Hand-projector Self-calibration Using Structured Light

Christian S. Wieghardt and Bernardo Wagner
Institute for Systems Engineering, Real Time Systems Group
Leibniz Universität Hannover, Appelstrae 9A, D-30167 Hannover, Germany

Keywords: Self-calibration, Pattern Projection, Structured Light System, Motion Estimation.

Abstract: This paper describes two methods for determining the extrinsic calibration parameters of a projector with respect to the robot hand. One of them simultaneously solves the transformation between a camera with respect to the robot base. Self-calibration means that no sort of calibration rig like a chessboard is needed. Since the projector has no exteroceptive capabilities, a camera is placed in the environment or rigidly attached to the robot base to detect the projected pattern. At different robot configurations correspondences between the camera and projector are established to recover the transformation between them up to an unknown scale factor. The common known formulations $\mathbf{AX} = \mathbf{XB}$ and $\mathbf{AX} = \mathbf{ZB}$ can be arranged in a linear form with respect to the unknown extrinsic parameters and scale factors, and solved in least square sense.

1 INTRODUCTION

In robotics, a projector is typically used as an interface or as a sensor in combination with at least one camera. Recent developments have lead to a significant decrease in projector size, which facilitates the usage in mobile applications. In our case, we are interested in tracking the robot end-effector by means of the projector. Because of the mobility we propose the set-up shown in Figure 1. In order to apply a projector appropriately on a robot hand, its homogeneous transformation to the manipulator has to be determined. This external calibration is addressed in this paper.

The main challenge is the restraint of calibration rigs. They are not allowed by definition of self-calibration. For calibration it is essential to move the robot arm to different poses and capture the motions of the projector. But with no known objects or constraints it is impossible to link the projector to the environment. However, by detecting the relative pose between the camera and the projector the hand-projector transformation is still recoverable.

The projector can be described as the inverse of the pinhole model - equal to most cameras. Therefore we can identify the cameras and projectors intrinsic parameters with camera calibration based methods e.g. (Zhang, 2000). We assume that these internal unknowns are determined beforehand with one of the structured light system calibration methods, such as (Zhang and Huang, 2006) or (Moreno and Taubin, 2012). The camera duality furthermore enables the use of techniques known from existing Hand-Eye-Calibrations. This problem was first addressed in (Shiu and Ahmad, 1989) and (Tsai and Lenz, 1989). Later works considered simultaneous robot-workspace calibration (Zhuang et al., 1994), (Dornaika and Horaud, 1998), which allows the calibration of an external camera. Of interest to our work is (Andreff et al., 2001), where an unknown scale fac-

Figure 1: An LED pico projector AXAA P2 Jr mounted on the KUKA youBots manipulator and a uEye camera UI-1241LE-C-HQ on its platform.

Wieghardt C. and Wagner B..
Hand-projector Self-calibration Using Structured Light.
DOI: 10.5220/0005065200850091
In *Proceedings of the 11th International Conference on Informatics in Control, Automation and Robotics* (ICINCO-2014), pages 85-91
ISBN: 978-989-758-039-0

tor in the camera motion is incorporated.

So far, there are few publications with hand-mounted projectors, that are usually calibrated only to a second device on the robot arm. In (Reinhart et al., 2007) a tracking system is used to manually assign the projected points to world coordinates. Calibration was implemented with respect to the robot's tool flange. In (De Tommaso et al., 2012) the projector is calibrated to an rgb-d camera via homographies to planes observed by both devices. Calibration to the end-effector was not concerned. Even though no robot arm was applied, (Gavaghan et al., 2011) utilized a projector to reference system calibration. Markers were mounted on the device and on a calibration plane to detect both movements with a stereo camera. Extrinsic and intrinsic calibration was done by projecting chessboard patterns on the plane and identifying them with the stereo camera.

Notice that an additional camera mounted beside the projector on the end-effector would result in a structured light system. That second device would ease the presented calibration, because the scale is given by the baseline. But we argue the renouncement of the camera with payload and space limitations of the end-effector. The device is also not necessary when tracking the robot hand by means of the projector, on which we will focus in future research. Therein we will increase the number of cameras to obtain higher accuracy and robustness. So the reverse set-up, where the camera is mounted on the end-effector and the projector placed e.g. at the robot base, is not relevant for us.

The presented method eases the calibration procedure since the calibration rig can be omitted. Thus, no limitations are arising by the necessity of fully covering the rig by the projection during the determination of good calibration robot arm poses. Furthermore, all projector-camera correspondences can be used for calibration.

