The 15th International Conference on Advanced Robotics
Tallinn University of Technology
Tallinn, Estonia, June 20-23, 2011

Robust robot-camera calibration

J. Ilonen and V.Kyrki

Abstract— **Calibrating the parameters of a vision system used in robotics is crucial for many tasks where the robot has to physically interact with the environment. This paper introduces a robust method for calibrating the relative poses between the base frame of the robot and one or more cameras. The method is based on tracking a marker attached to the end-effector of the robot and does not require manual bootstrapping. The method is robust to a large number of outliers (incorrectly detected marker positions) and provides an estimate of the uncertainty of the estimated parameters based on the variance of the observed errors, providing information on the accuracy of the estimate. Experimental results demonstrate that highly robust estimates can be obtained with relatively few measurements.**

I. Introduction

Calibrating the parameters of a vision system for robotics is crucial for many tasks where a robot has to physically interact with the environment. Especially in service robotics scenarios the relative pose between the vision system and a robot manipulator can change often and thus a calibration method should be both fast and automatic requiring no human intervention. Moreover, service robotics environments can exhibit much variation which can often result in some incorrect measurements during the calibration. To answer to these challenges, this paper introduces a robust method for calibrating the relative poses between the base frame of the robot and one or more cameras. The method is based on tracking a marker rigidly attached to the end-effector of the robot with known forward kinematics. The position of the marker relative to the end-effector is also estimated. No manual bootstrapping is required, the robot can be moved to random joint positions and if the marker is visible to the camera the data is stored. After acquiring the marker positions, the pose of the camera and the marker position are estimated. The estimation also provides information on the accuracy of the estimate by back-propagating the variance of the residual errors, which implicitly includes errors caused by flexibility of the robot arm. The estimation method is based on robust M-estimators and therefore robust to outliers, i.e., incorrectly detected marker positions do not affect the estimate.

The method is related to the camera calibration method by Zhang [17] (additional details in [16]), where a planar calibration target is viewed from several viewpoints and the method then solves the intrinsic camera parameters and relative poses of the calibration objects to the camera. Zhang's approach provides a non-robust maximum likelihood estimate for the parameters using the Levenberg-Marquardt algorithm. Zhang's method requires that the markers lie on a plane while in our case the requirement is that their relative poses are known from forward kinematics. Rotation matrix is parametrized using axis-angle presentation in both methods. In Zhang's method the main interest is in calibrating the intrinsic camera parameters and the pose of the calibration plane comes along for free. In our method the main interest is estimating the pose between camera and the robot, which holds the calibration target which is moved to different positions. In this article it is assumed that the intrinsic parameters of the camera are already known, but the method could be trivially extended to estimate also them on the cost of increasing the required number of data. Intrinsic camera parameters do not typicalle change often and therefore the steps have been kept separate.

Main contributions and design goals of the method are:

- Simple and accurate method for calibrating the relative robot-camera pose.
- Robust to noise in detected marker positions and to complete outliers.
- Provides information on accuracy of the estimate.
- Reasonably fast, both in the sense of not requiring much data and fast computation.

Section II covers shortly the related work, Section III includes details of the presented method based on non-linear parameter estimation using M-estimators, Section IV presents the experimental setup and results which show the the robustness to outliers and required number of measurements and finally Section V the conclusions.

II. Related work

The problem in our case is formulated similarly to extrinsic camera calibration using a known 3D object. Zhang's method [17] has been extended for 3D calibration objects for example in [10] and for moving 1D objects in multi-camera setups in [13]. There also numerous other calibration methods based on 3D calibration objects, for example [3], [4] where unit quaternions are used for handling rotations and the robustness to noise in marker positions are of special interest. However, the extrinsic camera calibration methods are not directly applicable to our case because if a separate calibration object is attached to the robot, there is still the problem of finding out its position relative to the robot, or the required shape of the calibration object (which in our is equal to movement of the robot's hand and the attached marker) would be difficult to realize.

The research leading to these results has received funding from the European Community's Seventh Framework Programme under grant agreement n^o 215821.

The authors are with Machine Vision and Pattern Recognition research group, Lappeenranta University of Technology, Skinnarilankatu 34, 53850 Lappeenranta, Finland. {ilonen,kyrki}@lut.fi

978-1-4577-1159-6/11/$26.00 ©2011 IEEE

In the field of robotics there are many studies on hand/eye calibration, where the location of the camera mounted on the end-effector has to be determined. For example in the method by Tsai and Lenz [12] few images of a planar calibration object are taken and the hand/eye calibration can be then performed. Later, even automatic calibration methods have been designed [9].

Many papers where the problem of calibrating the coordinate frames between the robot and the camera is of concern provide little detail on the calibration as the main problem presented the an article lies elsewhere. Nevertheless, the calibration is crucial for example in robots with an active head. In that situation, the calibration can be based on a checkerboard pattern which defines the world frame, i.e., the location of the base frame of the robot in the world frame is not considered explicitly as presented in [14]. The calibration is based on Zhang's method [17] and is mostly concentrating the kinematic calibration; the location of the calibration target in the robot frame is estimated as a separate step causing additional uncertainty.

Another approach proposed in [5] is to use a LED attached to the robot's end effector which is then moved in a pattern. Combined stereo and robot-camera calibration is performed at the same time and the method is based to that of Zhang [17]; the LED is moved in regular planar patterns instead of using a actual planar checkerboard. In comparison to this article, the method in [5] requires a human to select the pattern in which the LED is moved is manually so that the camera sees the whole pattern, and the position of the LED is measured by a human separately and not estimated jointly from the visual measurements.

The method presented in this work improves the existing approaches by directly estimating the pose of the camera relative to the robot base frame, not needing a separate calibration target defining the world frame causing additional uncertainty, and estimating the position of the marker simultaneously, making possible the back-propagation of the residuals to the uncertainty of the parameters.

III. Overview of the method

The robot-camera calibration is based on a marker attached rigidly to the end-effector of the robot. The position of the marker in the end-effector frame and the pose between the camera and the robot base frame are estimated during calibration. The overview of the estimated parameters is presented in Fig. 1. When more than one camera is calibrated at the same time they all have separate poses, but share the same marker position.

