ISPRS Journal of Photogrammetry and Remote Sensing 218 (2024) 645–662

Contents lists available at ScienceDirect

ISPRS Journal of Photogrammetry and Remote Sensing

journal homepage: www.elsevier.com/locate/isprsjprs

Vision-guided robot calibration using photogrammetric methods

Markus Ulrich [a,*,1], Carsten Steger [b,1], Florian Butsch [b,c], Maurice Liebe [a]

[a] Institute of Photogrammetry and Remote Sensing, Karlsruhe Institute of Technology, Englerstraße 7, 76131 Karlsruhe, Germany
[b] MVTec Software GmbH, Arnulfstraße 205, 80634 München, Germany
[c] School of Computation, Information and Technology, Technische Universität München, Arcisstraße 21, 80333 München, Germany

ARTICLE INFO

Keywords:
Robot calibration
Kinematic identification
Hand–eye calibration
Camera calibration
Industrial robotics
Pose accuracy
Gauss–Markov model
Gauss–Helmert model

ABSTRACT

We propose novel photogrammetry-based robot calibration methods for industrial robots that are guided by cameras or 3D sensors. Compared to state-of-the-art methods, our methods are capable of calibrating the robot kinematics, the hand–eye transformations, and, for camera-guided robots, the interior orientation of the camera simultaneously. Our approach uses a minimal parameterization of the robot kinematics and hand–eye transformations. Furthermore, it uses a camera model that is capable of handling a large range of complex lens distortions that can occur in cameras that are typically used in machine vision applications. To determine the model parameters, geometrically meaningful photogrammetric error measures are used. They are independent of the parameterization of the model and typically result in a higher accuracy. We apply a stochastic model for all parameters (observations and unknowns), which allows us to assess the precision and significance of the calibrated model parameters. To evaluate our methods, we propose novel procedures that are relevant in real-world applications and do not require ground truth values. Experiments on synthetic and real data show that our approach improves the absolute positioning accuracy of industrial robots significantly. By applying our approach to two different uncalibrated UR3e robots, one guided by a camera and one by a 3D sensor, we were able to reduce the RMS evaluation error by approximately 85% for each robot.

1. Introduction

It is well known that the repeatability (precision) with which a robot can position its end effector at a pose (rigid 3D transformation) programmed online, e.g., through a teach pendant, is much higher than the accuracy with which it can reach a pose that is obtained, for example, through machine vision and programmed offline. In the first kind of application, all that is required is to repeatably move the end effector to the same pose, although the actual pose in the workspace may differ slightly from the nominal pose in the robot controller. In the second kind of application, the pose is typically obtained with a sensor, e.g., a camera or a 3D sensor. Here, it is important that the exact pose measured by the machine vision system is reached. One prototypical example of this kind of application is a vision-guided assembly task. One of the major reasons for the suboptimal accuracy of a robot is errors in the kinematic parameters in its control software. These may occur, for example, because of manufacturing errors, assembly errors, and mechanical wear (Lou et al., 2009).

In this paper, we assume the robot is guided by an imaging sensor, i.e., a monocular camera or a 3D sensor, that is mounted on the robot's end effector. To achieve the required accuracy, it is essential that the entire robot system is calibrated. For the robot, this means that its kinematic parameters must be identified, i.e., calibrated. Furthermore, the pose of the imaging sensor with respect to the end effector or mechanical interface (the hand–eye pose) must be calibrated. Finally, for a camera-guided robot, the camera's interior orientation must be calibrated. For a robot that is guided by a 3D sensor, we assume that the 3D sensor returns a metric 3D point cloud, i.e., we assume the 3D sensor is calibrated. There are different technologies that are widely used to construct 3D sensors (Steger et al., 2018, Chapter 2.5). In this paper, we will assume that the 3D sensor is capable of returning a point cloud of the scene without the robot having to move the sensor across the scene. This includes stereo sensors, structured-light sensors, time-of-flight cameras, and sheet-of-light sensors in which the laser plane is actively moved by the static sensor, e.g., via a rotating mirror. Hence, we exclude sheet-of-light sensors in which the laser plane is static and must be moved by the robot.

Since we assume the robot is guided by the imaging sensor in the application, it is natural to use the imaging sensor also to calibrate

* Corresponding author.
E-mail addresses: markus.ulrich@kit.edu (M. Ulrich), carsten.steger@mvtec.com (C. Steger), florian.butsch@tum.de (F. Butsch), maurice.liebe@student.kit.edu (M. Liebe).
[1] Contributed equally to this work and share first authorship.

https://doi.org/10.1016/j.isprsjprs.2024.09.037
Received 24 June 2024; Received in revised form 27 September 2024; Accepted 27 September 2024
Available online 4 October 2024

the entire robot system. In this paper, we will refer to the calibration of the robot kinematics, the hand–eye pose, and, if used, the camera as the robot calibration. Using the existing imaging sensor for robot calibration has the advantage that the additional, often significant, costs of a sensor or other hardware for the sole purpose of kinematic calibration can be avoided. Instead, the imaging sensor that is already used in the application is used to calibrate the robot. Furthermore, using the existing imaging sensor has the advantage that the robot calibration can be performed frequently and with little downtime in the running process, if required.

The approach we propose is capable of calibrating the entire robot system end to end, i.e., it can simultaneously calibrate the robot kinematics, the hand–eye transformations, and, if used, the interior orientation of the camera. The calibration is performed by minimizing geometrically meaningful errors that are independent of the parameterization of the model. Our approach uses a camera model that can handle complex lens distortions. Our robot model includes a stochastic model for its model parameters. This allows the approach to assess the precision of the model parameters and to test whether a particular model parameter is significant, i.e., whether it has a statistically significant influence on the results of the calibrated model. This can be used to exclude insignificant parameters from the model. Finally, we propose novel evaluation procedures that are relevant in real-world applications and do not require ground truth values.

This paper is organized as follows. In Section 2, we discuss related work. The robot model is discussed in Section 3 and the camera model in Section 4. In Sections 5 and 6, the robot calibration algorithms and the stochastic model are discussed. Section 7 describes the evaluation of our approach. Section 8 concludes the paper.

2. Related work

A review of existing approaches has revealed that there are problems that occur frequently. To prevent tedious repetitions, we discuss the problems and list the approaches together with the problems rather than discussing the existing approaches individually.

Many kinematic calibration approaches are based on determining the pose or position of the robot's end effector using some kind of sensor. The sensors can be, for example, laser trackers (Chen et al., 2014; Li et al., 2016; Liu et al., 2018; Sun et al., 2020, 2021; Luo et al., 2023), coordinate measuring machines (Chen et al., 2001; Yang et al., 2014; Wu et al., 2015), monocular cameras (Motta et al., 2001; Rousseau et al., 2001; Boby and Klimchik, 2021; Balanji et al., 2022), stereo cameras (Liu et al., 2017; Zhang et al., 2017), portable photogrammetry systems (Filion et al., 2018), or 3D sensors (Yu and Xi, 2018; Peters and Knoll, 2024). Many kinds of these sensors often serve a single purpose in the application: the kinematic calibration of the robot. Imaging sensors, on the other hand, can be used for multiple purposes, e.g., machine vision tasks (Steger et al., 2018) or object detection and localization (Ulrich et al., 2012; Drost et al., 2010; Drost, 2016). In such applications, the imaging sensor is already part of the application. Therefore, using it for robot calibration is the preferred solution since it avoids the additional costs of a single-purpose sensor.

The pose or position of the end effector that is used for the kinematic calibration is either returned directly by the sensor or is determined via a suitable algorithm. For example, for monocular cameras, the perspective-n-point (PnP) algorithm is frequently used to determine the pose of a calibration object, i.e., the exterior orientation. The kinematic calibration is performed in these approaches by minimizing some kind of pose error. The parameters that describe a pose can always be divided into two groups of parameters: those that describe the position of the object and those that describe its orientation. Inherently, the parameters that describe the position are lengths. Hence, they are specified in units of meters, millimeters, inches, etc. On the other hand, the parameters that describe the orientation are inherently scalars. The pose-based approaches typically minimize the sum of the squared errors of the pose parameters or the sum of a rotation and a translation error. However, adding the squared or non-squared values of lengths and scalars does not result in a physically meaningful quantity and therefore does not produce a geometrically meaningful error. One consequence of optimizing this kind of pose error is that the results of the kinematic calibration will depend on the choice of units for the position parameters. For example, changing the units from meters to millimeters will re-weight the position errors by a factor of 1000 with respect to the orientation errors, i.e., it will increase the influence of position errors drastically. However, it is desirable that the kinematic calibration results do not depend on the unit that is chosen for the position parameters. Nevertheless, this requirement is ignored by almost all pose-based kinematic calibration approaches we have reviewed (Okamura and Park, 1996; Chen et al., 2001; Motta et al., 2001; Rousseau et al., 2001; Meng and Zhuang, 2007; He et al., 2010; Du and Zhang, 2013; Yang et al., 2014; Chen et al., 2014; Li et al., 2016; Liu et al., 2017; Zhang et al., 2017; Chang et al., 2018; Hayat et al., 2019; Madhusudanan et al., 2020; Sun et al., 2020, 2021; Boby and Klimchik, 2021; Balanji et al., 2022; Luo et al., 2023). Only one paper we have found explicitly describes and experimentally verifies that a change of units leads to different calibration results (Wu et al., 2015). In addition, Luo et al. (2022) acknowledge the problem theoretically and address it by splitting the optimization in each iteration into a part that optimizes the position error and another part that optimizes the orientation error. However, this is undesirable because it adds unnecessary complexity to the optimization. Another approach to handle this problem is to use only the position information returned by the pose sensor. However, if only one point is measured, not all kinematic parameters can be determined (Liu et al., 2018, 2023). A minimum of three linearly independent positions must be measured for each robot pose to be able to determine all kinematic parameters purely based on position measurements (Wu et al., 2015). This makes the calibration more cumbersome and time-consuming.

As described in Section 3, a kinematic model for a robot with r revolute and p prismatic joints with the minimum number of parameters has $4r + 2p + 6$ parameters. If the kinematic model uses more than the minimum number of parameters, i.e., if it is overparameterized, the Jacobian matrices that are used in the optimization of the kinematic parameters become ill-conditioned due to measurement errors or even singular. Ill-conditioned Jacobians typically lead to inaccurate calibration results, which is clearly undesirable. Several of the approaches we have reviewed do not handle the overparameterization explicitly (Chen et al., 2001; He et al., 2010; Liu et al., 2017; Boby and Klimchik, 2021; Boby, 2022; Balanji et al., 2022; Liu et al., 2023; Luo et al., 2023). Instead, they use the pseudoinverse to work around this problem, e.g., Chen et al. (2001), or they use unnecessarily complex optimization algorithms, e.g., Yang et al. (2014), Boby and Klimchik (2021), Boby (2022), and Liu et al. (2023). Kinematic models based on the product-of-exponentials (POE) formula, finite and instantaneous screw (FIS) theory, and unit dual quaternions are inherently overparameterized. The overparameterization must be removed explicitly to obtain minimal kinematic models. This can be done by adding suitable constraints to the kinematic model (Sun et al., 2020, 2021) or by using a suitable variant of a local POE parameterization (Yang et al., 2014; Chen et al., 2014; Wu et al., 2015; Li et al., 2016; Liu et al., 2018; Chang et al., 2018; Luo et al., 2022). Irrespective of how the overparameterization is removed, a significant complexity is introduced into the kinematic model and the optimization algorithm, which is undesirable. On the other hand, there are approaches that do not calibrate all possible $4r+2p+6$ parameters (Li et al., 2019; Ito et al., 2020; Peters and Knoll, 2024). This is also undesirable.

A kinematic model must be continuous with respect to its parameters. Not all kinematic models possess this property. For example, the Denavit–Hartenberg (DH) parameters (see Section 3) are not continuous if consecutive joint axes are parallel or nearly parallel. The following approaches use discontinuous kinematic models, which is

undesirable: Filion et al. (2018), Madhusudanan et al. (2020), Boby and Klimchik (2021), and Boby (2022).

Kinematic calibration approaches that use cameras sometimes use camera models that can only model a limited class of cameras. For example, some camera models do not model lens distortions (Zhuang and Roth, 1996, Chapter 7; Zhuang et al., 1995; Du and Zhang, 2013) or use a very simple lens distortion model (Motta et al., 2001). Other approaches assume that some camera parameters are fixed and known. For example, sometimes the principal point is assumed to lie at a known position (Meng and Zhuang, 2007). If a camera cannot be modeled adequately by the camera model, a suboptimal accuracy of the kinematic parameters may result.

Some approaches that use cameras require the interior orientation of the camera to be calibrated separately (Motta et al., 2001; Rousseau et al., 2001; Du and Zhang, 2013; Hayat et al., 2019; Boby, 2022). If the camera calibration requires the camera to be detached from the robot, long downtimes are caused. In addition, after the camera has been reattached to the robot, a robot calibration is necessary. This extra time and complexity is undesirable.

The following two approaches that use cameras for robot calibration are the closest to our approach in the sense that they are based not on 3D pose or position errors but on errors that are defined in the image coordinate system.

The approach by Zhuang and Roth (1996, Chapter 7) and Zhuang et al. (1995) uses a monocular camera mounted on the end effector. In addition to the problems that were described above, it algebraically transforms the 2D position error that the approach could use in principle into an error that is not a geometrically meaningful.

The approach by Boby (2022) also uses a monocular camera mounted on the end effector. Like our approach, it optimizes a photogrammetric error based on 2D image coordinates, which is a geometric error. In contrast to our approach, it does not optimize the camera parameters, i.e., the camera is assumed to be pre-calibrated. The approach uses the DH parameters, but the singularity of this parameterization is not handled. Furthermore, the model is overparameterized: it has 36 instead of 30 parameters for a robot with six revolute joints. The optimization uses a dampening factor in the Gauss–Newton optimization to ensure "slow but stable convergence" (Boby, 2022, Section IV.A), which is undesirable. We assume that this was necessary because of the overparameterization and the failure to handle the singularity of the DH parameterization.

We were only able to find two approaches that use the 3D data obtained from 3D sensors for kinematic calibration.

The approach by Yu and Xi (2018) uses a self-constructed sheet-of-light sensor mounted on the robot's end effector. The laser line is displaced using a rotating mirror, enabling the sensor to reconstruct a 3D point cloud without moving the robot. The calibration object is made up of four non-collinear spheres. One disadvantage of this approach is that the positions of the spheres must be measured very accurately in advance using a laser tracker. The costs of a laser tracker are significant, which makes this approach less desirable. For the purpose of calibration, spheres are fitted to the 3D point cloud returned by the sensor to compute the centers of the spheres. The kinematic parameters are determined by optimizing the Euclidean position error of the spheres. One potential drawback of this approach is that the calibration is performed based only on a relatively small number of measurements per scan (the centers of the spheres). Thus, it might be prone to errors or outliers in the fitted sphere centers.

An approach that calibrates the hand–eye and kinematic parameters simultaneously using 3D sensors, such as single beam LIDAR sensors, sheet-of-light scanners, and depth cameras, is proposed by Peters and Knoll (2024). It uses a simultaneous localization and mapping (SLAM) approach based on the iterative closest point (ICP) algorithm with arbitrary objects to perform the calibration. To achieve the required robustness and accuracy, more than 20 scans of the object are necessary.

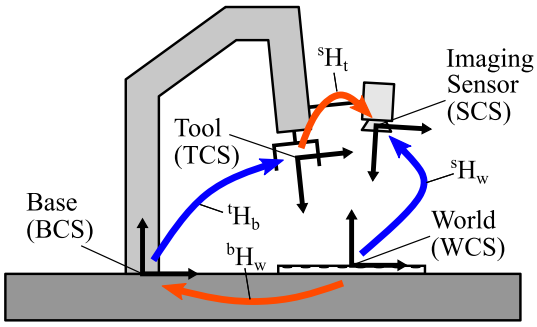


Fig. 1. Relevant 3D coordinate systems as well as transformations that change (blue) and that do not change (orange) during robot movements. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)
Source: Adapted from Ulrich and Hillemann (2021, 2024).

The approach is unable to calibrate all kinematic and hand–eye parameters. For example, for a robot with seven revolute joints, it is only capable of calibrating 28 of the 34 parameters that a minimal model includes. Another drawback of this approach is the enormous amount of time (several hours) that is required to perform the calibration, which is undesirable since it leads to unnecessarily long downtimes.

Finally, we note that none of the approaches discussed above include a stochastic model of the robot parameters.

