performs significantly better than the standard methods.

C. Results from Tests in the Real World

The accuracy of the results with the Unimation Stäubli RX90 is better than we can measure. Therefore the evaluation of real world tests uses the standard deviation of the results from a number of test runs, i.e. we measure repeatability. The test consisted of several calibration tasks, each performed in the same environment but with different start poses and different relative positions (using random variables during pose generation). The standard deviations of the extrinsic parameters were:

σ_x	σ_y	σ_z	σ_{yaw}	σ_{pitch}	σ_{roll}
1.5 mm	1.08 mm	2.54 mm	0.08°	0.15°	0.13°

Real world test results are not very suitable for comparing them to algorithms not running on the same system since the results strongly depend on robot accuracy, camera resolution and aperture angle. Therefore no comparison is given here.

D. Calibration Time

Data acquisition takes a few minutes, depending on the robot and image processing system used. In our setup about 6 minutes are required to collect measurements from around 40–50 robot poses. The final calculation of the calibration parameters requires around 20 seconds.

VII. CONCLUSIONS

A new algorithm for full calibration of camera-robot systems with eye-in-hand camera has been presented. It has the following features:

- Simultaneous and numerically stable calibration of intrinsic and extrinsic camera parameters
- Runs without user-given initial estimates
- Very easy usage; there are no parameters to adjust
- Simple application; only a small and easy to produce calibration pattern is required
- Versatility, since the environment and lens distortion models can be easily exchanged
- Higher accuracy than existing and well-established algorithms.

The method uses a very effective way of acquiring information to build an initial environment model using 6 very small robot movements. This environment model provides the ability to focus the calibration object from every robot/camera position and thereby enables the system to automatically acquire many (numerically) relevant measurements. The CMA-ES optimisation algorithm provides a stable and reliable way to calculate a resulting camera parameter vector.

Experiments from simulations show a very good and fast convergence of the algorithm to the ground truth values in a range of setups, proving its versatility. Experiments with a real robot show a high repeatability of calibration results with different start poses.

REFERENCES

[1] N. Hansen and A. Ostermeier, "Completely derandomized self-adaptation in evolution strategies," *Evolutionary Computation*, vol. 9, no. 2, pp. 159–195, 2001.

[2] R. Y. Tsai and R. K. Lenz, "Real time versatile robotics hand/eye calibration using 3D machine vision," *IEEE Transactions on Robotics and Automation*, vol. 1, pp. 554–561, 1988.

[3] ——, "A new technique fo fully autonomous end efficient 3D robotic hand/eye calibration," *IEEE Transactions on Robotics and Automation*, vol. 5, no. 3, pp. 345–558, 1989.

[4] Y. C. Shiu and S. Ahmad, "Calibration of wrist-mounted robotic sensors by solving homogeneous transform equations of the form ax=xb," *IEEE Transactions on Robotics and Automation*, vol. 5, no. 1, pp. 16–29, 1989.

[5] M. W. Spong, S. Hutchinson, and M. Vidyasagar, *Robot Modeling and Control*. New York, Chichester: John Wiley & Sons, 2005.

[6] J. Weng, P. Cohen, and M. Herniou, "Camera calibration with distortion models and accuracy evaluation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 10, pp. 965–980, October 1992.

[7] K. H. Strobl and G. Hirzinger, "Optimal hand-eye calibration," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2006)*, Beijing, China, October 2006, pp. 4647–4653.

[8] R. P. Horaud and F. Dornaika, "Hand-eye calibration," *International Journal on Robotics Research*, vol. 14, no. 3, pp. 195–210, June 1995.

[9] G.-Q. 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, February 1998.

[10] N. Andreff, R. P. Horaud, and B. Espiau, "On-line hand-eye calibration," in *Proceedings of the Second International Conference on 3-D Digital Imaging and Modeling (3DIM'99), Ottawa, Canada*, October 1999, pp. 430–436.

[11] J. Heikkilä and O. Silvén, "Calibration procedure for short focal length off-the-shelf ccd cameras," in *Proceedings of the 13th International Conference on Pattern Recognition (CVPR'96), Wien, Austria*, 1996, pp. 166–170.