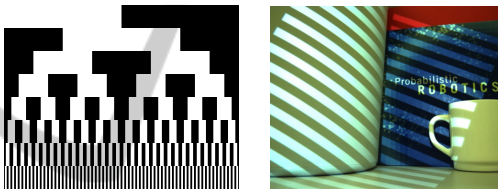
The paper is organized as follows: In Section 2 it is shown how to obtain the transformation between the camera and the projector up to an unknown scale factor. Those are needed for the calibration procedure discussed in Section 3. We then present the experimental results in Section 4, and finally Section 5 offers a conclusion and an outlook to some future works.

2 RELATIVE POSE ESTIMATION

This section introduces a procedure to establish the relative pose between the camera and the projector. Due to the unknown environment the translation can only be recovered up the scale factor. This reduces the homogeneous transformation between the camera and the projector to a three-dimensional rotation and a two-dimensional translation. As it is assumed that the camera's and projector's intrinsic parameters are known, five unknown parameters need to be determined.

2.1 Establishing Correspondences

First of all, correspondences between the projector and the camera are established by projecting vertical and horizontal fringe patterns onto the environment. The sequence of patterns is coded to match the projector points $\mathbf{x}_p = (x_p, y_p, 1)^T$ with the camera points $\mathbf{x}_c = (x_c, y_c, 1)^T$. We use Gray coded patterns, as they are easy to implement and allow for high density of points (Salvi et al., 2004). The sequence of the Gray code in one direction is shown in Figure 2 (a) and one typical captured image of fringes consisting patterns is given in Figure 2 (b).



(a) (b)

Figure 2: (a): Gray code, (b): One image of the Gray coded image sequence.

The pattern is temporary coded, thus the robot hand has to keep its pose during the image capturing. The projector pixels $\mathbf{x}_p$ are given by the Gray code. However, to obtain the camera pixels $\mathbf{x}_c$, decoding has to be done. For each image every pixel illumination (on/off) is specified by comparison to a reference image. The Gray code position is defined by the vertical and horizontal sequence of illumination. The mean of the covered pixels gives $\mathbf{x}_c$.

2.2 Determining the Rotation and the Direction of Translation

The epipolar geometry describes the intrinsic projective geometry for two views. This is also applicable for our camera/projector combination since the projector is modelled as an inverse camera. The epipolar geometry depends only on the intrinsic parameters and the relative pose, not on the environment's structure - an essential property for self-calibration. The fundamental matrix $\mathbf{F}$ relates the image points to-

gether:

$$\mathbf{x}_p^T \mathbf{F} \mathbf{x}_c = 0. \quad (1)$$

$\mathbf{F}$ can be transformed in a linear normalized form, filled with all correspondences and it can be solved in least square sense (Hartley, 1997). Notice that even though the fundamental matrix encapsulates the internal parameters, it can also be determined without them. The essential matrix $\mathbf{E}$ will be recovered with the projector's and camera's intrinsic matrices $\mathbf{K}_p$ and $\mathbf{K}_c$:

$$\mathbf{E} = \mathbf{K}_p^T \mathbf{F} \mathbf{K}_c. \quad (2)$$

The essential matrix $\mathbf{E}$ has five degrees of freedom, depends only on the pose between the camera and the projector and can be also expressed by:

$$\mathbf{E} = [\mathbf{t}]_x \mathbf{R}, \quad (3)$$

where $[\mathbf{t}]_x$ is an antisymmetric matrix defined as

$$[\mathbf{t}]_x = \begin{pmatrix} 0 & -t_z & t_y \\ t_z & 0 & -t_x \\ -t_y & t_x & 0 \end{pmatrix}. \quad (4)$$

$\mathbf{R}$ and $\mathbf{t}$ can be recovered from $\mathbf{E}$ and $\mathbf{t}$ up to scale. (Hartley and Zisserman, 2000) extract out of the singular value decomposition of $\mathbf{E}$:

$$\mathbf{E} = \mathbf{U} \Sigma \mathbf{V}^T \quad (5)$$

the four solutions:

$$(\mathbf{R}, \mathbf{t}) = (\mathbf{U}\mathbf{W}\mathbf{V}^T, \pm \mathbf{u}_3), \mathbf{W} = \begin{pmatrix} 0 & \pm 1 & 0 \\ \mp 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \quad (6)$$

one for which all points appear in front of the devices. The solution of (6) can be further refined via nonlinear optimization methods. We minimize the reprojection error:

$$\underset{\mathbf{R},\mathbf{t}}{\operatorname{argmin}} \sum_i d(\mathbf{x}_{ci}, \hat{\mathbf{x}}_{ci})^2 + d(\mathbf{x}_{pi}, \hat{\mathbf{x}}_{pi})^2 \quad (7)$$

by means of the Levenberg-Marquardt algorithm, implemented in (Lourakis, 2004).