3. Robot model

3.1. Coordinate systems and transformations

We will denote the 4×4 homogeneous transformation matrix that represents a rigid 3D transformation of points from coordinate system s1 to s2 by $^{s2}\mathbf{H}_{s1}$. Thus, the transformation of a homogeneous point $\boldsymbol{p}_{s1}$ in coordinate system s1 to the point $\boldsymbol{p}_{s2}$ in coordinate system s2 is given by

$$\boldsymbol{p}_{s2} = {}^{s2}\mathbf{H}_{s1}\boldsymbol{p}_{s1} = \begin{pmatrix} \mathbf{R} & \boldsymbol{t} \\ \mathbf{0}^{\top} & 1 \end{pmatrix} \boldsymbol{p}_{s1} \ , \tag{1}$$

where $\boldsymbol{t} = (t_x, t_y, t_z)^{\top}$ is a translation vector and $\mathbf{R}$ a rotation matrix that is parameterized via the Euler angles α, β, and γ by $\mathbf{R} = \mathbf{R}_x(\alpha)\mathbf{R}_y(\beta)\mathbf{R}_z(\gamma)$ (Kuipers, 1999, Chapters 4.3–4.4).

We will see in Section 6 that the robot calibration algorithm that uses 3D sensors must transform planes between coordinate systems instead of points. We will describe planes by a homogeneous vector $\boldsymbol{\pi} = (\pi_x, \pi_y, \pi_z, \pi_w)^{\top}$, where the normal vector $\boldsymbol{n} = (\pi_x, \pi_y, \pi_z)^{\top}$ of the plane is normalized to have length 1, i.e., $\|\boldsymbol{n}\|_2 = 1$. The signed distance d of a point $\boldsymbol{p}$ from the plane $\boldsymbol{\pi}$ is given by $d = \boldsymbol{\pi}^{\top}\boldsymbol{p}$. Hence, $\boldsymbol{p}$ lies on $\boldsymbol{\pi}$ if $\boldsymbol{\pi}^{\top}\boldsymbol{p} = 0$.

The vector $\boldsymbol{\pi}$ is transformed by the inverse transpose of a rigid transformation (1) (Hartley and Zisserman, 2003, Chapter 3.2.1). We will denote the inverse transpose of a homogeneous transformation matrix by $\mathbf{G}$. Hence, we have

$$\boldsymbol{\pi}_{s2} = {}^{s2}\mathbf{G}_{s1}\boldsymbol{\pi}_{s1} = {}^{s2}\mathbf{H}_{s1}^{-\top}\boldsymbol{\pi}_{s1} = \begin{pmatrix} \mathbf{R} & \mathbf{0} \\ -\boldsymbol{t}^{\top}\mathbf{R} & 1 \end{pmatrix} \boldsymbol{\pi}_{s1} \ . \tag{2}$$

Note that this transformation preserves the normalization of the normal vector of a plane.

Fig. 1 displays the coordinate systems that are relevant for robot calibration: the world (w, WCS), imaging sensor (s, SCS), robot tool (t, TCS), and robot base (b, BCS) coordinate systems. Furthermore, the relevant poses are shown: $^{t}\mathbf{H}_{b}$ (robot pose) and $^{s}\mathbf{H}_{w}$ (imaging sensor pose), which both vary when the robot is moved, as well as $^{t}\mathbf{H}_{s}$ (hand–eye pose) and $^{b}\mathbf{H}_{w}$ (world–base pose), which do not change during robot movements.

The imaging sensor pose, the hand–eye pose, and the world–base pose are represented by a translation vector $\boldsymbol{t}$ and a rotation matrix $\mathbf{R}$ as indicated in (1).

3.2. Kinematic model

The robot pose ${}^{\mathrm{b}}\mathbf{H}_{\mathrm{t}}$ is given by the forward kinematics of the robot. Many different parameterizations have been proposed to model the forward kinematics. It is essential that the kinematic model is complete, minimal, and continuous (Schröer et al., 1997). An incomplete model would not be able to model all possible robot poses. Non-continuity or non-minimality of the model will cause numerical problems in optimization algorithms that are used to determine the robot parameters and will lead to inaccuracies. It has been shown that, no matter what kinematic model is used, a robot with r revolute and p prismatic joints has $4r+2p+6$ independent parameters (Schröer et al., 1997; Chen et al., 2001; Li et al., 2014; Yang et al., 2014; Chen et al., 2014; Wu et al., 2019; Sun et al., 2020). This includes the parameters of the hand–eye and world–base poses.

One commonly used kinematic model is given by the Denavit–Hartenberg (DH) parameters (Denavit and Hartenberg, 1955; Zhuang and Roth, 1996, Chapter 3.III.A). Various conventions are used for these parameters, which differ in the way the kinematic parameters are assigned to robot joints. The conventions are called the original, distal, and proximal variants by Lipkin (2005). We use the original DH variant in our approach. The DH convention requires the assignment of a coordinate system to each joint in a specific manner, as described, for example, by Zhuang and Roth (1996, Chapter 3.III.A). Once the coordinate systems have been assigned, the transformation from joint coordinate system i to joint coordinate system $i-1$ is given by:

$$ {}^{i-1}\mathbf{H}_i = \mathbf{R}_z(\theta_i)\,\mathbf{T}_z(d_i)\,\mathbf{T}_x(a_i)\,\mathbf{R}_x(\alpha_i) \ . \tag{3} $$

Here, θ_i, d_i, a_i, and α_i are the joint parameters and $\mathbf{R}_\xi(\phi)$ denotes a rotation around the axis ξ by the angle ϕ, while $\mathbf{T}_\xi(\delta)$ denotes a translation along the axis ξ by the distance δ. For revolute joints, the joint is rotated around the z axis by the angle θ_i, while for prismatic joints, it is moved along the z axis by the distance d_i.

It is well known that the DH parameters are complete and minimal, but not continuous if consecutive joint axes are parallel or nearly parallel. To prevent this problem, Hayati and Mirmirani modified the transformation for nominally parallel joint axes as follows (Hayati and Mirmirani, 1985; Zhuang and Roth, 1996, Chapter 3.III.C):

$$ {}^{i-1}\mathbf{H}_i = \mathbf{R}_z(\theta_i)\,\mathbf{T}_x(a_i)\,\mathbf{R}_x(\alpha_i)\,\mathbf{R}_y(\beta_i) \ . \tag{4} $$

Hence, the parameter β_i replaces the parameter d_i, which is assumed to be 0.

The Hayati–Mirmirani parameterization is continuous for nearly parallel joint axes, but discontinuous for nearly perpendicular joint axes. Both conventions can be combined by using the following transformation (Veitschegger and Wu, 1986):

$$ {}^{i-1}\mathbf{H}_i = \mathbf{R}_z(\theta_i)\,\mathbf{T}_z(d_i)\,\mathbf{T}_x(a_i)\,\mathbf{R}_x(\alpha_i)\,\mathbf{R}_y(\beta_i) \ . \tag{5} $$

This is an overparameterization, which we will remove by not optimizing d_i for nominally parallel axes and not optimizing β_i for nominally perpendicular axes in the calibration algorithms that we present in Section 5. The resulting model is complete, minimal, and continuous. It is sometimes called the modified DH model. However, this term is also frequently used to denote the aforementioned variants of the DH model. To avoid ambiguities, we will denote this model as the DHHM model. Several other approaches for kinematic calibration also use this model, e.g., Motta et al. (2001), Rousseau et al. (2001), Du and Zhang (2013), Liu et al. (2017), Yu and Xi (2018), Li et al. (2019), and Ito et al. (2020).

Based on the DHHM transformation (5), the forward kinematics of a robot with n_{a} joints is given by:

$$ {}^{\mathrm{b}}\mathbf{H}_{\mathrm{t}} = {}^{0}\mathbf{H}_1 \cdot {}^{1}\mathbf{H}_2 \cdots {}^{n_{\mathrm{a}}-1}\mathbf{H}_{n_{\mathrm{a}}} \ . \tag{6} $$

As shown in Fig. 1, the robot calibration algorithms presented in Section 5 use the inverse of this transformation:

$$ {}^{\mathrm{t}}\mathbf{H}_{\mathrm{b}} = {}^{n_{\mathrm{a}}}\mathbf{H}_{n_{\mathrm{a}}-1} \cdots {}^{2}\mathbf{H}_1 \cdot {}^{1}\mathbf{H}_0 \ , \tag{7} $$

in which

$$ {}^{i}\mathbf{H}_{i-1} = \mathbf{R}_y(-\beta_i)\,\mathbf{R}_x(-\alpha_i)\,\mathbf{T}_x(-a_i)\,\mathbf{T}_z(-d_i)\,\mathbf{R}_z(-\theta_i) \ . \tag{8} $$

There are other parameterizations that are complete and continuous, such as the product-of-exponentials (POE) formula (Murray et al., 1994, Chapter 3.2.2; Lynch and Park, 2017, Chapter 4.1), finite and instantaneous screw (FIS) theory (Sun et al., 2020, 2021), and unit dual quaternions (Luo et al., 2023) that were mentioned in Section 2. These models are inherently overparameterized and require a significantly higher complexity in the model or optimization algorithm to handle or remove the overparameterization (Yang et al., 2014; Chen et al., 2014; Wu et al., 2015; Li et al., 2016; Liu et al., 2018; Chang et al., 2018; Luo et al., 2022; Sun et al., 2020, 2021). Furthermore, the experiments by Chen et al. (2014) and Li et al. (2016) show that there is no advantage in using a non-redundant POE model over a DHHM model and between different minimally parameterized POE models in terms of accuracy or convergence speed. Therefore, we see no advantage in using one of these kinematic models for our robot calibration algorithm.

To calibrate the robot kinematics, we model the kinematic parameters of joint i as offsets from their nominal values:

$$ \theta_i = \tilde{\theta}_i + \Delta_{\theta_i} \tag{9} $$

$$ d_i = \tilde{d}_i + \Delta_{d_i} \tag{10} $$

$$ a_i = \tilde{a}_i + \Delta_{a_i} \tag{11} $$

$$ \alpha_i = \tilde{\alpha}_i + \Delta_{\alpha_i} \tag{12} $$

$$ \beta_i = \tilde{\beta}_i + \Delta_{\beta_i} \ . \tag{13} $$

Here, $\tilde{\cdot}$ denotes the fixed nominal kinematic parameters, while Δ indicates the unknown offsets.

The robots we use in the evaluation in Section 7 use a kinematic model in their controller that is based on the original DH parameters with offsets, as in the above equations, i.e., it does not contain the parameter β_i. Therefore, to reconfigure the robot controller to use our calibration results, we must convert our calibrated DHHM parameters to DH parameters. The easiest way to do this is to convert our DHHM parameters to POE parameters using a straightforward extension of the corresponding algorithm for DH parameters given by Murray et al. (1994, Chapter 3.2.3); Lynch and Park (2017, Appendix C.5), and Wu et al. (2017). The resulting POE parameters can be converted to DH parameters by the algorithm given by Wu et al. (2017) and can then be converted to nominal values plus offsets. The advantage of this approach would be that the robot controller performs the necessary calculations to determine the inverse kinematics that are required to move the robot to a specific pose.

Alternatively, we can compute the inverse kinematics in our own software. In our approach, we use the inverse kinematics algorithm given by Villalobos et al. (2021a,b) with the nominal kinematic parameters to obtain initial joint angles. The joint angles are then refined using a nonlinear optimization algorithm with the calibrated DHHM parameters (i.e., nominal values plus calibrated offsets). The advantage of this approach is that we have full control over the kinematic model so that we are more flexible when adapting the kinematic model in the future, e.g., by modeling additional error sources such as the pitch error of revolute joints (Ma et al., 2018). Furthermore, there are robots that do not allow to change the kinematics data in the robot controller and, hence, implementing the inverse kinematics is the only way to use the calibrated parameters.

3.3. Model degeneracies

The model we have described so far is overparameterized. It has $4r + 2p + 12$ parameters: $4r + 2p$ parameters for the kinematics, six for the hand–eye pose, and six for the world–base pose. Therefore, six parameters are redundant and must be excluded from the optimization. Tests with the approach by Pashkevich (2001) showed that the parameters Δ_{θ_1} and Δ_{d_1} form a semi-identifiable group with the parameters of the world–base pose and that all parameters $\Delta_{\theta_{n_a}}$, $\Delta_{d_{n_a}}$, $\Delta_{\alpha_{n_a}}$, and $\Delta_{a_{n_a}}$ form a semi-identifiable group with the parameters of the hand–eye pose. Hence, each of these parameter groups contain parameters that are functionally dependent on each other, i.e., their respective values cannot be identified individually. Therefore, like Yu and Xi (2018) and Rousseau et al. (2001), we leave the values of these parameters fixed by excluding them from the optimization. Any error that might be present in these parameters will be compensated by the hand–eye and world–base poses.

4. Camera model

In this section, we briefly describe the camera model we use, which is the perspective camera model presented by Steger et al. (2018, Chapter 3.9.1): A 3D point $\boldsymbol{p}_\mathrm{w}$ in the WCS is transformed to a point $\boldsymbol{p}_\mathrm{s}$ in the SCS by

$$\boldsymbol{p}_\mathrm{s} = {}^\mathrm{s}\mathbf{H}_\mathrm{w}\boldsymbol{p}_\mathrm{w} \ . \tag{14}$$

As described in Section 3, ${}^\mathrm{s}\mathbf{H}_\mathrm{w}$ is parameterized by the translation vector $\boldsymbol{t} = (t_\mathrm{x}, t_\mathrm{y}, t_\mathrm{z})^\top$ and the Euler angles α, β, and γ. The parameters (t_x, t_y, t_z, α, β, γ) define the exterior orientation of the camera.

Subsequently, the point $\boldsymbol{p}_\mathrm{s} = (x_\mathrm{s}, y_\mathrm{s}, z_\mathrm{s})^\top$ is projected onto the image plane by

$$\begin{pmatrix} x_\mathrm{u} \\ y_\mathrm{u} \end{pmatrix} = \frac{c}{z_\mathrm{s}} \begin{pmatrix} x_\mathrm{s} \\ y_\mathrm{s} \end{pmatrix} \ , \tag{15}$$

where c represents the principal distance of the lens. Next, the undistorted point $(x_\mathrm{u}, y_\mathrm{u})^\top$ is transformed into the distorted point $(x_\mathrm{d}, y_\mathrm{d})^\top$ by applying either the division model (Lenz and Fritsch, 1990)

$$\begin{pmatrix} x_\mathrm{u} \\ y_\mathrm{u} \end{pmatrix} = \frac{1}{1 + \kappa r_\mathrm{d}^2} \begin{pmatrix} x_\mathrm{d} \\ y_\mathrm{d} \end{pmatrix} \ , \tag{16}$$

which models radial distortions via the parameter κ, or by applying the polynomial model (Brown, 1971)

$$\begin{pmatrix} x_\mathrm{u} \\ y_\mathrm{u} \end{pmatrix} = \begin{pmatrix} x_\mathrm{d}(1 + K_1 r_\mathrm{d}^2 + K_2 r_\mathrm{d}^4 + K_3 r_\mathrm{d}^6) \\ \quad + (P_1(r_\mathrm{d}^2 + 2x_\mathrm{d}^2) + 2P_2 x_\mathrm{d} y_\mathrm{d}) \\ y_\mathrm{d}(1 + K_1 r_\mathrm{d}^2 + K_2 r_\mathrm{d}^4 + K_3 r_\mathrm{d}^6) \\ \quad + (2P_1 x_\mathrm{d} y_\mathrm{d} + P_2(r_\mathrm{d}^2 + 2y_\mathrm{d}^2)) \end{pmatrix} \ , \tag{17}$$

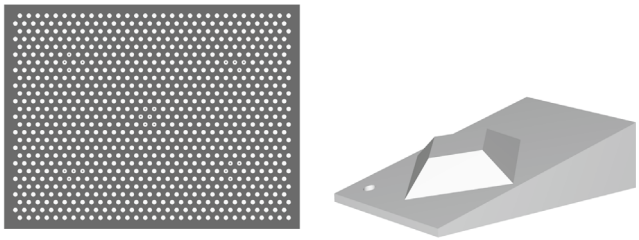
which represents radial distortions via parameters K_1, K_2, and K_3 as well as decentering distortions via P_1 and P_2. Here, $r_\mathrm{d}^2 = x_\mathrm{d}^2 + y_\mathrm{d}^2$. While the division model can be analytically inverted, the polynomial model needs a numerical root finding algorithm for inversion (Steger et al., 2018, Chapter 3.9.1.3).

In the last step, the distorted point $(x_\mathrm{d}, y_\mathrm{d})^\top$ is transformed to the image coordinate system:

$$\boldsymbol{p}_\mathrm{i} = \begin{pmatrix} x_\mathrm{i} \\ y_\mathrm{i} \end{pmatrix} = \begin{pmatrix} x_\mathrm{d}/s_x + c_x \\ y_\mathrm{d}/s_y + c_y \end{pmatrix} \ . \tag{18}$$

Here, $(c_x, c_y)^\top$ is the principal point and s_x and s_y indicate the pixel pitches on the sensor (x refers to the horizontal and y to the vertical image axis).

Depending on the chosen distortion model, the interior orientation $\boldsymbol{i}$ of the camera is described by the six parameters $\boldsymbol{i} = (c, \kappa, s_x, s_y, c_x, c_y)^\top$ for the division model or by the ten parameters $\boldsymbol{i} = (c, K_1, K_2, K_3, P_1, P_2, s_x, s_y, c_x, c_y)^\top$ for the polynomial model.