Intrinsic parameters of the cameras are assumed to be calibrated separately, because effective methods already exist [7], [17] and the number of measurements would increase needlessly as the intrinsic parameters (in fixed focus cameras) do not usually change. The intrinsic parameters have been estimated in this work with Matlab camera calibration toolbox [2]. The method can apply both radial and tangential distortion parameters of the camera.

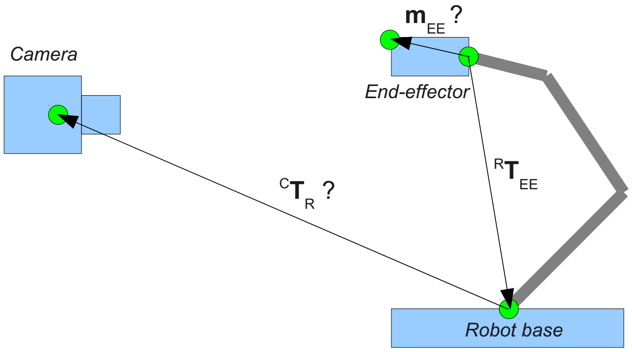


Fig. 1. Overview of the setup and what is being estimated.

In the following all necessary steps for calibration are described and in the end of this section the steps of the calibration procedure are enumerated. The partial derivatives needed for calculating Jacobians are included for the sake of completeness in Appendix.

A. Coordinate transformation from end-effector to camera frame

A joint rotation-translation matrix is composed as

$$\mathbf{T} = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ 0 & 1 \end{bmatrix} \tag{1}$$

where $\mathbf{R}$ is a 3×3 rotation matrix and translation vector $\mathbf{t} = [t_x, t_y, t_z]^T$.

It is assumed that the joint rotation-translation matrix between the end-effector and the robot base, $^{R}\mathbf{T}_{EE}$, is known from forward kinematics and can be computed for the current measurement from for example joint values.

In the camera frame the XYZ position of the marker $\mathbf{m}_{EE}$ is

$$\mathbf{m}_C = \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} =^C \mathbf{T}_R \, {}^{R}\mathbf{T}_{EE} \, \mathbf{m}_{EE}, \tag{2}$$

where $^{R}\mathbf{T}_{EE}$ is known from the forward kinematics and $\mathbf{m}_{EE}$ and $^{C}\mathbf{T}_{R}$ are the unknown parameters to be determined. The marker position $\mathbf{m}_{EE} = [m_x, m_y, m_z, 1]^T$ is the position of the marker in the EE frame in homogeneous coordinates. There are several possible parametrizations for rotation matrix, but the most natural for our case is the axis-angle presentation where vector $\boldsymbol{\theta} = [\theta_0, \theta_1, \theta_2]^T$ defines the axis of rotation and the length of vector $|\boldsymbol{\theta}|$ defines the rotation around the specified axis. The benefit of this formulation is that the partial derivatives, which are needed for estimation, have only one singular point (zero length vector) unlike Euler angles, and there are no extra constraints unlike with quaternions where there are four parameters presenting three degrees of freedom.

The rotation matrix from vector $\boldsymbol{\theta}$ is defined as [6]

$$\mathbf{R}(\boldsymbol{\theta}) = cos|\boldsymbol{\theta}|\mathbf{I} + \frac{sin|\boldsymbol{\theta}|}{|\boldsymbol{\theta}|}[\boldsymbol{\theta}]_x + \frac{1 - cos|\boldsymbol{\theta}|}{|\boldsymbol{\theta}|^2}\boldsymbol{\theta}\boldsymbol{\theta}^T, \tag{3}$$

where $\mathbf{I}$ is the identity matrix and $[\boldsymbol{\theta}]_x$ is the skew-symmetric matrix

$$[\boldsymbol{\theta}]_{\mathbf{x}} = \begin{bmatrix} 0 & -\theta_2 & \theta_1 \\ \theta_2 & 0 & -\theta_0 \\ -\theta_1 & \theta_0 & 0 \end{bmatrix}. \tag{4}$$

In case of one camera and one marker the parameters to be estimated are

$$\mathbf{X} = \{\theta_0, \theta_1, \theta_2, t_x, t_y, t_z, m_{EE_x}, m_{EE_y}, m_{EE_z}\}. \tag{5}$$

B. From camera frame to pixel position

To get from a 3D point in the camera frame to the actual camera pixel position a pinhole camera model with radial and tangential distortion is used [7].

The basic pinhole camera can be defined as

$$\begin{bmatrix} u \\ v \\ w \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} \tag{6}$$

where f_x, f_y is the focal length of the lens in x and y directions and c_x, c_y is the principal point (center of the image) and the actual pixel position is $(u/w, v/w)$. Distortion parameters are (k_1, k_2, k_3) for the radial distortion and (p_1, p_2) for the tangential distortion. The pixel position (u, v) can then be calculated as follows:

$$\begin{aligned} \begin{bmatrix} \dot{a} \\ \dot{b} \end{bmatrix} &= \begin{bmatrix} a \\ b \end{bmatrix} (1 + k_1 r_2 + k_2 r_2^2 + k_3 r_2^3) + \\ &\quad \begin{bmatrix} 2p_1 ab + p_2(r_2 + 2a^2) \\ p_1(r_2 + 2b^2) + p_2 ab \end{bmatrix} \\ u &= f_x \dot{a} + c_x \\ v &= f_y \dot{b} + c_y, \end{aligned} \tag{7}$$

where $a = x/z$, $b = y/z$ and $r_2 = a^2 + b^2$.

C. Weighted Gauss-Newton approximation and M-estimators

The problem is formulated as a bundle adjustment [11] problem using a non-quadratic M-estimator [15] which can explicitly handle outliers, unlike more classical maximum-likelihood or least-squares formulation. In this case when using a single camera and a single marker the problem is not sparse, but when several cameras (non-calibrated stereo system, for example) are estimated the problem becomes sparse because pixel position of the marker in a camera is not dependent of other cameras, they only share the same marker position, $\mathbf{m}_{EE}$. Parameters are estimated with iterative reweighted least squares (IRLS) method.