3 CALIBRATION METHOD

In this Section we introduce two linear formulations of the hand-projector calibration, one of which includes simultaneous robot-camera calibration. Therefore we alter the hand-eye and robot-workspace calibrations to support our calibrations by taking the unique scale factors for each pose into account. This procedure follows (Andreff et al., 2001), who introduces a linear formulation of the hand-eye calibration $\mathbf{A}_i(\lambda)\mathbf{X} = \mathbf{X}\mathbf{B}_i$ for structure-from-motion methods, thus containing one scale factor.

3.1 Hand-projector

For hand-projector calibration we need to determine $\mathbf{X}$ out of the common equation:

$$\mathbf{A}_i(\lambda_k, \lambda_l)\mathbf{X} = \mathbf{X}\mathbf{B}_i, \quad (8)$$

which is supplemented with the unknown scale factors - see Figure 3. $\mathbf{A}_i$ are the projectors and $\mathbf{B}_i$ the robot end-effectors movements. Movement i is defined by transformation from pose k to l:

$$\mathbf{A}_i(\lambda_k, \lambda_l) = \mathbf{P}_k^{-1}(\lambda_k)\mathbf{P}_l(\lambda_l) \quad (9)$$

$$= (\mathbf{R}_{ak}^T \mathbf{R}_{al}, \mathbf{R}_{ak}^T(\lambda_l \mathbf{u}_{al} - \lambda_k \mathbf{u}_{ak})). \quad (10)$$

Equation (8) can be split into a rotational and translational part:

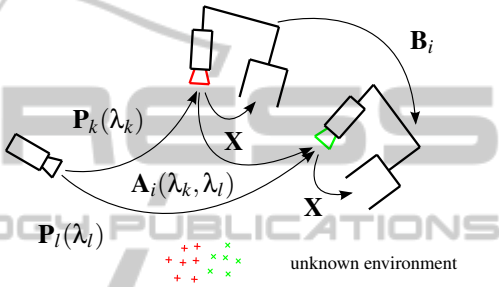


Figure 3: A projector is mounted on the end-effector and a camera placed in the environment. For each movement i the projected points change, involving different λs.

$$\mathbf{R}_{ai}\mathbf{R}_x = \mathbf{R}_x\mathbf{R}_{bi} \quad (11)$$

$$\mathbf{R}_{ai}\mathbf{t}_x + \mathbf{R}_{ak}^T(\lambda_l \mathbf{u}_{al} - \lambda_k \mathbf{u}_{ak})) = \mathbf{R}_x\mathbf{t}_{bi} + \mathbf{t}_x. \quad (12)$$

Using the vector operator:

$$vec(\mathbf{R}) = (\mathbf{R}_{11}, \ldots, \mathbf{R}_{1n}, \mathbf{R}_{2,1}, \ldots, \mathbf{R}_{mn}), \quad (13)$$

which rearranges matrices into vectors and its property on matrix multiplications (Brewer, 1978):

$$vec(\mathbf{CDE}) = (\mathbf{C} \otimes \mathbf{E}^T) vec(\mathbf{D}). \quad (14)$$

Equation (11) can be arranged in a linear form with respect to the unknown paramters:

$$vec\left(\mathbf{R}_{ai}\mathbf{R}_x (\mathbf{R}_{bi})^T\right) = vec(\mathbf{R}_x) \quad (15)$$

$$(\mathbf{R}_{ai} \otimes \mathbf{R}_{bi})\, vec(\mathbf{R}_x) = vec(\mathbf{R}_x) \quad (16)$$

$$(\mathbf{R}_{ai} \otimes \mathbf{R}_{bi})\, vec(\mathbf{R}_x) - \mathbf{I}_9 vec(\mathbf{R}_x) = \mathbf{0}_{9\times 1}. \quad (17)$$

The same applies to the translational part (12):

$$\left(\mathbf{I}_3 \otimes \mathbf{t}_{bi}^T\right) vec(\mathbf{R}_x) + \mathbf{t}_x - \mathbf{R}_{ai}\mathbf{t}_x \quad (18)$$

$$-\mathbf{R}_{ak}^T(\lambda_l \mathbf{u}_{al} - \lambda_k \mathbf{u}_{ak})) = \mathbf{0}_{3\times 1}. \quad (19)$$