(a) (b)

Fig. 2. (a) Planar calibration object with 873 control points used for the camera-based robot calibration in Section 5 and for its evaluation in Section 7. (b) CAD model of the 3D calibration object used for the 3D-sensor-based robot calibration in Section 6 and for its evaluation in Section 7.

5. Robot calibration using cameras

5.1. Overview

To calibrate the robot using a camera, a calibration object is placed at a fixed position within the workspace of the robot. For our experiments, we used the planar HALCON calibration object (MVTec Software GmbH, 2024, Chapter 3.2) with known hexagonally arranged circular control points (see Fig. 2(a)). Then, the robot tool is moved to different poses and an image of the calibration object is acquired at each pose. In each calibration image, the image coordinates of the calibration points are extracted. They represent the observations during robot calibration. The unknowns are the kinematic parameters of the robot joints, the parameters of the hand–eye pose that describes the pose of the camera with respect to the robot tool, and the parameters of the world–base pose that describes the pose of the calibration object with respect to the robot base. If the camera was not pre-calibrated, the parameters of the interior orientation are additional unknowns. To obtain the best accuracy, it is advantageous to cover the working range of each robot joint as completely as possible. Because this is difficult to achieve with the calibration object at a single pose, our proposed calibration model supports the acquisition of calibration images of the calibration object in multiple poses.

5.2. Calibration model

Let n_c be the number of different calibration object poses and $n_{\mathrm{r},l}$ the number of robot poses (and hence the number of acquired calibration images) at calibration object pose l ($l = 1, \ldots, n_\mathrm{c}$). Furthermore, let the 3D world points (given in the WCS) of the calibration object be $\boldsymbol{p}_k$ ($k = 1, \ldots, n_\mathrm{w}$) and their 2D projections into the image at calibration object pose l and robot pose j be $\boldsymbol{p}_{l,j,k}$ ($j = 1, \ldots, n_{\mathrm{r},l}$). Then, the projection of a 3D point into the image is described by:

$$\boldsymbol{p}_{l,j,k} = \boldsymbol{\pi}({}^\mathrm{s}\mathbf{H}_\mathrm{t}\,{}^\mathrm{t}\mathbf{H}_{\mathrm{b},l,j}\,{}^\mathrm{b}\mathbf{H}_{\mathrm{w},l}\boldsymbol{p}_k, \boldsymbol{i}) \ , \tag{19}$$

where ${}^\mathrm{s}\mathbf{H}_\mathrm{t}$ represents the hand–eye pose, ${}^\mathrm{t}\mathbf{H}_{\mathrm{b},l,j}$ represents the robot pose that is used to acquire calibration image j of the calibration object at calibration object pose l, and ${}^\mathrm{b}\mathbf{H}_{\mathrm{w},l}$ represents the pose of the calibration object at pose l with respect to the robot base (cf. Section 3). The function $\boldsymbol{\pi}(\boldsymbol{p}_\mathrm{s}, \boldsymbol{i})$ is the projection of the point $\boldsymbol{p}_\mathrm{s}$, which is given in the SCS, into the image by successive computation of (15), the inverse of either (16) or (17), and (18) using the parameters of the interior orientation $\boldsymbol{i}$. The matrix ${}^\mathrm{t}\mathbf{H}_{\mathrm{b},l,j}$ is obtained by applying the forward kinematics of (7) with the joint parameters given in (9)–(13).

We denote the vector that contains the six transformation parameters of the unknown hand–eye pose ${}^\mathrm{s}\mathbf{H}_\mathrm{t}$ by $\boldsymbol{e}_\mathrm{s}$, the vector of the unknown calibration object poses ${}^\mathrm{b}\mathbf{H}_{\mathrm{w},l}$ by $\boldsymbol{e}_{\mathrm{b},l}$, and the set of unknown kinematic parameters, i.e., the offsets for the DHHM parameters, of joint i by $\boldsymbol{d}_i$. For nominally perpendicular axes, $\boldsymbol{d}_i = (\Delta_{\theta_i}, \Delta_{d_i}, \Delta_{a_i}, \Delta_{\alpha_i})^\top$, while for nominally parallel axes $\boldsymbol{d}_i = (\Delta_{\theta_i}, \Delta_{a_i}, \Delta_{\alpha_i}, \Delta_{\beta_i})^\top$.

5.3. Parameter estimation in the Gauss–Markov model

Minimizing the reprojection error of 3D points in the image is a well-established method in photogrammetry and computer vision (Förstner and Wrobel, 2016, Chapter 15). Compared to the minimization of algebraic errors, which do not represent a geometric quantity, it results in a higher accuracy in most applications. Furthermore, minimizing the reprojection error has the advantage of directly minimizing the errors of the observations, i.e., the coordinates of the image points, instead of errors of quantities that are derived from the observations, like poses, for example. This facilitates the correct modeling of the statistical properties.

To minimize the photogrammetric reprojection error, we apply the parameter estimation in the Gauss–Markov model (Förstner and Wrobel, 2016, Chapter 4.4; Niemeier, 2008, Chapter 4.3), which is a standard tool in photogrammetry and distinguishes between the functional model and the stochastic model. In the functional model, the observations $\boldsymbol{l}$ are described as an explicit function $\boldsymbol{l} = \boldsymbol{f}(\boldsymbol{x})$ of the unknowns $\boldsymbol{x}$. In our calibration model, $\boldsymbol{f}$ corresponds to (19). In the stochastic model, the observations and the unknowns are modeled as random variables with uncertainties, where the uncertainties are represented by covariances.

The vector of observations $\boldsymbol{l}$ contains the measured image points

$$\boldsymbol{l} = (\boldsymbol{p}_{1,1,1}^\top, \dots, \boldsymbol{p}_{1,1,n_\mathrm{w}}^\top, \dots, \boldsymbol{p}_{1,n_\mathrm{r},1}^\top, \dots, \boldsymbol{p}_{n_\mathrm{c},n_\mathrm{r},n_\mathrm{w}}^\top)^\top \ . \tag{20}$$

If a 3D point is not visible in a given image, the corresponding entry in $\boldsymbol{l}$ is simply omitted. If n_i is the total number of measured image points over all calibration images, the number of observations n_l equals $2n_\mathrm{i}$. Each observed image point yields two equations in the form of (19), thus yielding $2n_\mathrm{i}$ equations in total.

The vector of unknowns $\boldsymbol{x}$ is initially built as

$$\boldsymbol{x} = (\boldsymbol{e}_\mathrm{s}^\top, \boldsymbol{e}_{\mathrm{b},1}^\top, \dots, \boldsymbol{e}_{\mathrm{b},n_\mathrm{c}}^\top, \boldsymbol{d}_1^\top, \dots, \boldsymbol{d}_{n_\mathrm{a}}^\top, \boldsymbol{i}^\top)^\top \ , \tag{21}$$

containing the six parameters of the hand–eye pose, the $6n_\mathrm{c}$ parameters of the n_c calibration object poses, the $4n_\mathrm{a}$ unknown kinematic parameters for each of the n_a joints, and the parameters of the interior orientation. If the camera was pre-calibrated, the interior orientation can be excluded from the robot calibration by omitting $\boldsymbol{i}$ in $\boldsymbol{x}$. Note that because of the model degeneracies described in Section 3.3, not all kinematic parameters can be estimated simultaneously together with the parameters of the hand–eye pose and the calibration object poses (which correspond to the world–base pose). Therefore, we exclude Δ_{θ_1} and Δ_{d_1} of the parameters $\boldsymbol{d}_1$ of the first joint and all parameters $\boldsymbol{d}_{n_\mathrm{a}}$ of the last joint from the parameter estimation by deleting these parameters from $\boldsymbol{x}$. Let the final number of unknowns in $\boldsymbol{x}$ be n_x.

The statistical properties of the observation process are specified in the stochastic model. In most applications, it is appropriate to assume that the observed image points are uncorrelated and have identical precision. In these cases, we set the $n_l \times n_l$ weight coefficient matrix (i.e., the initial covariance matrix) of the observations to the identity matrix: $\mathbf{Q}_{ll} = \mathbf{I}$. From $\mathbf{Q}_{ll}$, we can compute the weight matrix $\mathbf{P}_{ll} = \mathbf{Q}_{ll}^{-1}$. If more detailed knowledge about the image points' accuracy is known, the elements in $\mathbf{P}_{ll}$ can be set to appropriate weights or the elements in $\mathbf{Q}_{ll}$ to appropriate covariances.

For linearization, we compute the $n_l \times n_\mathrm{x}$ Jacobian $\mathbf{A}$ of $\boldsymbol{f}$ at the initial values of the unknowns $\boldsymbol{x}^{(0)}$ (cf. Section 5.4):

$$\mathbf{A} = \left.\frac{\partial \boldsymbol{f}(\boldsymbol{x})}{\partial \boldsymbol{x}}\right|_{\boldsymbol{x}=\boldsymbol{x}^{(0)}} \ . \tag{22}$$

With the residuals $\Delta\boldsymbol{l} = \boldsymbol{l} - \boldsymbol{f}(\boldsymbol{x}^{(0)})$ and the weight matrix $\mathbf{P}_{ll} = \mathbf{Q}_{ll}^{-1}$, the corrections for the unknowns are calculated by solving

$$\mathbf{A}^\top \mathbf{P}_{ll} \mathbf{A} \Delta\hat{\boldsymbol{x}} = \mathbf{A}^\top \mathbf{P}_{ll} \Delta\boldsymbol{l} \tag{23}$$

for $\Delta\hat{\boldsymbol{x}}$, for example, by Cholesky decomposition. Here, the hat $\hat{}$ denotes a fitted or estimated value. Finally, the corrections are applied to the unknowns by

$$\hat{\boldsymbol{x}}^{(1)} = \boldsymbol{x}^{(0)} + \Delta\hat{\boldsymbol{x}} \ . \tag{24}$$

Eqs. (22)–(24) are repeatedly applied until convergence. This corresponds to the minimization of the reprojection error of the 3D points, i.e., the differences between the 3D points projected into the image by (19) and the corresponding 2D points that are observed in the image. After convergence, the covariance matrix of the original observations is obtained by

$$\mathbf{C}_{ll} = \hat{\sigma}_0^2 \mathbf{Q}_{ll} \ , \tag{25}$$

with the variance factor

$$\hat{\sigma}_0^2 = \hat{\boldsymbol{v}}^\top \mathbf{P}_{ll} \hat{\boldsymbol{v}} / r \ , \tag{26}$$

the residuals $\hat{\boldsymbol{v}} = \mathbf{A}\Delta\hat{\boldsymbol{x}} - \Delta\boldsymbol{l}$, and the redundancy $r = n_l - n_\mathrm{x}$. The corrected observations are obtained by $\hat{\boldsymbol{l}} = \boldsymbol{l} + \hat{\boldsymbol{v}}$. The covariance matrix of the estimated unknowns is obtained by variance propagation:

$$\mathbf{C}_{\hat{\boldsymbol{x}}\hat{\boldsymbol{x}}} = \hat{\sigma}_0^2 (\mathbf{A}^\top \mathbf{P}_{ll} \mathbf{A})^{-1} \tag{27}$$

Details about the Gauss–Markov model are provided by Förstner and Wrobel (2016, Chapter 4.4.1) and Niemeier (2008, Chapter 4.3).

One advantage of the parameter estimation in the Gauss–Markov model is that information about the precision of the observations and the unknowns are obtained. While $\mathbf{C}_{ll}$ contains the covariances of the observations, $\mathbf{C}_{\hat{\boldsymbol{x}}\hat{\boldsymbol{x}}}$ contains the covariances of the unknowns. The latter can be used not only for assessing the precision of the estimated parameters but also for investigating correlations between them. It also enables parameter testing, which can yield valuable information about the significance of the model parameters (e.g., lens distortion parameters) or about the robot, such as the relevant error sources of each individual joint.

5.4. Initialization of the unknowns

Because of the nonlinearity of the optimization problem, we must provide starting values for the unknowns. We propose the following initialization for $\boldsymbol{x}^{(0)}$:

- The nominal DH parameters $\tilde{d}_i$, $\tilde{a}_i$, and $\tilde{\alpha}_i$ are obtained from the specification or data sheet of the robot. All values of $\tilde{\beta}_i$ are set to 0. The nominal joint angles $\tilde{\theta}_{i,l,j}$ for calibration object pose l and robot pose j are read out from the robot controller.
- For the unknown robot parameters, all offset values are set to 0.
- Initial values for the hand–eye pose $\boldsymbol{e}_\mathrm{s}$ and for $\boldsymbol{e}_{\mathrm{b},l}$ are obtained from a linear approach (e.g., Daniilidis (1999) for anthropomorphic robots or Ulrich and Steger (2016) for SCARA robots), optionally followed by a non-linear approach for more accurate initial values (e.g., Steger et al. (2018, Chapter 3.13.5) or Ulrich and Hillemann (2024)). If the chosen implementation of the hand–eye calibration does not support multiple calibration object poses, the hand–eye calibration must be performed for each calibration object pose separately to obtain initial values for the calibration object poses. In this case, the hand–eye pose can be computed by averaging the individual results, for example.
- Initial values for the interior camera parameters are taken from the camera (s_x and s_y) and the lens (c) data sheets. The principal point $(c_x, c_y)^\top$ is set to the center of the image and the distortion coefficients are set to 0.

5.5. Geometric degeneracy

During our experiments on real data (cf. Section 7.4), we first ran into the problem that the system of normal Eqs. (23) became almost singular. Based on typical camera constellations in close-range photogrammetric bundle-adjustment (Luhmann et al., 2020, Chapter 4.4), we had chosen the robot poses such that the optical centers of the cameras were located on the surface of a sphere. Furthermore, their optical axes intersected at the sphere's center, which coincided with the center of the calibration object. It turned out that this set of

robot poses is geometrically degenerate, i.e., it failed to uniquely define the estimated parameters during the robot calibration. To avoid this degeneracy, we chose the robot poses for the experiments in Section 7.4 randomly.

5.6. Correction of circular mark centers

In our experiments in Section 7.4, we use a planar calibration object with known circular control points (see Fig. 2(a)). The image coordinates of the control points are extracted by computing the centers of the ellipses that were fitted to their subpixel-accurate image edges. It is well known that this causes a slight positional bias in the points (Steger, 2017, Section 10). For the majority of applications, the distortion is negligible. Nonetheless, we eliminate the bias by using the method described by Steger (2017, Section 10).

6. Robot calibration using 3D sensors

6.1. Overview

To perform the robot calibration using a 3D sensor, a 3D calibration object is placed at a fixed position within the workspace of the robot. For our experiments, we used the HALCON sheet-of-light calibration object (MVTec Software GmbH, 2024, Chapter 6.3) (see Fig. 2(b)). It consists of a frustum on top of a wedge-shaped body. We only use the upper part (tilted plane and frustum) for the calibration because the sides are difficult to detect and measure robustly with a 3D sensor. The geometry of the calibration object is described by a triangulated CAD model to ensure that each polygon (i.e., triangle) can be represented by one plane. Our method is flexible with regard to the shape of the calibration object. The only requirement is that the pose of the calibration object can be determined without ambiguities from a 3D scan of the object, i.e., the object should not possess rotational symmetries or too little geometric structure to uniquely determine its pose.

To acquire the data for the calibration, the robot tool is moved to different poses and a 3D scan of the calibration object is acquired at each pose. The poses should be chosen such that the calibration object covers a substantial part of the 3D scan. In each scan, the 3D scene points that correspond to the surface of the calibration object represent the observations during the robot calibration. The optimization is based on transforming the planes of the CAD model using the world–base pose, the robot pose, and the hand–eye pose to the SCS and minimizing the distance of scene points acquired with the 3D sensor to their corresponding transformed planes of the CAD model. The point-to-plane correspondences are determined using the surface-based 3D matching (Drost et al., 2010; Drost, 2016; Steger et al., 2018, Chapter 3.12.3). Note that the pose of the calibration object in the SCS, which is also returned by the surface-based 3D matching, is not used in the calibration. The unknowns in the robot calibration are the kinematic parameters of the robot joints, the parameters of the hand–eye pose that describe the pose of the 3D sensor with respect to the robot tool, and the parameters of the world–base pose that describe the pose of the calibration object with respect to the robot base. Like for camera-based robot calibration (cf. Section 5.1), our proposed calibration model supports the acquisition of 3D scans of the calibration object in multiple poses to achieve the best accuracy.