A single iteration of weighted Gauss-Newton approximation consists of solving $\boldsymbol{\Delta}$ in equation

$$(\mathbf{J}^T \mathbf{W} \mathbf{J}) \boldsymbol{\Delta} = -\mathbf{J}^T \mathbf{W} \boldsymbol{\epsilon} \tag{8}$$

where $\mathbf{J}$ is the Jacobian which is formed from partial derivatives, (18), $\mathbf{W}$ is the weight matrix and $\boldsymbol{\epsilon}$ are the residuals (prediction error), $\boldsymbol{\epsilon} = \overline{\mathbf{z}} - \mathbf{z}(\mathbf{X})$, where $\overline{\mathbf{z}}$ are the measured and $\mathbf{z}(\mathbf{X})$ the predicted values. For least-squares estimate the weight matrix is the identity matrix. The estimated parameters are then adjusted,

$$\mathbf{X}_{i+1} = \mathbf{X}_i + \alpha \Delta, \tag{9}$$

where $\alpha =]0, 2]$ is selected so that the weighted residual is minimized [11],

$$\underset{\alpha \in]0,2]}{\arg\min}\, w \left((\overline{\mathbf{z}} - \mathbf{z}(\mathbf{X} + \alpha \boldsymbol{\Delta}))^2 \right). \tag{10}$$

where $w(\cdot)$ is a weight-function, which is $w(\cdot) = 1$ in the case of L_2 norm (least-squares). Using the assumption $\alpha = 1$ often leads to non-optimal improvement (i.e., more iterations are needed) and computing just the residuals instead of the full Jacobian and solving new $\boldsymbol{\Delta}$ is considerably faster. Iterations are continued until no improvement is found or a limit on number of iterations is reached.

Exact detection of the marker position may fail for various reasons. In our case the marker is a LED and a common cause for outliers is detecting its reflection instead of the marker itself. When used with Gauss-Newton approximation the M-estimator defines the diagonal elements in the weight matrix $\mathbf{W}$ [15]. Two M-estimators have been used to increase robustness to outliers compared to non-robust L_2 norm. With L_1 norm $w(x) = \frac{1}{|x|}$ the weight decreases with increasing residual, however the weight goes to infinity when x approaches zero. Therefore, $L_1 - L_2$ M-estimator is used, because like L_1 the influence of large errors is reduced and like L_2 the function is defined everywhere. Initial estimation is started with $L_1 - L_2$ M-estimator, (11), and after convergence more outlier-resistant Welsch M-estimator, (12), is used. The weight function of $L_1 - L_2$ M-estimator is

$$w(x) = \frac{1}{\sqrt{1 + x^2/2}}. \tag{11}$$

and the weight function of Welsch M-estimator is [15],

$$w(x) = e^{-\left(\frac{x}{c}\right)^2}. \tag{12}$$

where with suitably chosen value of c the weight for large outliers approaches zero.

Few examples of weight functions of $L_1 - L_2$ and Welsch M-estimators are presented inf Fig. 2. The value of $c = 2.9848$ for Welsch-function has been selected as one of the presented graphs because then it reaches 95% asymptotic efficiency with standard normal distribution [15].

D. Backward propagation of covariance

In general given a non-linear function $f : \mathbb{R}^M \to \mathbb{R}^N$ and $\mathbf{v}$ a random vector in $\mathbb{R}^M$, the approximation of mean and covariance of $f(\mathbf{v})$ can be computed in the vicinity of the mean $\overline{\mathbf{v}}$ of the distribution. The approximation of f is $f(\mathbf{v}) \approx f(\overline{\mathbf{v}}) + \mathbf{J}_f(\mathbf{v} - \overline{\mathbf{v}})$, where $\mathbf{J}_f$ is the Jacobian $\frac{\partial f}{\partial \mathbf{v}}$ evaluated at $\overline{\mathbf{v}}$. The first-order approximation of random variable $f(\mathbf{v})$ has mean $f(\overline{\mathbf{v}})$ and covariance $\boldsymbol{\Sigma}_f = \mathbf{J}_f \boldsymbol{\Sigma} \mathbf{J}_f^T$. In our case we can calculate or have a reasonable assumption

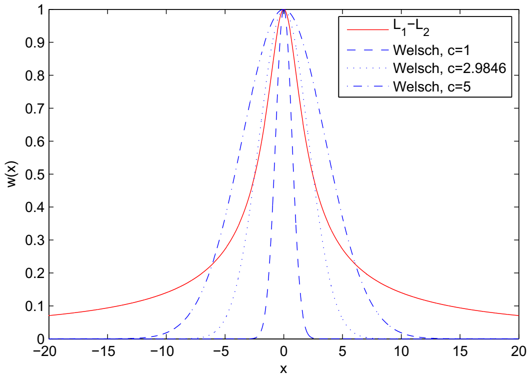


Fig. 2. Weight functions of $L_1 - L_2$ and Welsch M-estimators.

of covariance of $f(\mathbf{v})$ (the pixel positions given estimated parameters) and would like to propagate the covariance backwards. One option would be to create inverse mapping function f^{-1} to map pixel positions to parameters and their partial derivatives $\mathbf{J}_{f^{-1}}$, but fortunately that is not needed, because the inverse covariance propagation can be calculated as [1], [6]

$$\boldsymbol{\Sigma}_{f^{-1}} = \left(\mathbf{J}_f^T \boldsymbol{\Sigma}_f^{-1} \mathbf{J}_f \right)^{-1}. \tag{13}$$

In our case only inliers, defined as measurements with M-estimator weight > 0.1 are used, which corresponds with Welsch with $c = 5.0$ to < 7.59 pixel error. The variance σ^2 is assumed to be the same for all samples and is calculated from the residual of inliers. Jacobian $\mathbf{J}_{in}$ includes only the inlier measurements. The equation can be simplified to form

$$\boldsymbol{\Sigma} = \sigma^2 (\mathbf{J}_{in}^T \mathbf{J}_{in})^{-1}. \tag{14}$$

E. The complete calibration procedure

Following steps are included in the complete calibration procedure:

1) Collect data; move the robot to random poses, store the marker position $\overline{z_i}$ and ${}^R\mathbf{T}_{EE}^i$.
2) Start the parameter estimation, initialize the parameters $\mathbf{X}$.
3) For each measurement i, calculate the predicted measurements $z_i = z(\mathbf{X}, {}^R\mathbf{T}_{EE}^i)$, (2) and (7), residuals $\epsilon_i = \overline{z_i} - z_i$ and partial derivatives $\mathbf{J}_i$, (18) and (19).
4) Calculate M-estimator weights $\mathbf{W}$ based on the residuals, (11) or (12).
5) Do one iteration of Gauss-Newton estimation, (8) and (10).
6) If an improved solution was found, apply (9) and go back to step 2, otherwise stop.
7) Parameters estimated, check back-propagated variances if needed, (14).