Both parts result in the overall system:

$$\begin{pmatrix} \mathbf{I}_9 - \mathbf{R}_{ai} \otimes \mathbf{R}_{bi} & \mathbf{0}_{9\times 3} & \mathbf{0}_{9\times N} \\ \mathbf{I}_3 \otimes \mathbf{t}_{bi}^T & \mathbf{I}_3 - \mathbf{R}_{ai} & -\mathbf{R}_{ak}^T \mathbf{U}_{ai} \end{pmatrix} \begin{pmatrix} vec(\mathbf{R}_x) \\ \mathbf{t}_x \\ \Lambda \end{pmatrix} = \mathbf{0}_{12\times 1} \quad (20)$$

with the scale-free translation matrix and the corresponding unknown scale factors:

$$\mathbf{U}_{ai} = \mathbf{u}_{al} e_l - \mathbf{u}_{ak} e_k, \qquad \Lambda = (\lambda_1, \ldots, \lambda_N)^T. \quad (21)$$

3.2 Hand-projector and Camera-robot

Basis is the common equation:

$$\mathbf{A}_i(\lambda_i)\mathbf{X} = \mathbf{Z}\mathbf{B}_i, \quad (22)$$

which is supplemented with the unknown scale factors as well - see Figure 4. Here, $\mathbf{A}_i$ are the projector-camera and $\mathbf{B}_i$ the robot end-effector poses. Equation (22) can be split into a rotational and translational part:

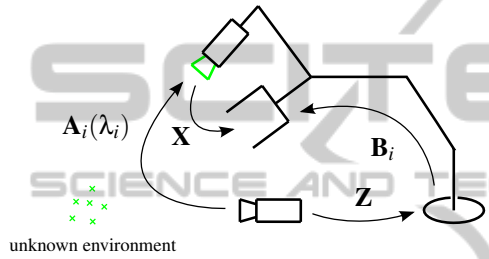


Figure 4: A projector is mounted on the end-effector and a camera is placed in the environment or attached to robot platform. The camera optical frame coincides with the world frame. For each pose i the projected points change, resulting in a new λ_i.

$$\mathbf{R}_{ai}\mathbf{R}_x = \mathbf{R}_z\mathbf{R}_{bi} \quad (23)$$

$$\mathbf{R}_{ai}\mathbf{t}_x + \lambda_i\mathbf{u}_{ai} = \mathbf{R}_z\mathbf{t}_{bi} + \mathbf{t}_z. \quad (24)$$

Using the vector operator and the property (Brewer, 1978) equation (23) can be arranged in a linear form with respect to the unknown paramters:

$$(\mathbf{R}_{ai} \otimes \mathbf{R}_{bi})\, vec(\mathbf{R}_x) - \mathbf{I}_9 vec(\mathbf{R}_z) = \mathbf{0}_{9\times 1}. \quad (25)$$

The same applies to the translational part (24):

$$\left(\mathbf{I}_3 \otimes \mathbf{t}_{bi}^T\right) vec(\mathbf{R}_z) + \mathbf{t}_z - \mathbf{R}_{ai}\mathbf{t}_x - \lambda_i\mathbf{u}_{ai} = \mathbf{0}_{3\times 1}. \quad (26)$$

Both parts result in the overall system:

$$\begin{pmatrix} \mathbf{R}_{ai}\otimes\mathbf{R}_{bi} & -\mathbf{I}_9 & \mathbf{0}_{9\times3} & \mathbf{0}_{9\times3} & \mathbf{0}_{9\times N} \\ \mathbf{0}_{3\times9} & \mathbf{I}_3\otimes\mathbf{t}_{bi}^T & -\mathbf{R}_{ai} & \mathbf{I}_3 & -\mathbf{U}_{ai} \end{pmatrix} \begin{pmatrix} vec(\mathbf{R}_x) \\ vec(\mathbf{R}_z) \\ \mathbf{t}_x \\ \mathbf{t}_z \\ \Lambda \end{pmatrix} = \mathbf{0}_{12\times1} \quad (27)$$

with the scale-free translation matrix and the corresponding unknown scale factors:

$$\mathbf{U}_i = \mathbf{u}_{ai} e_i, \qquad \Lambda = (\lambda_1, \ldots, \lambda_N)^T. \quad (28)$$