6.2. Calibration model

Let n_c be the number of different calibration object poses and $n_{r,l}$ the number of robot poses (i.e., the number of acquired calibration scans) at calibration object pose l ($l = 1, \dots, n_c$). Furthermore, let the 3D points that have been acquired at calibration object pose l and robot pose j that have been matched to one of the planes of the calibration object be denoted by $\boldsymbol{p}_{l,j,k}$ ($k = 1, \dots, n_{p,l,j}$). The corresponding calibration object plane in the triangulated CAD model is denoted by $\boldsymbol{\pi}_{c(l,j,k)}$. Note that $\boldsymbol{p}_{l,j,k}$ is given in the SCS while $\boldsymbol{\pi}_{c(l,j,k)}$ is given in the WCS. To compute the distance $d_{l,j,k}$ of the 3D points to their corresponding planes of the calibration object, the planes are transformed into the SCS. Hence,

$$d_{l,j,k} = \left({}^{\mathrm{s}}\mathbf{G}_{\mathrm{t}}\, {}^{\mathrm{t}}\mathbf{G}_{\mathrm{b},l,j}\, {}^{\mathrm{b}}\mathbf{G}_{\mathrm{w},l}\boldsymbol{\pi}_{c(l,j,k)}\right)^{\top}\boldsymbol{p}_{l,j,k} \ , \tag{28}$$

where ${}^{\mathrm{s}}\mathbf{G}_{\mathrm{t}}$ represents the inverse transpose of the hand–eye pose ${}^{\mathrm{s}}\mathbf{H}_{\mathrm{t}}$, ${}^{\mathrm{t}}\mathbf{G}_{\mathrm{b},l,j}$ represents the inverse transpose of the robot pose ${}^{\mathrm{t}}\mathbf{H}_{\mathrm{b},l,j}$ that is used to acquire calibration scan j of the calibration object at calibration object pose l, and ${}^{\mathrm{b}}\mathbf{G}_{\mathrm{w},l}$ represents the inverse transpose of the pose of the calibration object ${}^{\mathrm{b}}\mathbf{H}_{\mathrm{w},l}$ with respect to the robot base at pose l (cf. Section 3). The matrix ${}^{\mathrm{t}}\mathbf{H}_{\mathrm{b},l,j}$ is obtained by applying the forward kinematics of (7) with the joint parameters given in (9)–(13).

We denote the set of unknown kinematic parameters, i.e., the offsets for the DHHM parameters, of joint i by $\boldsymbol{d}_i$, the vector that contains the six transformation parameters of the unknown hand–eye pose ${}^{\mathrm{s}}\mathbf{H}_{\mathrm{t}}$ by $\boldsymbol{e}_{\mathrm{s}}$, and the vector of the unknown calibration object poses ${}^{\mathrm{b}}\mathbf{H}_{\mathrm{w},l}$ by $\boldsymbol{e}_{\mathrm{b},l}$. For robot axes that are nominally perpendicular, $\boldsymbol{d}_i = (\Delta_{\theta_i}, \Delta_{d_i}, \Delta_{a_i}, \Delta_{\alpha_i})^{\top}$, while for robot axes that are nominally parallel, $\boldsymbol{d}_i = (\Delta_{\theta_i}, \Delta_{a_i}, \Delta_{\alpha_i}, \Delta_{\beta_i})^{\top}$.

6.3. Parameter estimation in the Gauss–Helmert model

Minimizing (28) corresponds to the minimization of a geometric error. Compared to the minimization of algebraic errors, which do not represent a geometric quantity, this typically results in a higher accuracy. Note that in contrast to camera-based robot calibration (cf. Section 5.3), the observations are not an explicit function of the unknowns. Therefore, the Gauss–Markov model cannot be applied. Instead, for the optimization we apply the parameter estimation in the Gauss–Helmert model (Förstner and Wrobel, 2016, Chapter 4.8.2; Niemeier, 2008, Chapter 5.2), which is also a standard tool in photogrammetry and enables the correct modeling of the statistical properties of the observations and unknowns.

Like the Gauss–Markov model, the Gauss–Helmert model distinguishes between the functional model and the stochastic model. In the functional model, the relations between the observations and the unknowns are described. The stochastic model describes the observations and the unknowns as random variables with uncertainties, which are described by covariances.

The functional model in the Gauss–Helmert model is written as implicit functions

$$\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{l}) = 0 \ , \tag{29}$$

where $\boldsymbol{f}$ corresponds to (28). The vector $\boldsymbol{l}$ contains the measured 3D points:

$$\boldsymbol{l} = (\boldsymbol{p}_{1,1,1}^{\top}, \dots, \boldsymbol{p}_{1,1,n_{p,1,1}}^{\top}, \dots)^{\top} \ . \tag{30}$$

If n_p is the total number of matched 3D points over all calibration scans, the number of observations n_l equals $3n_p$. Each matched 3D point yields one equation in the form of (28), thus resulting in n_p equations in total. Initially, the vector of unknowns $\boldsymbol{x}$ is built as

$$\boldsymbol{x} = (\boldsymbol{e}_{\mathrm{s}}^{\top}, \boldsymbol{e}_{\mathrm{b},1}^{\top}, \dots, \boldsymbol{e}_{\mathrm{b},n_c}^{\top}, \boldsymbol{d}_1^{\top}, \dots, \boldsymbol{d}_{n_a}^{\top})^{\top} \ , \tag{31}$$

containing the six parameters of the hand–eye pose, the $6n_c$ parameters of the n_c calibration object poses, and the $4n_a$ unknown kinematic parameters for each of the n_a joints. Because of the model degeneracies that we described in Section 3.3, not all kinematic parameters can be estimated simultaneously with the parameters of the calibration object poses (i.e., the world–base poses) and the hand–eye pose. Thus, as in the camera-based robot calibration, we exclude Δ_{θ_1} and Δ_{d_1} of the parameters $\boldsymbol{d}_1$ of the first joint and all parameters $\boldsymbol{d}_{n_a}$ of the last joint from the parameter estimation by deleting these parameters from $\boldsymbol{x}$. Let the final number of unknowns in $\boldsymbol{x}$ be denoted by n_x.

The stochastic model specifies the statistical properties of the observation process. In the majority of applications, it is appropriate to assume that the measured 3D points are uncorrelated and possess

uniform precision. In such cases, we set the $n_l \times n_l$ weight coefficient matrix, i.e., the initial covariance matrix, of the observations to the identity matrix: $\mathtt{Q}_{ll} = \mathtt{I}$. From $\mathtt{Q}_{ll}$, the weight matrix $\mathtt{P}_{ll} = \mathtt{Q}_{ll}^{-1}$ can be computed. If more detailed knowledge about the accuracy of the 3D points is known, the elements in $\mathtt{Q}_{ll}$ can be set to appropriate covariances or the elements in $\mathtt{P}_{ll}$ to appropriate weights.

In the Gauss–Helmert model, we must linearize the functional model with respect to the observations and unknowns. For linearization with respect to the unknowns, we compute the $n_\mathrm{p} \times n_x$ Jacobian $\mathtt{A}$ of $\boldsymbol{f}$ at the initial values of the unknowns $\boldsymbol{x}^{(0)}$ (cf. Section 6.4):

$$\mathtt{A} = \left.\frac{\partial \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{l})}{\partial \boldsymbol{x}}\right|_{\boldsymbol{x}=\boldsymbol{x}^{(0)}} . \tag{32}$$

For linearization with respect to the observations, we compute the $n_\mathrm{p} \times n_l$ Jacobian

$$\mathtt{B} = \left.\frac{\partial \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{l})}{\partial \boldsymbol{l}}\right|_{\boldsymbol{x}=\boldsymbol{x}^{(0)}} . \tag{33}$$

With $\boldsymbol{w} = \boldsymbol{f}(\boldsymbol{x}^{(0)}, \boldsymbol{l})$, we can calculate the corrections for the unknowns by solving

$$\mathtt{A}^\top (\mathtt{B}\mathtt{Q}_{ll}\mathtt{B}^\top)^{-1}\mathtt{A}\Delta\hat{\boldsymbol{x}} = -\mathtt{A}^\top (\mathtt{B}\mathtt{Q}_{ll}\mathtt{B}^\top)^{-1}\boldsymbol{w} \tag{34}$$

for $\Delta\hat{\boldsymbol{x}}$, for example, by Cholesky decomposition. The corrections are applied to the unknowns by

$$\hat{\boldsymbol{x}}^{(1)} = \boldsymbol{x}^{(0)} + \Delta\hat{\boldsymbol{x}} . \tag{35}$$

Eqs. (32)–(35) are repeatedly applied until convergence. This corresponds to minimizing the distances of the 3D points to their corresponding planes of the calibration object.

After convergence, the covariance matrix of the originally observed 3D points is obtained by

$$\mathtt{C}_{ll} = \hat{\sigma}_0^2 \mathtt{Q}_{ll} , \tag{36}$$

with the variance factor

$$\hat{\sigma}_0^2 = \hat{\boldsymbol{v}}^\top \mathtt{P}_{ll} \hat{\boldsymbol{v}} / r , \tag{37}$$

where $r = n_\mathrm{p} - n_x$ denotes the redundancy of the estimation and $\hat{\boldsymbol{v}}$ contains the residuals, which are obtained by

$$\hat{\boldsymbol{v}} = \mathtt{Q}_{ll}\mathtt{B}^\top \hat{\boldsymbol{k}} \tag{38}$$

using the Lagrange multipliers

$$\hat{\boldsymbol{k}} = -\left(\mathtt{B}\mathtt{Q}_{ll}\mathtt{B}^\top\right)^{-1} (\mathtt{A}\Delta\hat{\boldsymbol{x}} + \boldsymbol{w}) . \tag{39}$$

Then, the corrected observations are computed by $\hat{\boldsymbol{l}} = \boldsymbol{l} + \hat{\boldsymbol{v}}$. By applying variance propagation, we obtain the covariance matrix of the estimated unknowns:

$$\mathtt{C}_{\hat{x}\hat{x}} = \hat{\sigma}_0^2 \left(\mathtt{A}^\top \left(\mathtt{B}\mathtt{Q}_{ll}\mathtt{B}^\top\right)^{-1} \mathtt{A}\right)^{-1} . \tag{40}$$

For further details about the Gauss–Helmert model, the reader is referred to Förstner and Wrobel (2016, Chapter 4.8.2) and Niemeier, 2008, Chapter 5.2.

Using the Gauss–Helmert model offers the same advantages as those of the Gauss–Markov model that were described in the last paragraph of Section 5.3.

6.4. Initialization of the unknowns

The initial values for the unknowns are obtained in the same manner as described in Section 5.4. The only difference is that there are no interior camera parameters that need to be initialized.

6.5. Point cloud preprocessing

3D scans typically include some errors, no matter which technology is used by the 3D sensor to perform the 3D reconstruction. These may negatively affect the accuracy of the calibration. For example, specular reflections or occlusions in the scene may cause outliers or inaccurate 3D reconstructions. To reduce or eliminate the effects of these kinds of errors on the optimization, a preprocessing is performed based on the pose calculated by surface-based 3D matching and the corresponding point cloud returned by the 3D sensor. It consists of following steps for each scan:

- All points of the point cloud that lie at a distance of more than 10 mm from the surface of the calibration object are removed.
- The median distance of the remaining scene points that were matched to the planes of the calibration object is calculated and all points that lie at a distance of more than twice the median distance are eliminated.
- The point cloud is subsampled such that approximately 500 points per 3D scan are used in the optimization.

7. Evaluation

In this section, we will show the validity and advantages of our robot calibration approach based on multiple experiments. We will first introduce new evaluation metrics that are motivated by typical vision-guided robot applications. The metrics simultaneously measure the accuracy of the robot calibration and of the hand–eye calibration even in the absence of ground-truth values for the estimated parameters. Then, we will show that the evaluation metrics provide meaningful results and evaluate different aspects of our robot calibration approach based on simulated data. Afterwards, experiments with robots that are available at our laboratories will demonstrate the advantages of our robot calibration on real data.

For the experiments with simulated data, the ground-truth values of the estimated parameters are known. In this case, we are able to directly compute the errors of the estimated parameters. For the hand–eye pose, the error is computed separately for its translation and rotation parts.

7.1. Evaluation metrics for camera-guided robots

To assess the quality of the robot calibration of Section 5 with real data, we propose new evaluation metrics. The basic ideas behind the metrics are that they should express the inaccuracy in the application of the robot because this inaccuracy represents both the inaccuracy of the kinematic calibration and the inaccuracy of the hand–eye calibration, both of which are relevant in real applications. We base this on a typical robot manipulation application, where, in the first step, the pose of an object is determined based on a camera image of the object and, in the second step, the object is manipulated by the robot. For this evaluation purpose, the object is represented by a calibration object, the pose of which relative to the camera can be determined with high accuracy. After the pose of the calibration object has been determined, we move the robot such that the camera sees the calibration object under a predefined reference pose. This step represents the manipulation task. To compute the necessary robot pose and to reach this pose with the robot tool, both the hand–eye pose as well as the kinematic parameters are necessary. Consequently, errors in each of these parameters influence the final camera pose. The camera pose is compared to the predefined reference pose by computing the position differences between the observed and the projected calibration marks in the image. The root mean square (RMS) reprojection error over all calibration marks represents the first new evaluation metric. We will call this metric the RMS evaluation error. Two additional evaluation metrics are the translation and rotation part of the pose difference between the

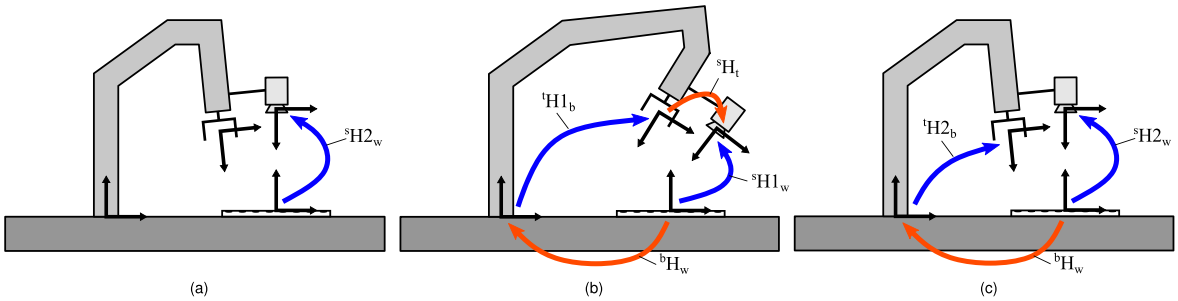


Fig. 3. Evaluation procedure. First, a reference pose $^{s}\mathtt{H2}_{w}$ is specified at which the RMS evaluation error is to be computed (a). Then, the robot is moved to an arbitrary pose $^{t}\mathtt{H1}_{b}$ and an image or 3D scan is acquired (b). Based on the pose of the calibration object $^{s}\mathtt{H1}_{w}$, the robot pose $^{t}\mathtt{H2}_{b}$ that moves the imaging sensor to the reference pose is computed. The robot is moved to the computed pose and the RMS evaluation error is computed (c). Steps (b) and (c) are repeated for different robot poses $^{t}\mathtt{H1}_{b}$.

camera pose and the predefined reference pose. The computation of the metric that simultaneously measures the accuracy of the hand–eye pose and of the kinematic parameters is illustrated in Fig. 3. It includes the following steps:

1. Place a calibration object in the workspace of the robot.
2. Specify a reference pose $^{s}\mathtt{H2}_{w}$ at which the camera should acquire an image of the calibration object to compute the RMS evaluation error. This pose should be chosen such that the calibration object covers the whole camera image. For planar calibration objects, like the one we used in our evaluation (cf. Fig. 2(a)), the pose could be chosen parallel to the calibration object ($\alpha = \beta = \gamma = 0$) and offset by an appropriate distance d in the viewing direction ($t_x = t_y = 0$, $t_z = d$).
3. Project the points of the calibration object into the image plane by using the reference pose $^{s}\mathtt{H2}_{w}$ and the calibrated interior orientation i. Let $\boldsymbol{p}_k^{\text{proj}}$ denote the projection of point k. Note that in this step no image is acquired at the reference pose. The projection is purely synthetic.
4. Move the robot to an arbitrary pose $^{t}\mathtt{H1}_{b}$ for which the calibration object is visible in the camera image. In this step, the calibrated DHHM parameters are used to solve the inverse kinematics.
5. Acquire an image of the calibration object and compute the pose of the calibration object with respect to the camera $^{s}\mathtt{H1}_{w}$ by applying a PnP algorithm or a pose estimation algorithm like Steger et al. (2018, Chapter 3.9) based on the 2D–3D point correspondences. In this step, the calibrated interior orientation of the camera is used.
6. Compute the pose of the calibration object in the robot base coordinate system $^{b}\mathtt{H}_{w} = {}^{b}\mathtt{H1}_{t}\,{}^{t}\mathtt{H}_{s}\,{}^{s}\mathtt{H1}_{w}$, where $^{t}\mathtt{H}_{s}$ denotes the calibrated hand–eye pose.
7. Compute the robot pose $^{t}\mathtt{H2}_{b} = {}^{t}\mathtt{H}_{s}\,{}^{s}\mathtt{H2}_{w}\,{}^{w}\mathtt{H}_{b}$ that brings the camera into the reference pose $^{s}\mathtt{H2}_{w}$ and move the robot accordingly. In this step, again the calibrated DHHM parameters are used to solve the inverse kinematics.
8. Acquire an image and extract the calibration points $\boldsymbol{p}_k^{\text{extr}}$.
9. Compute the RMS evaluation error

$$E_{\text{RMS}} = \sqrt{\frac{1}{n_{\text{w}}} \sum_{k=1}^{n_{\text{w}}} \|\boldsymbol{p}_k^{\text{extr}} - \boldsymbol{p}_k^{\text{proj}}\|_2^2} \,. \tag{41}$$

10. Compute the actual pose $^{s}\mathtt{H2}_{w}^{\text{extr}}$ based on the points $\boldsymbol{p}_k^{\text{extr}}$ by applying a PnP algorithm or pose estimation algorithm like Steger et al. (2018, Chapter 3.9) based on the 2D–3D point correspondences. Again, the calibrated interior orientation is used for this. Then, compute the absolute translation error E_{T} and absolute rotation E_{R} error of the pose $^{s}\mathtt{H2}_{w}\,(^{s}\mathtt{H2}_{w}^{\text{extr}})^{-1}$. Because the translation error depends on the direction of the transformation, we also compute the absolute translation error of $(^{s}\mathtt{H2}_{w})^{-1}\,{}^{s}\mathtt{H2}_{w}^{\text{extr}}$ and average both values.