IV. Experiments

A. Setup

The experiments have been performed using Mitsubishi RV3SB industrial robot with an attached Schunk PG 70 gripper and Bumblebee 2 stereo camera with 640×480 resolution. The marker was a blinking red LED gripped in an arbitrary pose by the gripper. The industrial robot is very rigid with a documented pose repeatibility of $\pm 0.02mm$, which is different to accuracy, but still it was assumed that the major error cause was the detection of the marker, not the forward kinematics of the robot.

The stereo camera was calibrated using camera calibration toolbox for Matlab [2]. The camera-robot calibration method presented here requires that the intrinsic parameters are known and the method can apply the information of respective poses of cameras in calibrated stereo by modifying (2) to include the joint rotation-translation between left and right cameras, ${}^{C_r}\mathbf{T}_{C_l}$, assuming the pose between robot and the left camera is being calibrated. In that case (2) becomes two separate equations for left and right cameras, but the number of estimated parameters (6 for ${}^{C_l}\mathbf{T}_R$ and 3 for $\mathbf{m}_{EE}$) stays the same as in one camera case,

$$\begin{aligned} \mathbf{m}_{C_l} &= {}^{C_l}\mathbf{T}_R \, {}^R\mathbf{T}_{EE} \, \mathbf{m}_{EE} \\ \mathbf{m}_{C_r} &= {}^{C_r}\mathbf{T}_{C_l} \, {}^{C_l}\mathbf{T}_R \, {}^R\mathbf{T}_{EE} \, \mathbf{m}_{EE}. \end{aligned} \tag{15}$$

Some tests were performed so that the two cameras of the calibrated stereo were treated as separate and both ${}^{C_l}\mathbf{T}_R$ and ${}^{C_r}\mathbf{T}_R$ were estimated, which means that there was 15 parameters to estimate in total as $\mathbf{m}_{EE}$ is still shared,

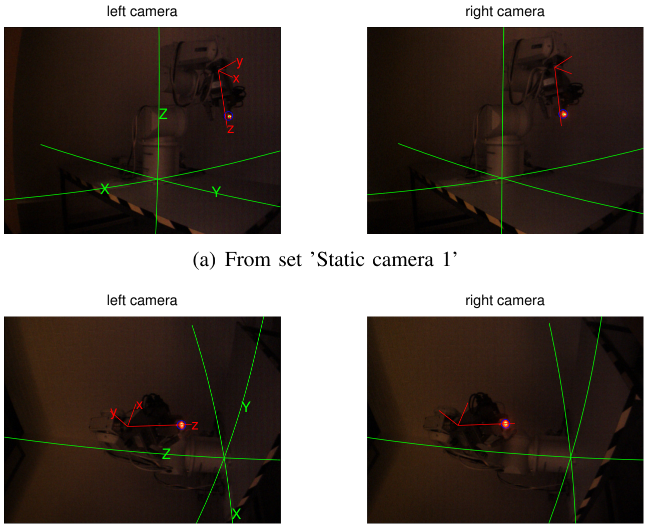
$$\begin{aligned} \mathbf{m}_{C_l} &= {}^{C_l}\mathbf{T}_R \, {}^R\mathbf{T}_{EE} \, \mathbf{m}_{EE} \\ \mathbf{m}_{C_r} &= {}^{C_r}\mathbf{T}_R \, {}^R\mathbf{T}_{EE} \, \mathbf{m}_{EE}. \end{aligned} \tag{16}$$

This way, we had a ”groundtruth” pose between two cameras measured with the camera calibration toolbox and we could compare the result between two different calibration methods.

Tests were repeated with the camera and the marker LED in several different locations. In the tests the robot arm was set in random joint positions until 50 such poses were found where both cameras found the position of the blinking LED in the same frame. Each test set includes 50 stereo image pairs with the marker location found with sub-pixel accuracy, however, in some of the images instead of the true LED marker location an erroneous reflection from a metallic part of the robot or surrounding environment was found instead. Examples of two test setups can be seen in Fig. 3.

The parameter estimation was initialized so that the origin of the robot base frame was assumed to be one meter directly in front of the camera, the marker LED 0.2m directly in front of the robot's end effector frame and the three axis-angle parameters were initialized randomly. This initialization strategy was used because there is a reasonable assumption for the two translations but the rotation matrix can be almost anything depending on which side of the robot the camera is.

Unless otherwise noted, the tests were run initially using the $L_1 - L_2$ M-estimator and after convergence the M-estimator was changed to Welsch with $c = 5.0$ to remove



(a) From set 'Static camera 1'

(b) From set 'Static LED 4'

Fig. 3. Examples of test setups. Green lines and capital letters X, Y and Z mark the axes of the robot base frame, red lines and small letters mark the axes of the end-effector frame and a blue circle marking the estimated position of the marker LED.

the effect of outliers. The value of $c = 5.0$ means that the weight of a measurement approaches zero for residuals larger than 10 pixels (see Fig. 2) and the value should in general be based on the resolution of the camera, accuracy of the forward kinematics of the robot and how accurately the marker can be detected. Note that the residuals have been calculated as euclidean distance between the detected and estimated marker positions of the marker, not separately for x and y coordinates.