3.3 Formulations with Known Scale

Here we give the final single linear systems to avoid duplicity. Following previous sections, the formulations are straightforward to get. With fully known $\mathbf{t}_{ai}$ (20) reduces to:

$$\begin{pmatrix} \mathbf{I}_9 - \mathbf{R}_{ai}\otimes\mathbf{R}_{bi} & \mathbf{0}_{9\times3} \\ \mathbf{I}_3\otimes\mathbf{t}_{bi}^T & \mathbf{I}_3 - \mathbf{R}_{ai} \end{pmatrix} \begin{pmatrix} vec(\mathbf{R}_x) \\ \mathbf{t}_x \end{pmatrix} = \begin{pmatrix} \mathbf{0}_{9\times1} \\ \mathbf{t}_{ai} \end{pmatrix}. \quad (29)$$

and (27) reduces to:

$$\begin{pmatrix} \mathbf{R}_{ai}\otimes\mathbf{R}_{bi} & -\mathbf{I}_9 & \mathbf{0}_{9\times3} & \mathbf{0}_{9\times3} \\ \mathbf{0}_{3\times9} & \mathbf{I}_3\otimes\mathbf{t}_{bi}^T & -\mathbf{R}_{ai} & \mathbf{I}_3 \end{pmatrix} \begin{pmatrix} vec(\mathbf{R}_x) \\ vec(\mathbf{R}_z) \\ \mathbf{t}_x \\ \mathbf{t}_z \end{pmatrix} = \begin{pmatrix} \mathbf{0}_{9\times1} \\ \mathbf{t}_{ai} \end{pmatrix}. \quad (30)$$

3.4 Solvability

From (20) we see, that we have $12+i$ unknowns and can solve for $12i$. In (27) we have $24+i$ unknowns and can also solve for $12i$. Thus, at least three robot arm poses have to be approached in both methods to solve the homogeneous systems. Furthermore, we notice from (Tsai and Lenz, 1989) that at least two movements with nonparallel rotational axes are necessary to get a unique solution of the hand-eye calibration. This also applies for the cases with known scale (29) and (30).

The homogeneous systems (20) and (27) are linear in the parameters and can be solved in least square sense. Optionally, one can separately estimate the rotational parameters $\mathbf{R}_x$ and $\mathbf{R}_z$ of the upper part and afterwards the translational parameters $\mathbf{t}_x$, $\mathbf{t}_z$ and Λ of the lower part. Separating is reasonable having good rotational data. Otherwise the error would propagate into the translational data.

4 EXPERIMENTAL RESULTS

In this section the performances of both proposed self-calibration formulations are presented. Further comparison is given by their linear calibration equivalent without any scale factors. All methods were carried out in simulation and with real hardware.

Simulations validate the algorithm based on ground truths and show the effect resulting from noisy transformation data of the robot arm and the camera-projector pair. Due to the increase of unknown parameters by one for each pose, less accurate results are to be expected. This influence of the parameter growth will also be shown in simulation by means of their equivalents.

Since the physical robot arm and camera data is strongly noisy, we also give a comparison to the

equivalents with the help of a chessboard. This outlines the achievable accuracy.

For evaluation, we used 135 different robot arm configurations. The poses for computation were randomly selected in a clustered manner to enlarge translation and rotation between the poses. We calculated the rotational error $e_{\mathbf{R}}$ of the estimated $\hat{\mathbf{R}}$ by recovering its rotation vector and determine the angle between its ground truth $\mathbf{R}_{gt}$, so that $0 \leq e_{\mathbf{R}} \leq \pi$ is fulfilled. The average transformations given by:

$$\bar{\mathbf{t}} = \sum_{i=1}^{N} \frac{||\mathbf{t}_{ai}|| + ||\mathbf{t}_{bi}||}{2n}. \tag{31}$$

Thus, the relative error results in $e_{\mathbf{t}} = ||\hat{\mathbf{t}} - \mathbf{t}_{gt}||/\bar{\mathbf{t}}$. We took all possible movements between the poses into account, leading to $(N-1)N/2$ homogeneous transformations. All four applied formulations are listed in Table 1 and referenced in the following by their number.

Table 1: Applied Methods.

#1	Hand-Projector $\mathbf{A}_i(\lambda_k, \lambda_l)\mathbf{XXB}_i$
#2	Hand-Projector and Camera-Robot $\mathbf{A}_i(\lambda_i)\mathbf{XZB}_i$
#3	Hand-Projector, known scale $\mathbf{A}_i\mathbf{XXB}_i$
#4	Hand-Projector and Camera-Robot, known scale $\mathbf{A}_i\mathbf{XZB}_i$

4.1 Synthesized Data

For our simulation, the robot arm poses $\mathbf{B}_i$ are uniformly placed on a circle and alternately rotated around two distinguished axes at each position. This arrangement is based on the later used manipulator's workspace. With known $\mathbf{X}$ and $\mathbf{Z}$ the projector-camera transformation is given by:

$$\mathbf{A}_i = \mathbf{ZB}_i\mathbf{X}^{-1}. \tag{32}$$

We considered two sources of error: Noise in translation and noise in rotation. Both are applied to the robot arm poses and the camera-projector transformations. For translation and rotation distribution we randomly generated unit vectors with zero mean. Either of them were multiplied by their corresponding Gaussian distributed standard deviation $\sigma_{\mathbf{R}}$ and $\sigma_{\mathbf{t}}$.