Note that each of the three metrics E_{RMS}, E_{T}, and E_{R} is able to evaluate the accuracy of the camera-guided robot, including the accuracy of the calibrated interior orientation, which is applied in steps (3), (5), and (10), the accuracy of the calibrated hand–eye pose, which is applied in steps (6) and (7), and the accuracy of the calibrated kinematic parameters, which are applied in solving the inverse kinematics in steps (4) and (7).

The three evaluation metrics can be extended by repeating steps (4)–(10) for different poses $^{t}\mathtt{H1}_{b}$ to cover a broader range of robot configurations. In this case, the mean RMS evaluation error $\bar{E}_{\text{RMS}}$, the mean absolute translation error $\bar{E}_{\text{T}}$, and the mean absolute rotation error $\bar{E}_{\text{R}}$ over all individual E_{RMS}, E_{R}, and E_{T} values are used as the evaluation metrics.

7.2. Evaluation metrics for 3D-sensor-guided robots

To assess the quality of the robot calibration of Section 6 with real data, we use the same principles that were described in Section 7.1. Owing to the different sensor characteristics, the evaluation procedure must be modified as follows:

1. Place a calibration object in the workspace of the robot.
2. Specify a reference pose $^{s}\mathtt{H2}_{w}$ at which the 3D sensor should acquire a scan of the calibration object to compute the RMS evaluation error. This pose should be chosen such that the calibration object covers a substantial part of the 3D scan. For the calibration object we use in our evaluation (cf. Fig. 2(b)), we chose the pose to lie at a suitable distance d perpendicularly above the center of the frustum.
3. Transform the planes of the calibration object into SCS by applying the transformation $^{s}\mathtt{H2}_{w}$ using the corresponding inverse transpose $^{s}\mathtt{G2}_{w}$. Let $\boldsymbol{\pi}_i^{\text{ref}}$ denote the transformed plane i. Note that in this step no 3D scan is acquired at the reference pose. The transformation is purely synthetic.
4. Move the robot to an arbitrary pose $^{t}\mathtt{H1}_{b}$ for which the calibration object is visible for the 3D sensor. In this step, the calibrated DHHM parameters are used to solve the inverse kinematics.
5. Acquire a 3D scan of the calibration object and compute the pose of the calibration object with respect to the 3D sensor $^{s}\mathtt{H1}_{w}$ using the surface-based 3D matching.
6. Compute the pose of the calibration object in the robot base coordinate system $^{b}\mathtt{H}_{w} = {}^{b}\mathtt{H1}_{t}\,{}^{t}\mathtt{H}_{s}\,{}^{s}\mathtt{H1}_{w}$, where $^{t}\mathtt{H}_{s}$ denotes the calibrated hand–eye pose.
7. Compute the robot pose $^{t}\mathtt{H2}_{b} = {}^{t}\mathtt{H}_{s}\,{}^{s}\mathtt{H2}_{w}\,{}^{w}\mathtt{H}_{b}$ that brings the 3D sensor into the reference pose $^{s}\mathtt{H2}_{w}$ and move the robot accordingly. In this step, again the calibrated DHHM parameters are used to solve the inverse kinematics.
8. Acquire a 3D scan and compute the actual pose $^{s}\mathtt{H2}_{w}^{\text{extr}}$ of the calibration object with respect to the 3D sensor using the surface-based 3D matching. The matching also determines the point-to-plane correspondences between the subset of the scene points

Table 1
Maximum error amplitudes of the DHHM parameters.

Δ_θ	Δ_d	Δ_a	Δ_α	Δ_β
0.05°	0.2 mm	0.2 mm	0.05°	0.05°

Table 2
Four experiments on simulated data.

Set fixed parameters to ...	Vary sensor noise	Vary DHHM errors
... ground-truth	Experiment 1	Experiment 2
... nominal	Experiment 3	Experiment 4

that correspond to the planes of the calibration object. Let the points that have been matched to one of the planes of the calibration object be denoted by $\boldsymbol{p}_k^{\text{extr}}$ and let the calibration object plane corresponding to $\boldsymbol{p}_k^{\text{extr}}$ be denoted by $\boldsymbol{\pi}_{c(k)}^{\text{ref}}$. Let the number of points in the scene that were matched to a corresponding plane of the calibration object be denoted by n_{m}. Note that $\boldsymbol{\pi}_{c(k)}^{\text{ref}}$ refers to the plane positions computed in step (3) for the reference pose ${}^{\text{s}}\mathtt{H2}_{\text{w}}$ and not to the plane positions in the actual pose ${}^{\text{s}}\mathtt{H2}_{\text{w}}^{\text{extr}}$. Hence, the point-to-plane correspondences determined by the surface-based 3D matching are used to match the points in the scan at the actual pose to where they are expected to lie in the reference pose. Consequently, the metric compares the point positions to the planes of the CAD model in the predefined reference pose (as opposed to the planes of the CAD model in the actual calibration object pose).

9. Compute the RMS evaluation error

$$E_{\text{RMS}} = \sqrt{\frac{1}{n_{\text{m}}} \sum_{k=1}^{n_{\text{m}}} \left((\boldsymbol{\pi}_{c(k)}^{\text{ref}})^{\top} \boldsymbol{p}_k^{\text{extr}} \right)^2} . \tag{42}$$

10. Compute the absolute rotation error E_{R} and the absolute translation error E_{T} of the pose ${}^{\text{s}}\mathtt{H2}_{\text{w}}\,({}^{\text{s}}\mathtt{H2}_{\text{w}}^{\text{extr}})^{-1}$. Furthermore, compute the absolute translation error of $({}^{\text{s}}\mathtt{H2}_{\text{w}})^{-1}\,{}^{\text{s}}\mathtt{H2}_{\text{w}}^{\text{extr}}$ and average the two translation errors.

Like in Section 7.1, the three evaluation metrics can be extended by repeating steps (4)–(10) for different poses ${}^{\text{t}}\mathtt{H1}_{\text{b}}$ to cover a broader range of robot configurations, resulting in $\bar{E}_{\text{RMS}}$, $\bar{E}_{\text{T}}$, and $\bar{E}_{\text{R}}$.

7.3. Experiments on simulated data

To evaluate the calibration of 3D-sensor-guided robots, we simulated an industrial robot arm with six revolute joints by using the nominal DH parameters of a UR3e robot (Universal Robots, 2024) and a 3D sensor mounted at the tool with a hand–eye pose of $(t_x, t_y, t_z) = (50, 80, 100)$ mm, $\alpha = 5°$, $\beta = 355°$, and $\gamma = 0°$. As described in Section 3.2, we converted the DH parameters to DHHM parameters. For the calibration, we simulated a HALCON sheet-of-light calibration object (cf. Fig. 2(b)) of size 150 mm × 100 mm × 40 mm at a world–base pose of $(t_x, t_y, t_z) = (162, -340, 35)$ mm, $\alpha = 347°$, $\beta = 357°$, and $\gamma = 15°$. Note that the origin of the WCS is located in the center of the square plane on top of the frustum of the calibration object.

We added uniform random errors to the DHHM parameters of all robot joints and links with the maximum amplitudes shown in Table 1. We then created 15 random robot poses for which the calibration object is tilted by at most 60° with respect to the 3D sensor coordinate system and for which the calibration object is assured to be detectable in the simulated 3D sensor scan. We subsampled the point cloud so that approximately 300 points are used as calibration object points. Finally, we added isotropic Gaussian noise to the calibration object points.

In the robot calibration, we estimated all parameters $\boldsymbol{x} = (\boldsymbol{e}_{\text{s}}^{\top}, \boldsymbol{e}_{\text{b,1}}^{\top}, \boldsymbol{d}_1^{\top}, \ldots, \boldsymbol{d}_5^{\top})^{\top}$ while excluding Δ_{θ_1} and Δ_{d_1} of the first joint $\boldsymbol{d}_1$ and all parameters of the last joint $\boldsymbol{d}_6$ (see Section 6.3). We initialized the robot parameters that were included in the calibration by setting them to their nominal values. To obtain initial values for the hand–eye pose $\boldsymbol{e}_{\text{s}}$ and the world–base pose $\boldsymbol{e}_{\text{b,1}}$, we executed the linear hand–eye calibration approach of Daniilidis (1999).

We performed four different experiments (see Table 2): In the first experiment, we varied the standard deviation of the calibration object points in the range [0, 2] mm to simulate different amounts of sensor noise. In the second experiment, we fixed the standard deviation of the calibration object points at 0.5 mm. Here, we varied the amplitudes of the random errors of the DHHM parameters by multiplying the maximum error amplitudes in Table 1 by factors in the range [0, 4]. For both experiments, we evaluated two different sub-cases: In the first sub-case, the robot parameters that were excluded from the calibration were set to their ground-truth values, while in the second sub-case, they were set to their nominal values.

We repeated all experiments 20 times for each error or noise level, respectively, and computed the mean absolute errors of the DHHM parameters and of the translation and rotation part of the hand–eye poses. Additionally, we specified a reference pose ${}^{\text{s}}\mathtt{H2}_{\text{w}}$ with $d = 0.4$ m and performed the evaluation procedure described in Section 7.2 for ten random poses ${}^{\text{t}}\mathtt{H1}_{\text{b}}$ and computed the mean RMS evaluation error $\bar{E}_{\text{RMS}}$, the mean absolute translation error $\bar{E}_{\text{T}}$, and the mean absolute rotation error $\bar{E}_{\text{R}}$.

Fig. 4 displays the results of experiment 1, i.e., the errors as a function of the sensor noise when setting the fixed DHHM parameters to their ground-truth values. Fig. 4(a) plots the mean absolute errors in the DHHM parameters for the six robot joints obtained from the results of the robot calibration. The results for the noise-free case show that the robot calibration reliably converges to the correct result. For non-zero noise, the parameter errors increase linearly with the noise level. Because the fixed DHHM parameters are set to their ground-truth values, their errors are always zero. Fig. 4(b) and (c) display the mean absolute errors in the calibrated hand–eye pose for its translation and rotation parts. Note that while our approach calibrates the world–base pose, we do not evaluate it because it represents the pose of the calibration object, which is irrelevant because it is not present in an application that makes use of the calibration data. Again, the errors increase linearly with the noise level. Fig. 4(d) shows the mean RMS evaluation error $\bar{E}_{\text{RMS}}$, while Fig. 4(e) and (f) show the mean absolute errors in the translation part $\bar{E}_{\text{T}}$ and the rotation part $\bar{E}_{\text{R}}$ of the reference pose. As discussed in Section 7.2, these three metrics do not require the ground-truth values of the robot parameters. It can be seen that all three metrics return the results that are desirable for an evaluation metric. They return a zero error if all parameters are correct and the errors increase proportionally to the errors in the robot parameters. Hence, they are geometrically meaningful errors that describe the misalignment of the robot in a gripping task that is caused by the errors in the robot parameters.

To evaluate the standard deviations that are returned by the robot calibration, we display them for experiment 1. The mean standard deviations of the estimated robot parameters and of the estimated hand–eye pose are displayed in Fig. 5. We computed the standard deviation of the rotation and translation part of the hand–eye poses as the vector length of the three individual standard deviations. A comparison to the errors shown in Fig. 4 shows that the standard deviations represent the errors very well. Therefore, they are a reliable measure for quantifying the uncertainty of the results.

Fig. 6 displays the results of experiment 2, i.e., errors depending on the errors in the DHHM parameters when setting the fixed DHHM parameters to their ground-truth values. Fig. 6(a)–(c) show that the errors in the estimated DHHM parameters and the hand–eye pose do not depend on the errors in their initial values, which confirms the good convergence properties of the robot calibration. Fig. 6(d)–(f) show that the errors of the complete robot calibration remain at a very low and constant value. Hence, the results indicate that the robot calibration is able to eliminate all significant errors in the DHHM and hand–eye

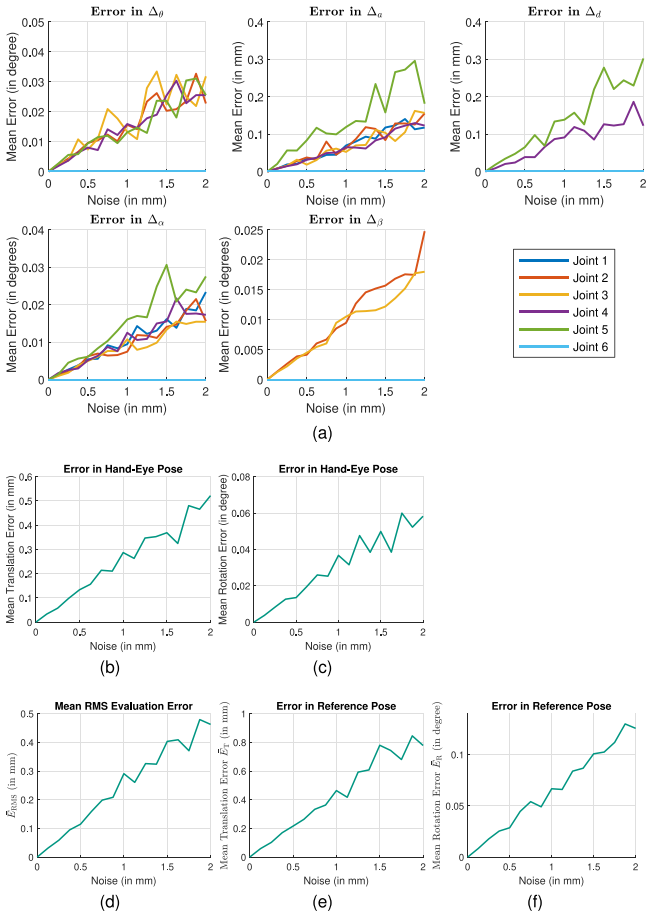


Fig. 4. Robot calibration using a 3D sensor: Results of experiment 1. The fixed DHHM parameters were set to their ground-truth values and the errors are computed depending on the 3D scene point noise. (a) Mean absolute errors in the DHHM parameters for the six robot joints after robot calibration. (b)–(c) Mean absolute errors in the translation and rotation part of the hand–eye pose. (d) Mean RMS evaluation errors $\bar{E}_{\mathrm{RMS}}$. (e), (f) Mean absolute errors in the translation part $\bar{E}_{\mathrm{T}}$ and rotation part $\bar{E}_{\mathrm{R}}$ of the reference pose.

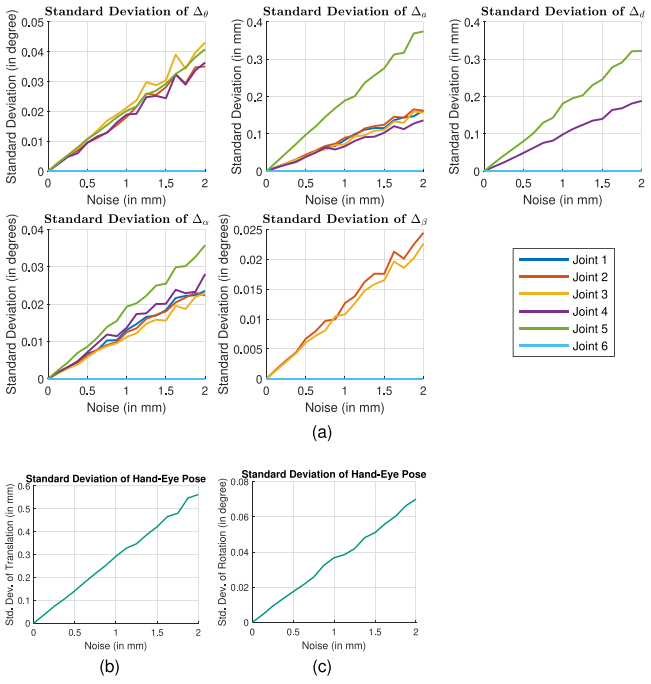


Fig. 5. Standard deviations obtained from our 3D-sensor-based robot calibration based on experiment 1. The fixed DHHM parameters were set to their ground-truth values and the standard deviations are computed depending on the 3D scene point noise. (a) Mean standard deviations of the DHHM parameters for the six robot joints. (b)–(c) Mean standard deviations of the translation and rotation part of the hand–eye pose. To improve comparability, the axis scaling was chosen identical to that of Fig. 4.

parameters and only a small error that reflects the constant noise in the scene points remains.