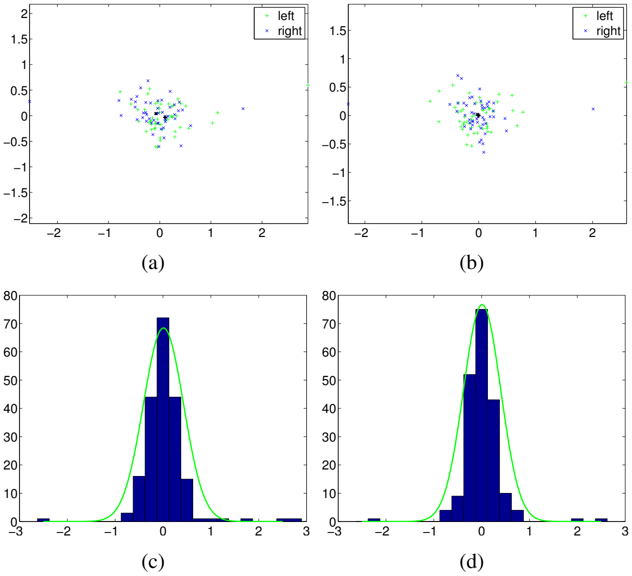
The speed of the parameter estimation was not of special interest, but a C implementation estimated parameters for 50 measurements in under one second with a few hundreds of Gauss-Newton iterations.

B. Gaussianity of errors

Propagating covariance requires that the errors are distributed roughly according to Gaussian distribution. Therefore, the distribution of residuals was the point of interest in this experiment. In the experiment non-robust least-squares estimation was used with a test-set where there were no outliers, the residuals were caused by imprecision of detecting the marker. In addition the potential bias caused by estimating only the position of the left camera and using the pre-calibrated pose between cameras, (15), and estimating both cameras separately, (16), was studied.

The results are presented in Fig. 4. Fig. 4(a)&(c) present results when estimating only one camera and Fig. 4(b)&(d) the results when estimating cameras separately. In both cases the residuals were roughly normally distributed, however, as is reflected by the kurtosis values (≈ 20) the distributions have sharper peaks and fatter tails [8]. On the other hand, because the residuals are not exactly Gaussian, least-squares estimation is not the optimal choice even in the absence of outliers.

There was a slight bias between the right and left cameras (Fig. 4(a), means marked with bold 'x' and '+'), the distance between means was 0.194 pixels. The bias is mainly caused by the fact that the marker LED is not truly a point and it is seen from slightly different points of view by the cameras, in the view of the right camera the marker is always left of the position compared to what the left camera sees. In the case where both cameras were estimated separately (Fig. 4(b)) there was no bias.



(a)

(b)

(c)

(d)

Fig. 4. Distribution of residuals; (a) residuals when estimating one camera; (b) residuals when estimating cameras separately; (c)&(d) histograms of residuals for both cases, green line presents the Gaussian distribution fit to the data.

C. Estimation accuracy in practice

These tests study the effect of changing some physical parameter, the position of the camera or the marker, and how repeatable the results are with different settings. Eight test sets in total were collected, each with 50 stereo image pairs.

In four tests the camera was kept in the same position and the marker LED was attached to a different position in the gripper. The results for these four test are presented in Table I. The ± values report the inaccuracy calculated from back-propagated variance of residuals. With 50 stereo image pairs there is maximum of 100 inliers. With the camera staying stationary in best case $\mathbf{t}_{x,y,z}$ would be identical in every test. Angle* is the average difference between rotation matrices to the other sets. The differences in the coordinates were a few millimeters and also the angular differences between the test sets were small, about $0.4°$. One thing to note is the large number of outliers in Set 3, which was caused by the LED being positioned so that its reflection was detected often instead of the actual LED.

In four of the tests the marker LED was kept in the same position and the camera was moved to other positions. The results are presented in Table II. In this experiment the

TABLE I

RESULTS WITH FOUR TEST SETS WHERE THE CAMERA HAS STAYED STATIONARY AND THE MARKER LED HAS BEEN CHANGED TO DIFFERENT POSITIONS. THE UNITS ARE MILLIMETERS.

	Set 1	Set 2	Set 3	Set 4
inliers	96	90	56	93
angle*	$0.444°$	$0.444°$	$0.508°$	$0.398°$
t_x	104.1 ± 1.2	100.3 ± 0.8	102.2 ± 1.2	103.9 ± 0.9
t_y	275.8 ± 1.6	283.8 ± 0.9	280.5 ± 1.4	281.9 ± 1.2
t_z	1379.4 ± 2.0	1373.3 ± 1.3	1380.8 ± 2.9	1377.5 ± 1.6
M_{EE_x}	5.8 ± 0.5	-13.0 ± 0.5	-5.5 ± 0.8	-0.3 ± 0.5
M_{EE_y}	8.8 ± 0.4	-29.5 ± 0.4	-9.5 ± 0.7	0.0 ± 0.4
M_{EE_z}	204.5 ± 1.1	215.4 ± 0.7	170.4 ± 1.0	214.7 ± 0.8

marker LED positions $\mathbf{m}_{EE_x,y,z}$ would be equal in the best case. In x and y directions the changes were very small, in the order of $0.5mm$, but in the z direction the changes were slightly larger in one of the test sets, Set 2, the difference was about $3mm$.

TABLE II

RESULTS WITH FOUR TEST SETS WHERE THE CAMERA HAS BEEN MOVED AND THE MARKER LED HAS BEEN KEPT IN THE SAME POSITION. THE UNITS ARE MILLIMETERS.