First, we show the dependency of transformation perturbation in Figure 5 for $N = 4$ poses. A good criterium is given by the relative position error of $\mathbf{X}$, since it is calculated for all four methods. By comparing $\bar{e}(\mathbf{t}_{x_{\#1-4}})$ we can conclude, that the most robust solutions against translational error are #3 and #4, even though #4 determines twice as many parameters as #3. Whereas, incorporating the unknown scale factors Λ lead to a noticeable increase of errors in #1 and #2. Thus, at no expense of accuracy in $\mathbf{X}$ one can additionally solve for the camera pose $\mathbf{Z}$. The errors of $\mathbf{X}$ and $\mathbf{Z}$ depend on the set-up and the used manipulator configurations, therefore usually deviate from each other.

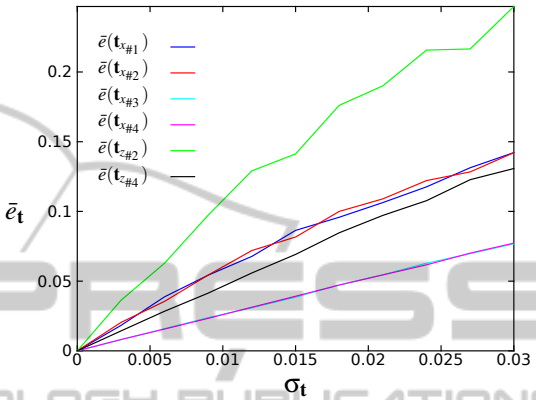


Figure 5: Average relative error of $\mathbf{t}_x$ and $\mathbf{t}_z$.

In Figure 6 we depicted the average relative error for different numbers of poses. We deviated the camera and robot arm poses with $\sigma_{\mathbf{t}} = 0.01$. It is shown, that even for a high number of used poses, a gap between the solutions #1 $\leftrightarrow$ #3 and #2 $\leftrightarrow$ #4 remains due to the unknown scale factors. Again, the comparison of #1 $\leftrightarrow$ #2 and #3 $\leftrightarrow$ #4 reveals that solving for the unknown $\mathbf{Z}$ does not effect the accuracy.

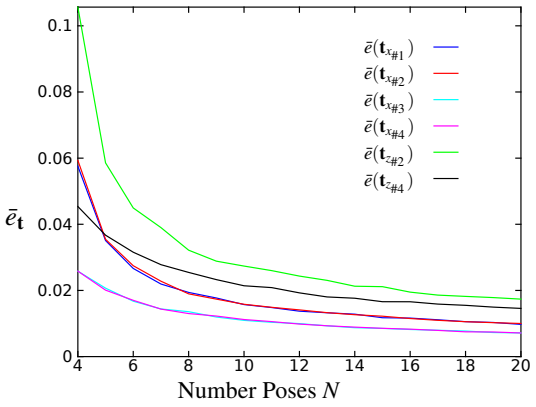


Figure 6: Average relative error of $\mathbf{t}_x$ and $\mathbf{t}_z$ at $\sigma_{\mathbf{t}} = 0.01$.

The dependency of rotation perturbation in shown Figure 7 for $N = 4$ poses. It can bee clearly seen that the rotation uncertainty has almost the same influence on all methods. That is due to the high weighted rotational parts of the linear systems. Since the upper parts of systems (20) resp. (27) coincide with (29)

resp. (30) the average rotational errors are mostly dependent of $\mathbf{R}_{ai}$ and $\mathbf{R}_{bi}$.

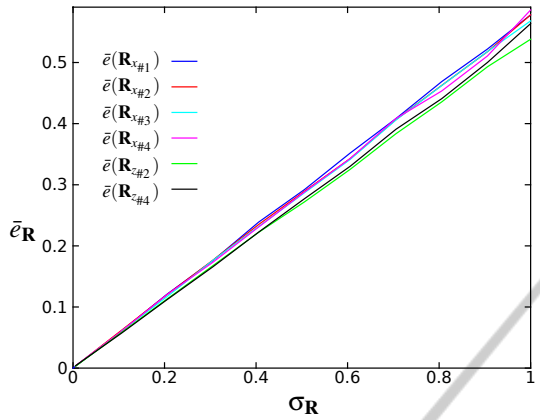


Figure 7: Average rotational error of $\mathbf{R}_x$ and $\mathbf{R}_z$.