The results of experiment 3, i.e., errors depending on the sensor noise when setting the fixed DHHM parameters to their erroneous nominal values, are displayed in Fig. 7. In Fig. 7(a), the errors in the DHHM parameters are similar to those of experiment 1. Of course, the fixed DHHM parameters now still exhibit the errors of their initially set values. Despite the errors in the fixed DHHM parameters, the other robot parameters are estimated correctly in the noise-free case. As expected, the remaining errors in the fixed DHHM parameters are now represented by corresponding errors in the hand–eye poses (Fig. 7(b) and (c)). Notably, the errors of the evaluation metrics in Fig. 7(d)–(f) are still zero in the noise-free case. This shows that, even if the fixed DHHM parameters and the hand–eye pose deviate from their true values, the simultaneous calibration of the robot parameters and the hand–eye poses yields a consistent and accurate robot model. Again, the three evaluation metrics show the desirable behavior. The errors are zero for a geometrically consistent set of robot parameters and they increase proportionally to the errors in the robot parameters. Hence, they correctly describe the misalignment error in the gripping task caused by errors in the robot parameters.

Fig. 8 displays the results of experiment 4, i.e., errors depending on the errors in the DHHM parameters when setting the fixed DHHM parameters to their erroneous nominal values. In contrast to experiment 3, as is to be expected, the errors of the fixed parameters increase in Fig. 8(a). Nevertheless, like in experiment 2, the errors of the estimated DHHM parameters are constant. In contrast to experiment 3, the errors of the evaluation metrics in Fig. 8(d)–(f) remain constant, again reflecting only the constant noise in the 3D scene points, which once more indicates a consistent overall system.

To evaluate the calibration of camera-guided robots, we simulated an industrial robot arm with six revolute joints by adopting the nominal DH parameters of a UR5e robot (Universal Robots, 2024) and a camera ($c = 8.43\,\mathrm{mm}$, $\kappa = 2000.0\,\mathrm{m}^{-2}$, $s_x = 5.21\,\mu\mathrm{m}$, $s_y = 5.2\,\mu\mathrm{m}$, $(c_x, c_y) = (645, 502)\,\mathrm{px}$) with radial distortions and 1280×1024 image size mounted at the tool with a hand–eye pose of $(t_x, t_y, t_t) = (50, 30, 100)\,\mathrm{mm}$, $\alpha = 20°$, $\beta = 30°$, and $\gamma = 40°$. Note that we chose a different robot than in the evaluation of the 3D-sensor-based calibration to increase the variability of the evaluation data and hence to strengthen the validity of our calibration model. As described in Section 3.2, we converted the DH parameters to DHHM parameters. For the calibration, we simulated a planar HALCON calibration object (cf. Fig. 2(a)) of size $640\,\mathrm{mm} \times 500\,\mathrm{mm}$ at a world–base pose of $(t_x, t_y, t_z) = (200, 200, 400)\,\mathrm{mm}$, $\alpha = 10°$, $\beta = 20°$, and $\gamma = 30°$.

As in the evaluation of 3D-sensor-guided robot, we added random errors to the DHHM parameters of all robot joints and links with the maximum amplitudes shown in Table 1. Then, we created 15 random robot poses for which at least 60% of the calibration points are visible in the camera image and for which the calibration object is tilted by at most 50° with respect to the image plane. Finally, we added Gaussian noise to the projected calibration points.

During the robot calibration, we estimated all parameters $\boldsymbol{x} = (\boldsymbol{e}_{\mathrm{c}}^{\mathsf{T}}, \boldsymbol{e}_{\mathrm{b},1}^{\mathsf{T}}, \boldsymbol{d}_1^{\mathsf{T}}, \ldots, \boldsymbol{d}_5^{\mathsf{T}}, \boldsymbol{i}^{\mathsf{T}})^{\mathsf{T}}$ while excluding Δ_{θ_1} and Δ_{d_1} of the first joint $\boldsymbol{d}_1$ and all parameters of the last joint $\boldsymbol{d}_6$ (see Section 5.3). We initialized the robot parameters that were included in the calibration by setting them to their nominal values. The interior orientation was initialized with $c = 8.0\,\mathrm{mm}$, $\kappa = 0.0\,\mathrm{m}^{-2}$, $s_x = 5.2\,\mu\mathrm{m}$, $s_y = 5.2\,\mu\mathrm{m}$, $(c_x, c_y) = (640, 512)\,\mathrm{px}$. To obtain initial values for the hand–eye pose $\boldsymbol{e}_{\mathrm{s}}$ and

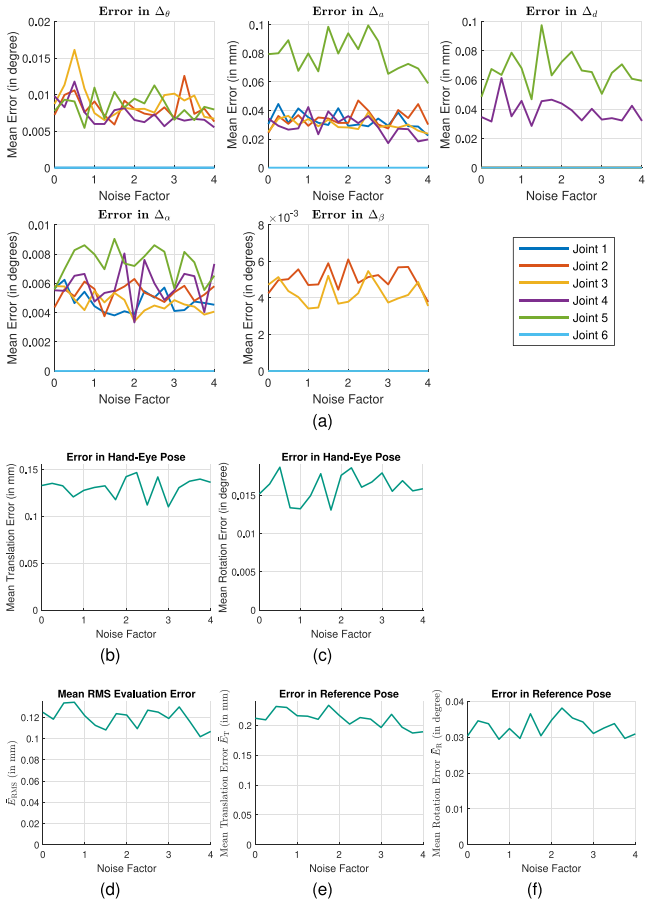


Fig. 6. Robot calibration using a 3D sensor: Results of experiment 2. The fixed DHHM parameters were set to their ground-truth values and the errors are computed depending on the errors in the DHHM parameters (see the caption of Fig. 4 for details).

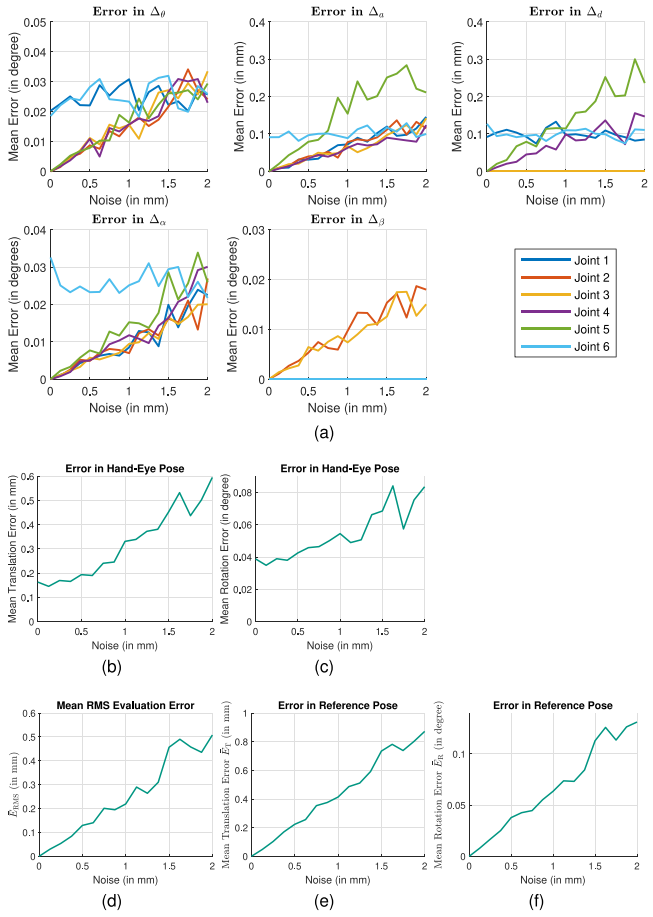


Fig. 7. Robot calibration using a 3D sensor: Results of experiment 3. The fixed DHHM parameters were set to their erroneous nominal values and the errors are computed depending on the image noise (see the caption of Fig. 4 for details).

the world–base pose $\boldsymbol{e}_{\mathrm{b},1}$, we executed the linear hand–eye calibration approach of Daniilidis (1999).

Initially, we again analyzed four different experiments (see Table 2): In experiments 1 and 3, we varied the standard deviation of the image points in the range $[0, 4]$ px. In experiments 2 and 4, we set the standard deviation of the image points to 1 px and instead varied the amplitudes of the random errors of the DHHM parameters by multiplying the maximum error amplitudes in Table 1 with factors in the range $[0, 4]$.

For each error or noise level, respectively, we repeated all experiments 20 times and computed the mean absolute errors of the DHHM parameters and of the translation and rotation part of the hand–eye poses. Furthermore, to compute the mean RMS evaluation error $\bar{E}_{\mathrm{RMS}}$, the mean absolute translation error $\bar{E}_{\mathrm{T}}$, and the mean absolute rotation error $\bar{E}_{\mathrm{R}}$, we specified a reference pose ${}^{\mathrm{s}}\mathbf{H2}_{\mathrm{w}}$ with $t_z = d = 0.5\,\mathrm{m}$ and performed the evaluation procedure described in Section 7.1 for 10 random poses ${}^{\mathrm{t}}\mathbf{H1}_{\mathrm{b}}$.

Because the results of the four experiments were qualitatively equivalent to the results obtained in the evaluation of the 3D-sensor-guided robot, we restrict ourselves to presenting the results of experiment 1, i.e., the errors depending on the image noise when setting the fixed DHHM parameters to their ground-truth values. In Fig. 9(a), the mean absolute errors in the DHHM parameters for the six robot joints after robot calibration are plotted. As in the evaluation of the 3D-sensor-guided robot (cf. Fig. 4), the robot calibration reliably converges to the correct result in the noise-free case. Furthermore, the errors linearly increase with the noise level. Fig. 9(b) and (c) show the mean absolute errors of the resulting hand–eye pose in translation and rotation. Again, in the noise-free case, the hand–eye pose is estimated correctly. Furthermore, the errors increase linearly with the noise level. Fig. 9(d) shows the mean RMS evaluation error $\bar{E}_{\mathrm{RMS}}$, Fig. 9(e) and (f) the mean absolute errors in the translation part $\bar{E}_{\mathrm{T}}$ and the rotation part $\bar{E}_{\mathrm{R}}$ of the reference pose. The corresponding mean standard deviations of the estimated robot parameters and of the estimated hand–eye pose are plotted in Fig. 10. As in the evaluation of the 3D-sensor-guided robot, the standard deviations represent the errors very well.

To investigate the influence of the number of robot poses on the calibration result, we performed an additional experiment (experiment 5): we set the fixed DHHM parameters to their ground truth values, the standard deviation of the image points to 1 px, and added uniform random errors to the DHHM parameters of all robot joints and links with the maximum amplitudes shown in Table 1. We varied the number of robot poses in the range $[6, 30]$, repeated each experiment 20 times, and computed the same evaluation measures as in experiments 1 to 4. The results are shown in Fig. 11. As is to be expected, the errors decrease with increasing number of robot poses. The experiment suggests that at least 20 robot poses should be used for robot calibration in practice since a considerable reduction in error values can be observed up to this number.

In conclusion, the experiments on simulated data show that our robot calibration is able to robustly converge to the true values in the noise-free case. When noise is present, the remaining errors in the estimated DHHM parameters are small and mainly reflect the noise in the 3D scene points or in the image points, respectively. Furthermore, the experiments show that the errors decrease with the number of robot poses. Moreover, the standard deviations that are returned by our robot

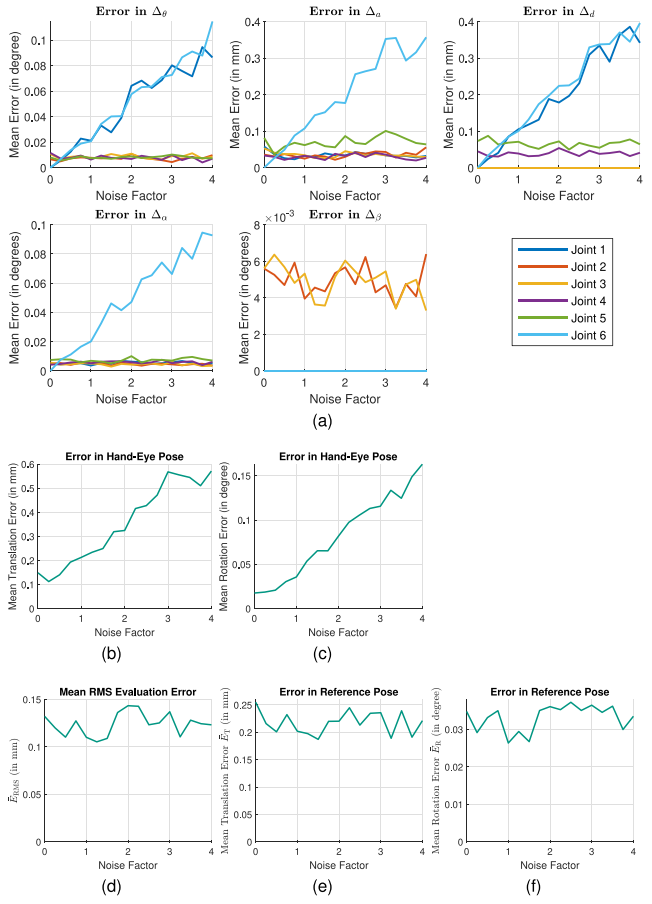


Fig. 8. Robot calibration using a 3D sensor: Results of experiment 4. The fixed DHHM parameters were set to their erroneous nominal values and the errors are computed depending on the errors in the DHHM parameters (see the caption of Fig. 4 for details).

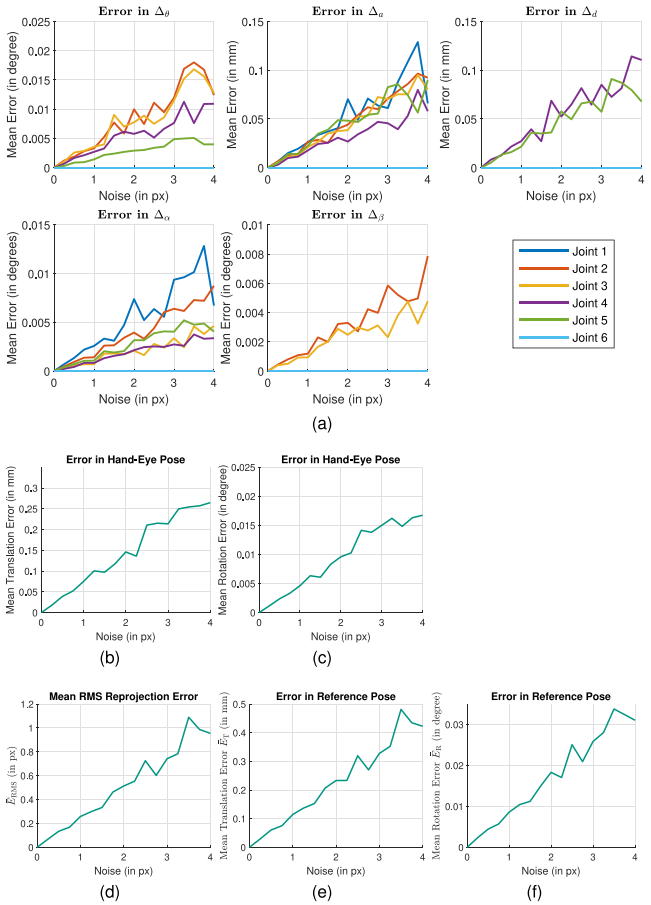


Fig. 9. Robot calibration using a camera: Results of experiment 1. The fixed DHHM parameters were set to their ground-truth values and the errors are computed depending on the image noise (see the caption of Fig. 4 for details).

calibration are meaningful. The results also show that the mean RMS evaluation error and the mean absolute errors in the reference pose are meaningful metrics for analyzing the accuracy of robot calibration approaches in the absence of ground-truth values. The metrics indicate that the simultaneous calibration of the robot and the hand–eye pose yields a consistent and accurate system.