	Set 1	Set 2	Set 3	Set 4
inliers	96	100	93	96
t_x	-235.2 ± 0.7	-57.6 ± 0.7	-110.7 ± 0.6	529.2 ± 0.6
t_y	282.8 ± 0.8	258.2 ± 0.8	405.8 ± 0.6	216.8 ± 0.8
t_z	905.5 ± 1.2	1950.4 ± 2.7	1309.6 ± 1.2	1356.7 ± 1.4
M_{EE_x}	-0.3 ± 0.5	-0.3 ± 0.6	-0.9 ± 0.2	-0.7 ± 0.3
M_{EE_y}	-0.7 ± 0.5	-0.6 ± 0.4	-0.5 ± 0.2	-0.0 ± 0.3
M_{EE_z}	216.3 ± 0.7	212.8 ± 1.0	215.5 ± 0.5	216.2 ± 0.7

The results comparing estimation of joint rotation-translation between ${}^{C_r}\mathbf{T}_{C_l}$ cameras in a stereo system are presented in Table III. The translation between cameras estimated by the camera calibration toolbox for Matlab [2] was $(120.5, -0.5, 0.8)$mm with $0.30°$ difference between directions of the cameras (which are designed to be parallel). Average translation with the eight test sets estimated with (16) was $(118.6, 0.2, 0.8)$mm. The specifications of the Bumblebee 2 stereo camera state that the distance between cameras is 120mm and with both estimation methods the difference was below 2%. There was a slight constant bias between the results of the two estimation methods in the x and slightly also on y translation. The same bias was also noticeable in Fig. 4(a) and the underlying reason is probably the same – a non-point marker and a constant difference in camera viewpoints.

D. Required number of measurements

These experiments study the effect of used number of measurements (stereo image pairs). For each number of measurements and test set the tests were repeated 25 times.

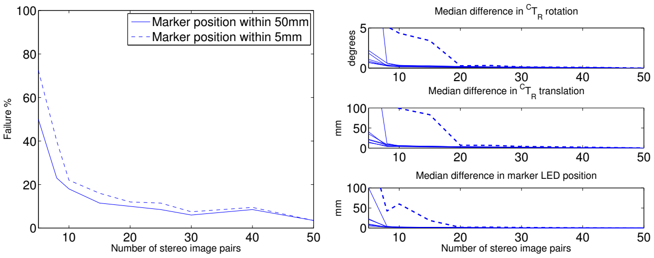
Fig. 5 shows the results when measuring the position of only the left camera, (15). All 8 datasets were included. Fig. 5(a) shows how often the estimated position of the marker LED was within $50mm$ or $5mm$ of the position estimated when using all 50 measurements. After 15 measurements the failure percent was fairly constant. There are 9 parameters to estimate and a single measurement gives 4 datapoints (x and y positions in two frames) so in theory three measurements are sufficient. When repeating the test with the full dataset, there still were some failures because the estimation starts from a partly random initialization and Gauss-Newton sometimes fails to converge. However, in those cases the errors were extremely large and the median difference was exactly zero, i.e., when the estimation succeeded the same parameters were always found.

TABLE III

RESULTS COMPARING ${}^{C_r}T_{C_l}$ ESTIMATED BY CAMERA CALIBRATION TOOLBOX [2] AND BY ESTIMATING TRANSFORMATION BETWEEN THE ROBOT BASE FRAME AND BOTH CAMERAS SEPARATELY, (16).

Test set	Difference in angle	Difference in translation
Static cam 1	$0.115°$	$(-1.56, 0.88, -1.33)$ mm
Static cam 2	$0.062°$	$(-1.49, 0.45, -0.87)$ mm
Static cam 3	$0.236°$	$(-1.72, 1.75, 3.59)$ mm
Static cam 4	$0.057°$	$(-0.36, 0.71, 0.17)$ mm
Static LED 1	$0.103°$	$(-1.10, 0.33, -0.22)$ mm
Static LED 2	$0.214°$	$(-5.35, 0.58, -2.49)$ mm
Static LED 3	$0.159°$	$(-1.25, 1.78, 0.29)$ mm
Static LED 4	$0.172°$	$(-2.37, -0.03, -0.31)$ mm

Fig. 5(b) shows the median differences in estimated ${}^{C}\mathbf{T}_R$ angle and translation and in marker position separately for all 8 datasets. Results for dataset 'Static camera 3' are highlighted, because the results differ from other datasets due to the dataset having a large number of outliers (as seen in Table I). That dataset required 20 measurements for the median errors to become near the values estimated with the full dataset, but in all other datasets 10, or even 5, measurements gave very low median errors (under 10mm for marker position and under 20mm for the robot-camera translation).

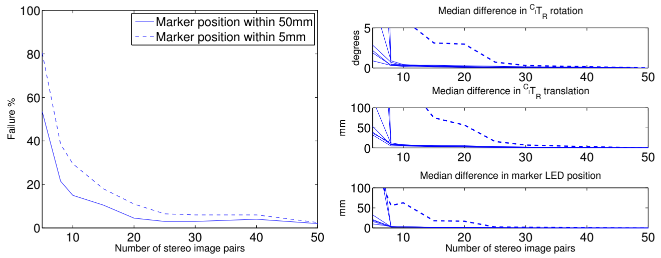


(a)

(b)

Fig. 5. The effect of number of measurements; (a) How often the estimation failed to find the marker position with specified accuracy; (b) Median differences in estimated ${}^{C}T_R$ angle and translation and in marker position for all 8 datasets, dataset 'Static camera 3' highlighted.

Similar tests were repeated also when estimating the position of the both cameras separately, (16). The results are presented in Fig. 6. For simplicity of the presentation results are presented only for the left camera. In this case there was 15 parameters to estimate, so theoretically 4 measurements are enough. With some test sets 5 measurements gave reasonably small median errors, but the increased need for measurements can be seen that for dataset 'Static camera 3' 25 measurements were needed instead of 20 which was enough in the previous experiment.

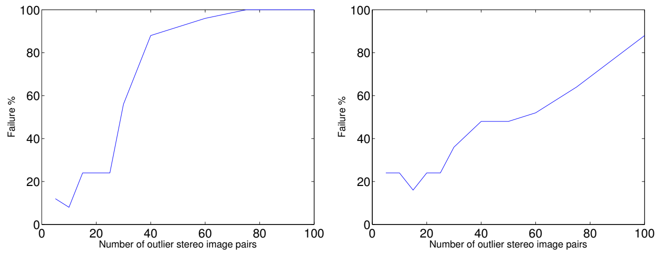


(a) (b)

Fig. 6. The effect of number of measurements when estimating position of both cameras; (a) How often the estimation failed to find the marker position with specified accuracy; (b) Median differences in estimated ${}^{C_l}T_R$ angle and translation and in marker position for all 8 datasets, dataset 'Static camera 3' highlighted.