In Figure 8 we depicted the average relative error for different numbers of poses. We deviated the camera and robot arm poses with $\sigma_{\mathbf{R}} = 1.0°$. As assumed, the congruence continues at more applied poses.

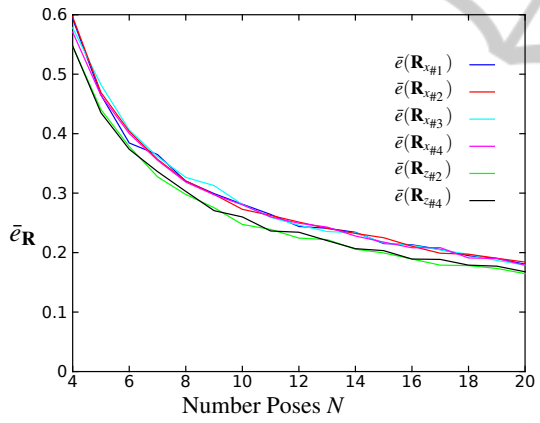


Figure 8: Average rotational error of $\mathbf{R}_x$ and $\mathbf{R}_z$ at $\sigma_{\mathbf{R}} = 1.0°$.

4.2 Real Data

The experimental set-up is shown in Figure 1. The KUKA youBot articulated robot arm consists of five serial rotary joints. Due to the robot arm kinematic, the end-effector poses are limited to those planes with the axis of the first joint as a common line. The LED pico projector AXAA P2 Jr at the end-effector has a resolution of 1920x1080 pixels. The uEye camera UI-1241LE-C-HQ from IDS has a resolution of 1280x1024 pixels. The presented hand-projector configuration is convenient since the projector is pointed at the workspace. For full calibration, two distinct rotations between the poses are necessary. This can become a difficult task, because the projector has to be pointed to the same workspace in all poses. A good rotation axis is given by the last joint since the projector's view is just slightly changing. The second rotation is given by the second to fourth joints. As no ground truth is given, we select $n = 1000$ sets of $N = 20$ poses and take the average solution.

To show the ability of self-calibration, the methods #1 and #2 were applied in an unknown environment with few desktop items - see Figure 2 (b). For methods #3 and #4 we established the projectors motion by means of a chessboard, placed in the environment. Local homographies are used to determine the homogeneous transformation between the projector and the camera to the chessboard (Moreno and Taubin, 2012). The results are given in Figure 9 and Figure 10.

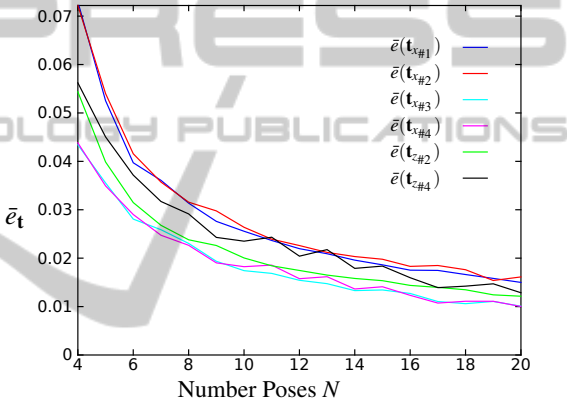


Figure 9: Average relative error of $\mathbf{t}_x$ and $\mathbf{t}_z$ over the number of poses N.

As already shown with help of the synthesized data, the errors of $\mathbf{X}$ through #1 and #2 respectively #3 and #4 coincide. Notice that in the simulation we used the transformations $\mathbf{X}$ and $\mathbf{Z}$ determined from the real data. The main difference is the restricted workspace of the manipulator. With these changed manipulator configurations the resulting relation of accuracies shifts. Compared with $\mathbf{Z}$, the error of $\mathbf{X}$ increases in position and decreases in orientation. The effect appears in a different amount for the methods, since #1 and #2 used slightly different configurations than #3 and #4.

In this experiment, we emphasized the achievable accuracy of the applied hardware. Even though the solutions converge well compared to their movements, some errors remain. Unfortunately, the biggest source of noise can not be assigned.

As assumed, the use of chessboard gives more accurate results, since full pose data has been used. Figure 10 confirms the good rotational data, following the procedure of Section 2.