7.4. Experiments on real data

To evaluate our method on real data, we calibrated a six-axis UR3e robot arm from Universal Robots that is available at MVTec using a 3D sensor and a UR3e robot that is available at KIT's Institute of Photogrammetry and Remote Sensing using an industrial camera.

The experiments for 3D-sensor-guided robots on real data use the following three steps:

1. Acquire the data for the hand–eye calibration and robot calibration.
2. Perform the hand–eye calibration and robot calibration.
3. Evaluate the results.

Two different experiments were conducted to compare our approach with an uncalibrated robot, i.e., a robot with its nominal DH parameters (Universal Robots, 2024). The uncalibrated robot is evaluated in experiment 1 while experiment 2 evaluates our robot calibration, including the hand–eye calibration.

For both experiments, we mounted a Zivid Two M70 sensor on the robot's end effector. The Zivid Two sensor uses a structured-light approach for the 3D reconstruction (Zivid AS, 2023). We used a HALCON sheet-of-light calibration object (cf. Fig. 2(b)) of size 150 mm×100 mm×40 mm for the calibration and the evaluation. The setup is shown in Fig. 12.

Fig. 13(a) shows the sensor and calibration object poses used for the hand–eye calibration and our robot calibration. The calibration object displayed in black in the figure was placed on the base mounting surface in front of the robot arm and the calibration object displayed in blue was placed in a pose tilted by 90° above the base mounting surface. For each of the two calibration object poses, 15 robot poses were taught. The robot poses were chosen such that the calibration object covered a substantial part of the 3D scan and such that a large variety of joint angles were assumed by the robot. For experiment 1 (uncalibrated robot), a hand–eye calibration was performed with the nominal kinematic parameters for the pose of the calibration object displayed in black and the 3D sensor poses displayed in gray using the non-linear approach of Steger et al. (2018, Chapter 3.13.5). For experiment 2, the 3D scans were acquired with the calibration object in the poses displayed in black and blue. The hand–eye calibration was performed for both calibration object poses and the hand–eye poses were averaged to obtain starting values for the hand–eye parameters. Our robot calibration was then performed with all scans of both calibration object poses. Fig. 14 shows an example of an acquired point cloud and the pose of the calibration object obtained by using surface-based matching.

Table 3 shows the initial as well as the optimized DHHM and 3D sensor pose parameters. The estimated parameters were then used to control the robot in experiment 2 using the corresponding inverse kinematics.

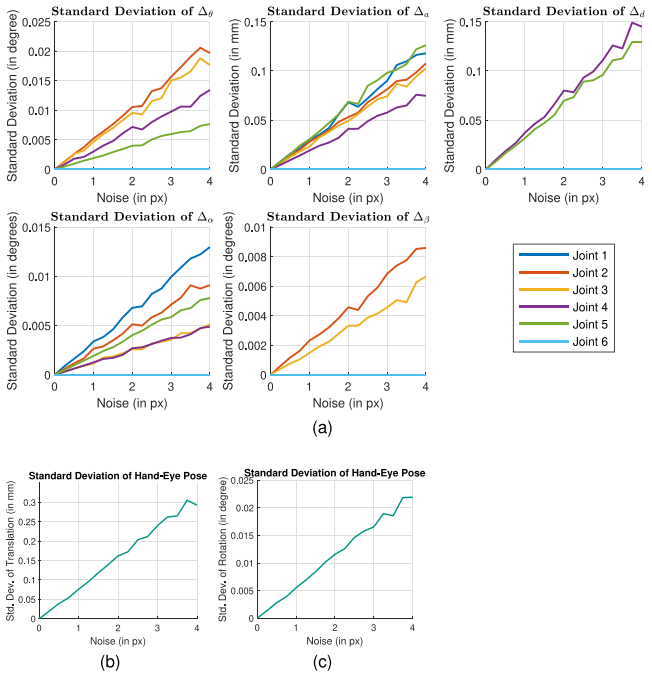


Fig. 10. Standard deviations obtained from our camera-based robot calibration based on experiment 1. The fixed DHHM parameters were set to their ground-truth values and the standard deviations are computed depending on the image noise. (a) Mean standard deviations of the DHHM parameters for the six robot joints. (b)–(c) Mean standard deviations of the translation and rotation part of the hand–eye pose. To improve comparability, the axis scaling was chosen identical to that of Fig. 9.

Table 3
Initial values, estimated values, and standard deviations of the estimated values of our 3D-sensor-guided robot calibration (experiment 2).

Parameter	Initial value	Estimated value	Standard deviation
DHHM parameters			
Δ_{a_1} (mm)	0.0	0.033	24.70×10^{-3}
Δ_{α_1} (°)	0.0	0.072	2.70×10^{-3}
Δ_{θ_2} (°)	0.0	0.053	4.81×10^{-3}
Δ_{a_2} (mm)	0.0	−0.137	23.32×10^{-3}
Δ_{α_2} (°)	0.0	0.162	2.65×10^{-3}
Δ_{β_2} (°)	0.0	0.008	2.61×10^{-3}
Δ_{θ_3} (°)	0.0	−0.063	3.46×10^{-3}
Δ_{a_3} (mm)	0.0	0.247	15.84×10^{-3}
Δ_{α_3} (°)	0.0	−0.492	1.64×10^{-3}
Δ_{β_3} (°)	0.0	−0.122	2.04×10^{-3}
Δ_{θ_4} (°)	0.0	−0.093	6.22×10^{-3}
Δ_{d_4} (mm)	0.0	−1.025	24.79×10^{-3}
Δ_{a_4} (mm)	0.0	−0.143	28.18×10^{-3}
Δ_{α_4} (°)	0.0	−0.064	3.75×10^{-3}
Δ_{θ_5} (°)	0.0	−0.061	4.63×10^{-3}
Δ_{d_5} (mm)	0.0	0.777	32.57×10^{-3}
Δ_{a_5} (mm)	0.0	−0.618	33.19×10^{-3}
Δ_{α_5} (°)	0.0	0.095	4.67×10^{-3}
${}^{s}\mathbf{H}_{t}$			
t_x (mm)	54.478	55.015	23.89×10^{-3}
t_y (mm)	82.472	84.976	29.09×10^{-3}
t_z (mm)	−7.604	−7.036	40.63×10^{-3}
α (°)	0.178	0.483	3.56×10^{-3}
β (°)	355.197	355.004	3.13×10^{-3}
γ (°)	358.720	358.551	4.53×10^{-3}

The standard deviations of all optimized parameters are also displayed in Table 3. They are relatively small when compared to the data in Fig. 4. The optimization returned a value of $\hat{\sigma}_0 = 0.1297$ mm. According to (36), this is an estimate for the precision of the points reconstructed by the 3D sensor. The standard deviations in our real experiments correspond very well with those in Fig. 5 for the estimated point precision, i.e., noise level. Furthermore, the value of $\hat{\sigma}_0$ corresponds very well with the local planarity precision and the global planarity trueness that are given in the technical specification of the Zivid Two M70 sensor for the working distance that we used in our experiments (Zivid AS, 2023). This, again, validates the usefulness of the estimated standard deviations of the robot parameters as well as the estimate of the precision of the 3D points.

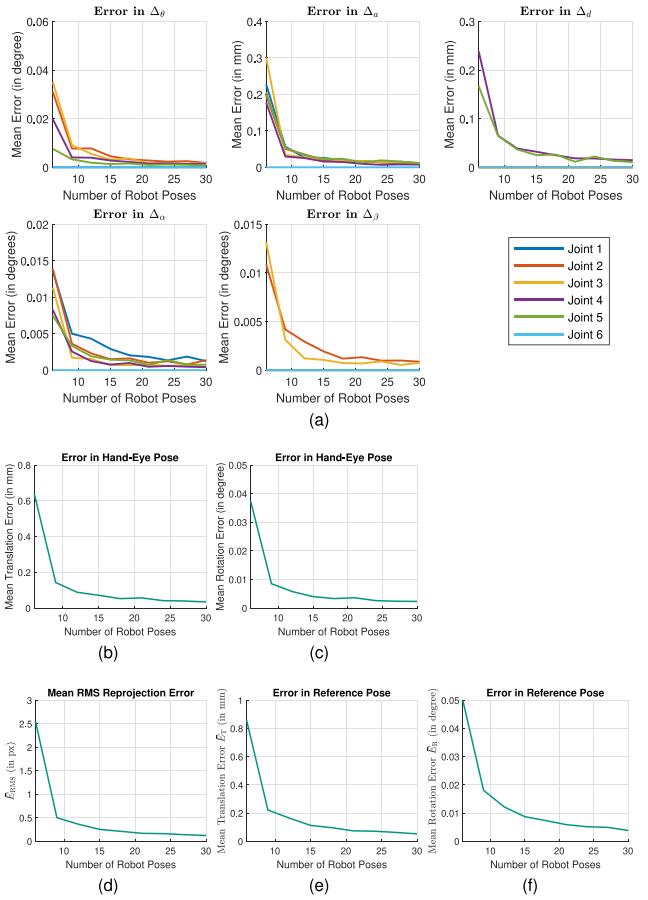


Fig. 11. Robot calibration using a camera: Results of experiment 5. The fixed DHHM parameters were set to their ground-truth values and the errors are computed depending on the number of robot poses (see the caption of Fig. 4 for details).

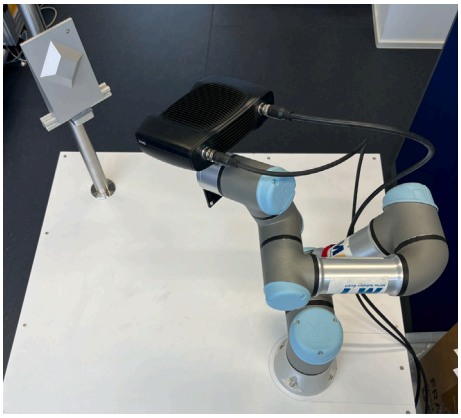


Fig. 12. Setup used for evaluating our robot calibration on real data using a 3D sensor. A Zivid Two M70 sensor that is mounted on the end effector of a UR3e robot arm acquires 3D points of a HALCON sheet-of-light calibration object.

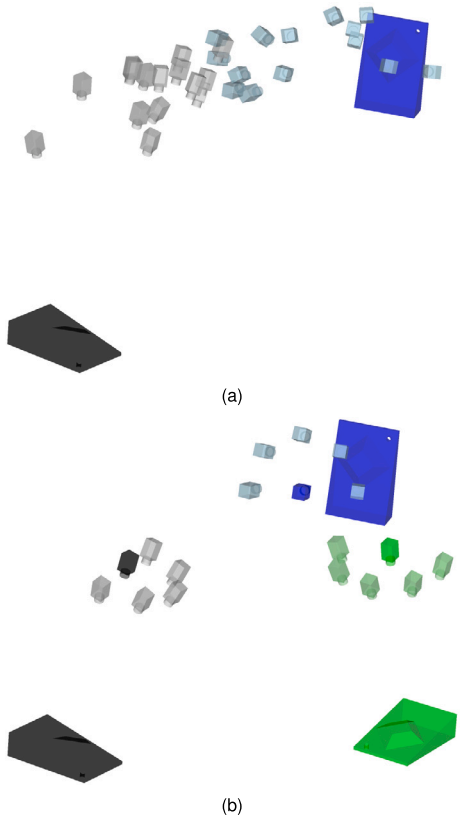


Fig. 13. (a) Visualization of the calibration object poses ${}^{b}\mathbf{H}_{w,l}$ and the 3D sensor poses that result from the robot poses ${}^{t}\mathbf{H}_{b,l,j}$ that were used for the robot calibration experiments. (b) Visualization of the calibration object poses and the 3D sensor poses that result from the robot poses ${}^{t}\mathbf{H1}_{b}$ that were used for the evaluation experiments. In both cases, 3D sensor poses corresponding to a particular calibration object pose are displayed in matching colors. In (b), the evaluation reference poses ${}^{s}\mathbf{H2}_{w}$ are displayed as opaque objects. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

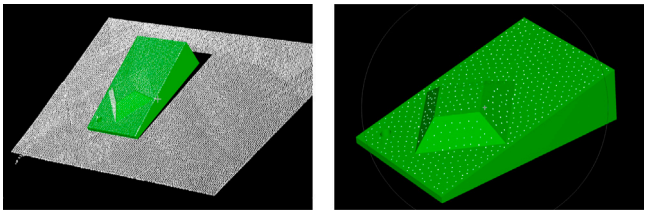


Fig. 14. (a) Acquired unprocessed point cloud (white, slightly downsampled for better visibility) and pose of the calibration object (green) for one example robot pose. (b) Preprocessed point cloud and calibration object pose (cf. Section 6.5). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Fig. 13(b) shows the three evaluation poses that were used for experiments 1 and 2. A gripping motion is simulated, where for each pose, five poses of the sensor (transparent colors) and the reference pose ${}^{s}\mathbf{H2}_{w}$ (opaque colors) are used to evaluate the accuracy of the respective robot kinematics. Note that the evaluation poses are different from the calibration poses in Fig. 13(a). Table 4 shows the average error of all the evaluation poses, split into the RMS evaluation error $\bar{E}_{\mathrm{RMS}}$ as well as the translation error $\bar{E}_{\mathrm{T}}$ and the rotation error $\bar{E}_{\mathrm{R}}$ (cf. Section 7.2). The uncalibrated robot (experiment 1) leads to significantly larger errors than our proposed calibration (experiment 2).

We now turn to the experiments for camera-guided robots on real data. They use the following four steps:

Table 4
Evaluation results of the two 3D-sensor-guided robot calibration experiments.

Evaluation metric	Experiment 1 Uncalibrated robot	Experiment 2 Proposed calibration
$\bar{E}_{\mathrm{RMS}}$ (mm)	1.90	0.28
$\bar{E}_{\mathrm{T}}$ (mm)	1.82	0.45
$\bar{E}_{\mathrm{R}}$ (°)	0.21	0.06

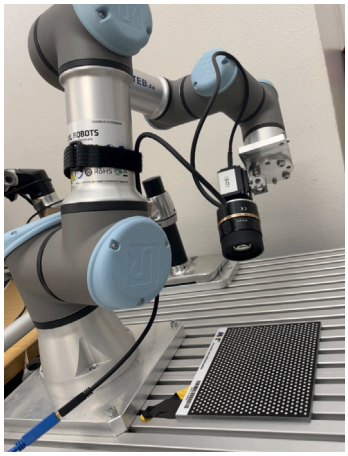


Fig. 15. Setup used for evaluating our robot calibration on real data using a camera. A monochrome IDS U3-3280SE-M-GL that is mounted on the end effector of a UR3e robot arm acquires images of a HALCON calibration object.

1. Perform an initial camera calibration.
2. Acquire data for the hand–eye calibration and robot calibration.
3. Perform hand–eye calibration and robot calibration.
4. Evaluate the results.

We performed the same two experiments as for the 3D-sensor-guided robot. Experiment 1 evaluates the uncalibrated robot while experiment 2 evaluates our robot calibration, including the hand–eye calibration and simultaneous camera calibration. An initial camera calibration is performed to ensure consistent initial camera parameters throughout the experiments.

For both experiments, we mounted a monochrome IDS U3-3280SE-M-GL camera with image size 2448 × 2440 and a 12 mm TAMRON M23FM12 lens on the robot's end effector. We used a planar HALCON calibration object (MVTec Software GmbH, 2024, Chapter 3.2) of size 160 mm × 120 mm for the calibration. The setup is shown in Fig. 15.

For the initial camera calibration, we placed the calibration object approximately horizontally in the center of the workspace (${}^{b}\mathbf{H}_{w,1}$) and acquired $n_{r,1} = 30$ images at different robot poses. The robot poses were randomly distributed while ensuring that the camera poses cover a large pose range. In Fig. 16, the calibration object pose and the camera poses are visualized in blue. We chose the polynomial model described in Section 4 to model lens distortions. The initial values of the camera parameters are shown in Table 5. Because for perspective cameras, c, s_x, and s_y cannot be determined uniquely (Steger et al., 2018, Chapter 3.9.4), we excluded s_y from the calibration. The RMS reprojection error after camera calibration was 0.27 px.

The robot poses and the poses of the calibration object with respect to the camera obtained from step (1) were also utilized for the hand–eye calibration of the uncalibrated robot. The camera parameters were set to the results of the camera calibration from step (1). For hand–eye calibration, we used the non-linear approach of Steger et al. (2018, Chapter 3.13.5).