E. Robustness to outliers

Here the effect of number of outliers is studied. In these tests 25 best inliers, those having the largest weight values after the estimation, from one of the test sets were selected and then a number of outliers was added and the parameter estimation was performed again. For each number of outliers the test was repeated 25 times with a different set of outliers. The results are shown in Fig. 7 where the graphs show how often the marker LED was not found within $5mm$ compared to the outlier-free estimation. In Fig. 7(a) the "outliers" are selected randomly from all other test sets and in Fig. 7(b) the marker positions are additionally randomized. In the first case the estimation begun to fail very often when the number of outliers grew over 25 but in the second case the estimation only grew linearly and still succeeded over 10% of the time with 100 outliers (4 times more than true inliers). With outliers selected from real data the estimation fails earlier because there actually are several valid parameter sets and the estimation may end up converging to a wrong one.



(a) (b)

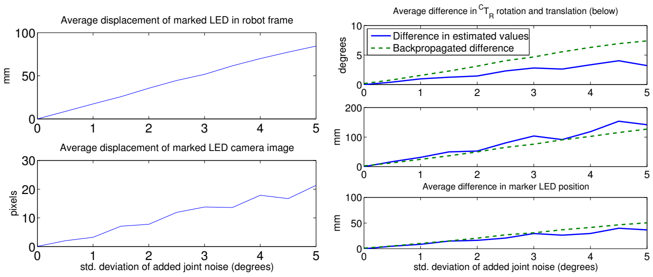
Fig. 7. The effect of number of added outliers with 25 valid inlier measurements; (a) Outliers added from other test sets; (b) Outliers in randomized marker positions.

Note that the performance in the second case (randomized marker positions) would be considerably higher if the Welsch M-estimator used some adaptive scheme for selecting the c parameter instead of constant $c = 5.0$.

F. Error back-propagation with artificial joint noise

This experiment studies error back-propagation when noise is artificially added to the joint values, which causes inaccuracy in ${}^{R}\mathbf{T}_{EE}$. The main interest here is the use of back-propagating variance of the residuals to the estimated parameters.

In these tests zero mean Gaussian noise was added to the joint values of the robot which are used to calculate the position of the marker LED in the base frame of the robot. To avoid having to tune the M-estimators for increasing amount of noise, a non-robust least-squares estimator was used. The test set having smallest number of outliers was therefore used, same as in Fig. 4. The results are presented in Fig. 8. The effect of added noise in the joint positions is presented in Fig. 8(a), both in the world coordinates and in camera pixels coordinates. Fig. 8(b) presents the differences to non-noisy estimates. Actual realized average differences are solid line and the dashed lines present the back-propagated variances from the variance of the residuals (pixel position errors), (14). Back-propagated variances were very close to the realized differences, except for the rotation matrix angle where the overestimated error is caused by the non-linear nature of the axis-angle presentation.



(a) (b)

Fig. 8. The effect of added joint noise; (a) Displacement caused to world coordinates and pixel values; (b) Differences to parameter estimates with no added noise.

V. Conclusions

In this work a new camera-robot calibration method was presented. The method is based on tracking a marker attached to the end-effector of the robot. The estimation can be performed for one or multiple cameras simultaneously and a known pose between cameras in a stereo system can also be used. The accuracy of the estimate can be established using back-propagation of the variance of the measurement residuals.

In the experiments various aspects of the parameter estimation procedure were studied. In tests where the camera or the marker was moved, it was noticed that the estimations agreed to within a few millimeters, as well as the stereo calibration result between this method and the camera calibration toolbox [2]. As few as five measurements were noticed to provide reasonably accurate estimates, but a larger number improves the result and makes the method robust to a large number of outliers. The back-propagation of the variance of the residuals to the estimated parameters was noticed to very accurately reflect the realized uncertainty in an experiment where noise was added to the joint positions.

Possible future improvements include changing the Gauss-Newton approximation method to more robust Levenberg-Marquardt algorithm and using an adaptive scheme for selecting the M-estimator parameters for cases where the measurement errors are unknown. The estimation of camera intrinsic parameters could also be easily added, which however would increase the number of required measurements considerably.

References

[1] M. Bauer, M. Schlegel, D. Pustka, N. Navab, and G. Klinker. Predicting and estimating the accuracy of n-occular optical tracking systems. In *Proceedings of Mixed and Augmented Reality (ISMAR)*, pages 43–51, 2006.

[2] J. Y. Bouguet. Camera calibration toolbox for matlab. http://www.vision.caltech.edu/bouguetj/calib_doc/, February 2011.

[3] F. Dornaika F and C. Garcia. Robust camera calibration using 2d-to-3d feature correspondences. In *Proceedings of Videometrics V*, pages 123–133, 1997.

[4] C. Garcia. Fully vision-based calibration of a hand-eye robot. *Autonomous Robots, Special issue on Perception-Based Intelligent Robots*, 6(2):223–238, 1999.

[5] X. Gratal, J. Bohg, M. Björkman, and D. Kragic. Scene representation and object grasping using active vision. In *IROS'10 Workshop on Defining and Solving Realistic Perception Problems in Personal Robotics*, 2010.

[6] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision, 2nd. edition*. Cambridge University Press, 2003.

[7] J. Heikkila and O. Silven. A four-step camera calibration procedure with implicit image correction. In *Proceedings of Int. Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 1106 –1112, June 1997.

[8] A. Hyvärinen, J. Karhunen, and E. Oja. *Independent Component Analysis*. John Wiley & Sons, 2001.

[9] A. Jordt, N. T. Siebel, and G. Sommer. Automatic high-precision self-calibration of camera-robot systems. In *Proceedings of Int. Conf. on Robotics and Automation (ICRA)*, pages 1244 –1249, May 2009.

[10] Bo Sun, Qing He, Chao Hu, and M.Q.-H. Meng. A new camera calibration method for multi-camera localization. In *Proceedings of Int. Conf. on Automation and Logistics (ICAL)*, pages 7 –12, 2010.