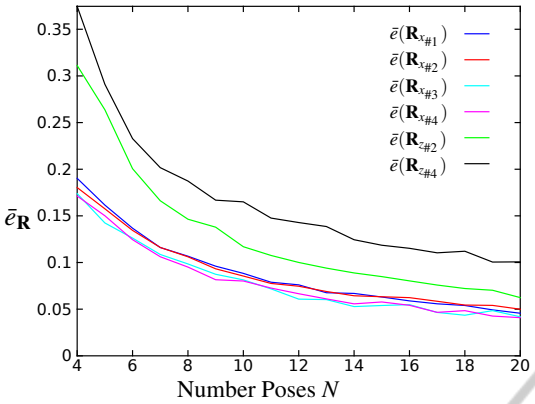


Figure 10: Average rotational error of $\mathbf{R}_x$ and $\mathbf{R}_z$ over the number of poses N.

5 CONCLUSIONS

This paper presents two self-calibration methods for externally calibrating a projector to a robotic hand. One of them additionally gives the solution of the transformation between the robot base and an external camera.

In contrast to existing approaches, the presented solutions need neither any additional device at the end-effector nor any calibration unit in the environment. The capability is demonstrated by the positive results of using real hardware. Provided with ground truth data, simulations prove the accuracy of the methods.

It has been shown that the error that results from the unknown scale factors quickly converges by increasing the number of poses. The additional error resulting from simultaneously estimating the camera pose is negligible. This is beneficial since the camera-projector pair can be used as a structured light sensor.

Future works will tackle optimizing of the solution and identifying good robot arm poses for calibration. Spatial coded light will be considered, as it enables us to use the method in dynamic environments.

REFERENCES

Andreff, N., Horaud, R., and Espiau, B. (2001). Robot hand-eye calibration using structure-from-motion. *International Journal of Robotics Research*, 20(3):228–248.

Brewer, J. W. (1978). Kronecker products and matrix calculus in system theory. *IEEE Trans Circuits Syst*, CAS-25(9):772–781.

De Tommaso, D., Calinon, S., and Caldwell, D. (2012). Using compliant robots as projective interfaces in dynamic environments. *Lecture Notes in Computer Science*, 7621:338–347.

Dornaika, F. and Horaud, R. (1998). Simultaneous robot-world and hand-eye calibration. *IEEE Transactions on Robotics and Automation*, 14(4):617–622.

Gavaghan, K., Peterhans, M., Oliveira-Santos, T., and Weber, S. (2011). A portable image overlay projection device for computer-aided open liver surgery. *IEEE Transactions on Biomedical Engineering*, 58(6):1855–1864.

Hartley, R. (1997). In defense of the eight-point algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(6):580–593.

Hartley, R. and Zisserman, A. (2000). *Multiple View Geometry in Computer Vision*.

Lourakis, M. (Jul. 2004). levmar: Levenberg-marquardt nonlinear least squares algorithms in C/C++. [web page] http://www.ics.forth.gr/∼lourakis/levmar/. [Accessed on 31 Jan. 2005.].

Moreno, D. and Taubin, G. (2012). Simple, accurate, and robust projector-camera calibration. In *3D Imaging, Modeling, Processing, Visualization and Transmission (3DIMPVT), 2012 Second International Conference on*, pages 464–471.

Reinhart, G., Vogl, W., and Kresse, I. (2007). A projection-based user interface for industrial robots. In *Proceedings of the 2007 IEEE International Conference on Virtual Environments, Human-Computer Interfaces, and Measurement Systems, VECIMS 2007*, pages 67–71.

Salvi, J., Pags, J., and Batlle, J. (2004). Pattern codification strategies in structured light systems. *Pattern Recognition*, 37(4):827–849.

Shiu, Y. C. and Ahmad, S. (1989). Calibration of wrist-mounted robotic sensors by solving homogeneous transform equations of the form ax = xb. *IEEE Transactions on Robotics and Automation*, 5(1):16–29.

Tsai, R. Y. and Lenz, R. K. (1989). New technique for fully autonomous and efficient 3d robotics hand/eye calibration. *IEEE Transactions on Robotics and Automation*, 5(3):345–358.

Zhang, S. and Huang, P. (2006). Novel method for structured light system calibration. *Optical Engineering*, 45(8).

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

Zhuang, H., Roth, Z. S., and Sudhakar, R. (1994). Simultaneous robot/world and tool/flange calibration by solving homogeneous transformation equations of the form ax=yb. *IEEE T. Robotics and Automation*, 10(4):549–554.