[11] B. Triggs, P. McLauchlan, R. Hartley, and A. Fitzgibbon. Bundle adjustment - a modern synthesis. In *Vision Algorithms: Theory and Practice*. Springer-Verlag Berlin, Germany, 2000.

[12] R.Y. Tsai and R.K. Lenz. Real time versatile robotics hand/eye calibration using 3d machine vision. In *Proceedings of Int. Conf. on Robotics and Automation (ICRA)*, pages 554 –561 vol.1, April 1988.

[13] L. Wang, F.C. Wu, and Z.Y. Hu. Multi-camera calibration with one-dimensional object under general motions. In *Proceedings of Int. Conf. on Computer Vision (ICCV)*, pages 1 –7, 2007.

[14] K. Welke, M. Przybylski, T. Asfour, and R. Dillmann. Kinematic calibration for saccadic eye movements. Technical report, Institute for Anthropomatics, Universität Karlsruhe, 2008.

[15] Z. Zhang. Parameter estimation techniques: a tutorial with application to conic fitting. *Image and Vision Computing*, 15(1):59–76, 1997.

[16] Z. Zhang. A flexible new technique for camera calibration. Technical report, Microsoft Research, 1998 (last update in 2009). http://research.microsoft.com/ zhang/Calib/.

[17] Z. Zhang. A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(11):1330 – 1334, November 2000.

Appendix

For Gauss-Newton approximation of the parameters (position of the marker $\mathbf{m}_{EE}$ and joint rotation/translation matrix from robot to camera frame ${}^{C}\mathbf{T}_R$) partial derivatives of (2) with respect to each parameter are needed. Applying the product rule of derivation it can be seen that partial derivatives of $\mathbf{m}_{EE}$ and ${}^{C}\mathbf{T}_R$ can be solved separately and the joint rotation-translation matrix ${}^{C}\mathbf{T}_R$ can be divided to its rotation and translation parts.

The partial derivative of a rotation matrix defined by axis-angle vector $\boldsymbol{\theta}$, (3), is

$$\begin{aligned}\frac{\partial \mathbf{R}(\boldsymbol{\theta})}{\partial \theta_i} = & -\theta_i \frac{\sin|\theta|}{|\theta|}\mathbf{I} + \\ & \theta_i\left(\frac{\cos|\theta|}{|\theta|^2} - \frac{\sin|\theta|}{|\theta|^3}\right)[\theta]_{\mathbf{x}} + \\ & \frac{\sin|\theta|}{|\theta|}\frac{\partial[\theta]_x}{\partial \theta_i} + \\ & \theta_i\left(\frac{\sin|\theta|}{|\theta|^3} + 2\frac{\cos|\theta|-1}{|\theta|^4}\right)\theta\theta^T + \\ & \frac{1-\cos|\theta|}{|\theta|^2}\frac{\partial(\theta\theta^T)}{\partial\theta_i}\end{aligned} \tag{17}$$

where the partial derivatives of $\frac{\partial[\theta]_x}{\partial\theta_i}$ can be solved trivially from (4). Partial derivatives of the translation vector and the position of the marker are also trivial, for example, $\frac{\partial \mathbf{t}}{\partial t_x} = [1,0,0]^T$. The complete partial derivatives of (2) are

$$\begin{aligned}\frac{\partial \mathbf{m}_C}{\partial \theta_i} &= \begin{bmatrix} \frac{\partial \mathbf{R}(\theta)}{\partial \theta_i} & 0 \\ 0 & 0 \end{bmatrix} {}^{R}\mathbf{T}_{EE}\,\mathbf{m}_{EE} \\ \frac{\partial \mathbf{m}_C}{\partial t_{x,y,z}} &= \begin{bmatrix} 0 & \frac{\partial \mathbf{t}}{\partial t_{x,y,z}} \\ 0 & 0 \end{bmatrix} {}^{R}\mathbf{T}_{EE}\,\mathbf{m}_{EE} \\ \frac{\partial \mathbf{m}_C}{\partial m_{EE_{x,y,z}}} &= {}^{C}\mathbf{T}_R\,{}^{R}\mathbf{T}_{EE}\,\frac{\partial \mathbf{m}_{EE}}{\partial m_{EE_{x,y,z}}}\end{aligned} \tag{18}$$

A Jacobian is formed from the partial derivatives and is used during Gauss-Newton approximation. The partial derivative of a rotation matrix is not a proper rotation matrix, e.g., the sum of rows/columns is not necessarily 1, and the results in general are not in homogeneous coordinates as the last value of the resulting vector is zero.

To estimate the parameters with a maximum-likelihood estimator the derivatives in the camera coordinates are also needed to calculate to which direction the marker would move in camera pixel coordinates if a of the model parameter is adjusted. For this derivatives of (7) are needed, i.e, $\frac{\partial u}{\partial X}$ where X is one of the model parameters. As the equations are the same for all parameters a simplified notation is used, where $\frac{\partial u}{\partial X} = u'$. The derivatives can be calculated from (7) and

$$\begin{aligned}\begin{bmatrix} \dot{a}' \\ \dot{b}' \end{bmatrix} = & \begin{bmatrix} a' \\ b' \end{bmatrix}(1 + k_1 r_2 + k_2 r_2^2 + k_3 r_2^3) + \\ & \begin{bmatrix} a \\ b \end{bmatrix}(k_1 r_2' + 2k_2 r_2 r_2' + 3k_3 r_2^2 r_2') + \\ & \begin{bmatrix} 2p_1(a'b + ab') + p_2(r_2' + 4aa') \\ p_1(r_2' + 4bb') + 2p_2(a'b + ab') \end{bmatrix} \\ u' = & f_x \dot{a}' \\ v' = & f_y \dot{b}',\end{aligned} \tag{19}$$

where $a' = \frac{zx' - xz'}{z^2}$, $b' = \frac{zy' - yz'}{z^2}$ and $r_2' = \frac{z^2(2xx' + 2yy') - 2zz'(x^2 + y^2)}{z^4}$.