For our robot calibration, we also used the robot poses from the camera calibration step. However, to better cover the working range of the robot joints, we used two additional calibration object poses ${}^{b}\mathbf{H}_{w,2}$

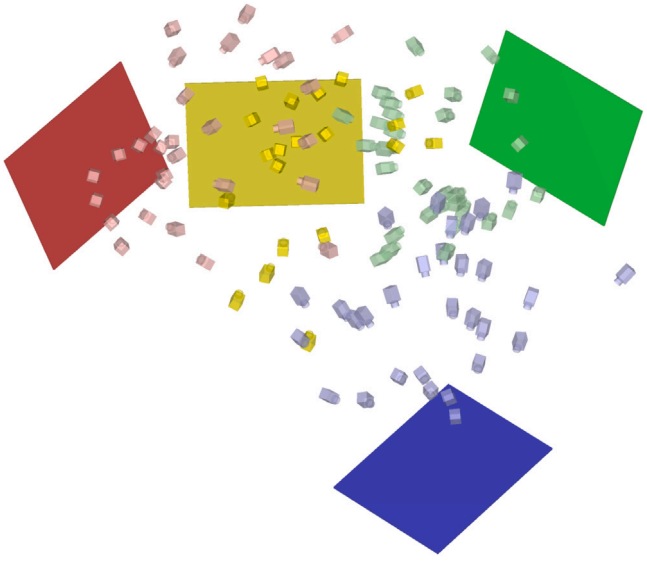
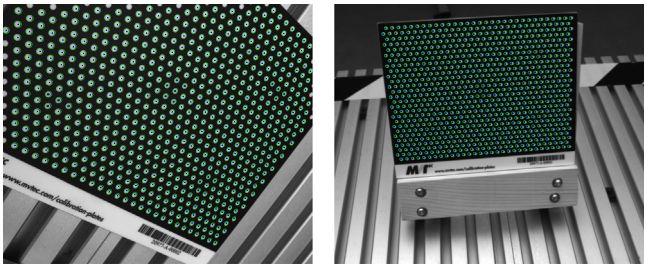


Fig. 16. Visualization of the calibration object poses ${}^{\mathrm{b}}\mathbf{H}_{\mathrm{w},l}$ and the camera poses that result from the robot poses ${}^{\mathrm{t}}\mathbf{H}_{\mathrm{b},l,j}$. The blue (${}^{\mathrm{b}}\mathbf{H}_{\mathrm{w},1}$), green (${}^{\mathrm{b}}\mathbf{H}_{\mathrm{w},2}$), and red (${}^{\mathrm{b}}\mathbf{H}_{\mathrm{w},3}$) calibration object poses are used for the robot calibration. The yellow calibration object pose (${}^{\mathrm{b}}\mathbf{H}_{\mathrm{w},4}$) is used for the evaluation. The camera poses corresponding to a particular calibration object pose are displayed in matching colors. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)



(a) (b)

Fig. 17. Two examples of acquired calibration images with extracted contours (green) and computed centers (blue) of the calibration marks. (a) Horizontal calibration object pose. (b) Tilted calibration object pose. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

and ${}^{\mathrm{b}}\mathbf{H}_{\mathrm{w},3}$, which were tilted by approximately 65° to the horizontal. For each additional calibration object pose, we acquired $n_{\mathrm{r},2} = n_{\mathrm{r},3} = 30$ images at different robot poses, resulting in a total of 90 calibration images at $n_{\mathrm{c}} = 3$ calibration object poses, which are visualized in Fig. 16 in red, green, and blue together with the corresponding camera poses. For each acquired calibration image, additionally the joint angles $\tilde{\theta}_{l,j,i}$ and the robot poses ${}^{\mathrm{t}}\mathbf{H}_{\mathrm{b},l,j}$ were stored. Fig. 17 shows two examples of the acquired calibration images and the extracted calibration marks.

We initialized the DH parameters with the nominal DH parameters. The initial camera parameters were set to the results of the initial camera calibration from step (1). To obtain initial values for the hand–eye pose and the calibration object poses, we performed a separate hand–eye calibration for each calibration object pose. The average of the resulting three hand–eye poses served as initial values for the robot calibration. During the robot calibration, we estimated all parameters $\boldsymbol{x} = (\boldsymbol{e}_{\mathrm{c}}^{\mathsf{T}}, \boldsymbol{e}_{\mathrm{b},1}^{\mathsf{T}}, \boldsymbol{e}_{\mathrm{b},2}^{\mathsf{T}}, \boldsymbol{e}_{\mathrm{b},3}^{\mathsf{T}}, \boldsymbol{d}_1, \ldots, \boldsymbol{d}_5, \boldsymbol{i})^{\mathsf{T}}$ while excluding Δ_{θ_1} and Δ_{d_1} of $\boldsymbol{d}_1$ of the first joint in addition to all DHHM parameters of the last joint $\boldsymbol{d}_6$ (see Section 5.3). Again, s_y was excluded as well. The initial values are shown in Table 5. The result of the robot calibration are the DHHM parameters, the hand–eye pose, the calibration object poses, and the camera parameters. All results except for the calibration object poses are shown in Table 5 together with their estimated standard deviations.

To compute the mean RMS evaluation error $\bar{E}_{\mathrm{RMS}}$, the mean absolute translation error $\bar{E}_{\mathrm{T}}$, and the mean absolute rotation error $\bar{E}_{\mathrm{R}}$, we specified a reference pose ${}^{\mathrm{s}}\mathbf{H2}_{\mathrm{w}}$ with $t_z = d = 0.2\,\mathrm{m}$ and performed the evaluation procedure described in Section 7.1 for 20 random poses ${}^{\mathrm{t}}\mathbf{H1}_{\mathrm{b}}$. The distance of the reference pose was chosen such that the calibration object fills the entire image. The calibration object pose and the 20 evaluation poses are visualized in yellow in Fig. 16. The results of the two experiments are shown in Table 6. Compared to the uncalibrated robot (experiment 1), our proposed calibration (experiment 2) reduces the errors significantly.

To show that the DHHM parameters and the distortion parameters are geometrically significant, and hence cannot be omitted from the model, we performed a significance test. For each parameter, we formulated the null-hypothesis that the respective parameter is equal to 0. Then, we computed the test criterion T proposed by Grün (1978). The null-hypothesis is true with a predefined significance level α if T is below a threshold value. The threshold value corresponds to the percentile value $F(1-\alpha, 1, r)$ of the Fisher distribution with the

Table 5
Initial values, estimated values, standard deviations of the estimated values, and test statistics of our camera-guided robot calibration (experiment 2).

Parameter	Initial value	Estimated value	Standard deviation	Test statistic T
DHHM parameters				
Δ_{a_1} (mm)	0.0	0.114	1.43×10^{-3}	6.39×10^{3}
Δ_{α_1} (°)	0.0	0.046	0.20×10^{-3}	55.09×10^{3}
Δ_{θ_2} (°)	0.0	0.012	0.37×10^{-3}	1.03×10^{3}
Δ_{a_2} (mm)	0.0	−0.215	1.91×10^{-3}	12.75×10^{3}
Δ_{α_2} (°)	0.0	0.002	0.12×10^{-3}	0.26×10^{3}
Δ_{β_2} (°)	0.0	0.008	0.15×10^{-3}	3.22×10^{3}
Δ_{θ_3} (°)	0.0	−0.020	0.24×10^{-3}	6.61×10^{3}
Δ_{a_3} (mm)	0.0	−0.162	1.96×10^{-3}	6.81×10^{3}
Δ_{α_3} (°)	0.0	0.124	0.10×10^{-3}	1.54×10^{6}
Δ_{β_3} (°)	0.0	0.220	0.15×10^{-3}	2.07×10^{6}
Δ_{θ_4} (°)	0.0	−0.016	0.23×10^{-3}	4.64×10^{3}
Δ_{d_4} (mm)	0.0	0.203	1.59×10^{-3}	16.39×10^{3}
Δ_{a_4} (mm)	0.0	−0.087	0.73×10^{-3}	21.92×10^{3}
Δ_{α_4} (°)	0.0	−0.108	0.08×10^{-3}	2.34×10^{6}
Δ_{θ_5} (°)	0.0	−0.010	0.33×10^{-3}	0.99×10^{3}
Δ_{d_5} (mm)	0.0	−0.261	1.99×10^{-3}	17.20×10^{3}
Δ_{a_5} (mm)	0.0	0.048	2.27×10^{-3}	0.45×10^{3}
Δ_{α_5} (°)	0.0	−0.102	0.28×10^{-3}	0.14×10^{6}
${}^{\mathrm{s}}\mathbf{H}_{\mathrm{t}}$				
t_x (mm)	−0.351	−0.090	9.52×10^{-3}	
t_y (mm)	76.064	75.990	8.03×10^{-3}	
t_z (mm)	−129.101	−128.916	6.61×10^{-3}	
α (°)	0.211	0.189	3.58×10^{-3}	
β (°)	0.401	0.432	4.25×10^{-3}	
γ (°)	224.999	225.102	0.41×10^{-3}	
Camera parameters				
c (mm)	12.0	12.123	0.32×10^{-3}	
K_1 (m^{-2})	0.0	661.240	6.07	11.85×10^{3}
K_2 (m^{-4})	0.0	-5.063×10^{6}	0.52×10^{6}	95.68
K_3 (m^{-6})	0.0	112.398×10^{9}	13.04×10^{9}	74.30
P_1 (m^{-1})	0.0	13.198×10^{-3}	1.81×10^{-3}	53.21
P_2 (m^{-1})	0.0	-21.494×10^{-3}	1.58×10^{-3}	0.19×10^{3}
s_x (µm/px)	3.45	3.450	0.25×10^{-3}	
c_x (px)	1224.0	1214.214	0.26	
c_y (px)	1024.0	1020.814	0.22	

Table 6
Evaluation results of the two camera-guided robot calibration experiments.

Evaluation metric	Experiment 1 Uncalibrated robot	Experiment 2 Proposed calibration
$\bar{E}_{\mathrm{RMS}}$ (px)	23.37	3.54
$\bar{E}_{\mathrm{T}}$ (mm)	1.14	0.25
$\bar{E}_{\mathrm{R}}$ (°)	0.22	0.14

redundancy $r = 108941$ in this example. A significance level of $\alpha = 0.01$ yields $F = 6.6$, a significance level of $\alpha = 0.001$ yields $F = 10.8$, for instance. The last column of Table 5 shows the values of the test criterion. They all exceed the threshold values by far. Even the smallest test value of 53.213 (P_1) corresponds to a significance level $\alpha = 3.0 \times 10^{-13}$. Consequently, all tested parameters are highly significant.

8. Conclusions and future work

We have proposed two novel photogrammetry-based robot calibration methods for robots that are equipped with a camera or a 3D sensor. They are capable of calibrating the entire system, i.e., the robot kinematics, the hand–eye transformations, and, if used, the interior orientation of the camera, in an end-to-end manner by modeling the entire chain of transformations from world to robot to imaging sensor. The camera model used by our approach is capable of modeling a large range of lens distortions that occur for lenses that are typically used in machine vision applications.

In contrast to most existing approaches, the optimization in each of the proposed methods uses a geometrically meaningful photogrammetric error measure. This makes the approaches invariant to the parameterization of the model and, in particular, to the choice of units of the model parameters. A further unique feature of our approaches is the use of the Gauss–Markov and Gauss–Helmert models for the optimization, two well-established methods in photogrammetry. They include the use of a stochastic model for all parameters. This allows our approaches to return estimates of the standard deviations of the optimized robot model parameters, for example. Our experiments on synthetic data have shown that these estimates are meaningful, i.e., they closely correspond to the actual mean errors obtained in the experiments.

Furthermore, we have proposed novel evaluation procedures that are relevant in real-world applications and do not require ground truth values. We have validated the evaluation procedures based on experiments on synthetic data. Experiments on real data have shown that the results with our approaches increase the accuracy of the robots we used significantly.

Although we have not tested this, we believe our method will also work for serial robots with more than six axes. However, the cases of serial robots with fewer than six axes (e.g., SCARA robots) and parallel robots seem to be more interesting from a research point of view. They should be examined in more detail in future work. Furthermore, the extension of our approach to sheet-of-light senors in which the laser plane is static and must be moved by the robot should be investigated in the future. This would require the synchronization of the image acquisition with the robot movement. Also, the influence of the distribution of the robot poses on the accuracy of the calibration result would be interesting to investigate. In this context, the computation of optimum robot poses for our calibration approach based on the nominal kinematic parameters would be a promising future research direction.

CRediT authorship contribution statement

Markus Ulrich: Writing – review & editing, Writing – original draft, Validation, Software, Methodology, Investigation, Conceptualization. **Carsten Steger:** Writing – review & editing, Writing – original draft, Validation, Software, Methodology, Investigation, Conceptualization. **Florian Butsch:** Writing – original draft, Validation, Software, Methodology, Investigation. **Maurice Liebe:** Writing – original draft, Validation, Software, Methodology, Investigation.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

Balanji, H.M., Turgut, A.E., Tunc, L.T., 2022. A novel vision-based calibration framework for industrial robotic manipulators. Robot. Comput.-Integr. Manuf. 73, 102248.

Boby, R.A., 2022. Kinematic identification of industrial robot using end-effector mounted monocular camera bypassing measurement of 3-D pose. IEEE/ASME Trans. Mechatronics 27 (1), 383–394.

Boby, R.A., Klimchik, A., 2021. Combination of geometric and parametric approaches for kinematic identification of an industrial robot. Robot. Comput.-Integr. Manuf. 71, 102142.

Brown, D.C., 1971. Close-range camera calibration. Photogram. Eng. 37 (8), 855–866.

Chang, C., Liu, J., Ni, Z., Qi, R., 2018. An improved kinematic calibration method for serial manipulators based on POE formula. Robotica 36 (8), 1244–1262.

Chen, G., Wang, H., Lin, Z., 2014. Determination of the identifiable parameters in robot calibration based on the POE formula. IEEE Trans. Robot. 30 (5), 1066–1077.

Chen, I.-M., Yang, G., Tan, C.T., Yeo, S.H., 2001. Local POE model for robot kinematic calibration. Mech. Mach. Theory 36 (11–12), 1215–1239.

Daniilidis, K., 1999. Hand-eye calibration using dual quaternions. Int. J. Robot. Res. 18 (3), 286–298.

Denavit, J., Hartenberg, R.S., 1955. A kinematic notation for lower-pair mechanisms based on matrices. J. Appl. Mech. 22 (2), 215–221.

Drost, B., 2016. Point Cloud Computing for Rigid and Deformable 3D Object Recognition (Dissertation). Fakultät für Informatik, Technische Universität München.

Drost, B., Ulrich, M., Navab, N., Ilic, S., 2010. Model globally, match locally: Efficient and robust 3D object recognition. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 998–1005.

Du, G., Zhang, P., 2013. Online robot calibration based on vision measurement. Robot. Comput.-Integr. Manuf. 29 (6), 484–492.

Filion, A., Joubair, A., Tahan, A.S., Bonev, I.A., 2018. Robot calibration using a portable photogrammetry system. Robot. Comput.–Integr. Manuf. 49, 77–87.

Förstner, W., Wrobel, B.P., 2016. Photogrammetric Computer Vision: Statistics, Geometry, Orientation and Reconstruction. Springer International Publishing, Cham.

Grün, A., 1978. Accuracy, reliability and statistics in close-range photogrammetry. In: International Archives of Photogrammetry. Vol. 22.

Hartley, R., Zisserman, A., 2003. Multiple View Geometry in Computer Vision, second ed. Cambridge University Press, Cambridge.

Hayat, A.A., Boby, R.A., Saha, S.K., 2019. A geometric approach for kinematic identification of an industrial robot using a monocular camera. Robot. Comput. Integr. Manuf. 57, 329–346.

Hayati, S., Mirmirani, M., 1985. Improving the absolute positioning accuracy of robot manipulators. J. Robot. Syst. 2 (4), 397–413.

He, R., Zhao, Y., Yang, S., Yang, S., 2010. Kinematic-parameter identification for serial-robot calibration based on POE formula. IEEE Trans. Robot. 26 (3), 411–423.

Ito, A., Li, J., Maeda, Y., 2020. SLAM-integrated kinematic calibration using checkerboard patterns. In: IEEE/SICE International Symposium on System Integration. pp. 551–556.

Kuipers, J.B., 1999. Quaternions and Rotation Sequences: A Primer with Applications to Orbits, Aerospace, and Virtual Reality. Princeton University Press, Princeton, NJ.

Lenz, R., Fritsch, D., 1990. Accuracy of videometry with CCD sensors. ISPRS J. Photogramm. Remote Sens. 45 (2), 90–110.

Li, J., Ito, A., Yaguchi, H., Maeda, Y., 2019. Simultaneous kinematic calibration, localization, and mapping (SKCLAM) for industrial robot manipulators. Adv. Robot. 33 (23), 1225–1234.

Li, C., Wu, Y., Li, Z., 2014. Identifiability and improvement of adjoint error approach for serial robot calibration. In: International Conference on Robotics and Automation. pp. 1361–1366.

Li, C., Wu, Y., Löwe, H., Li, Z., 2016. POE-based robot kinematic calibration using axis configuration space and the adjoint error model. IEEE Trans. Robot. 32 (5), 1264–1279.

Lipkin, H., 2005. A note on Denavit-Hartenberg notation in robotics. In: International Design Engineering Technical Conferences & Computers and Information in Engineering Conference. pp. 921–926.

Liu, Z., Liu, X., Cao, Z., Gong, X., Tan, M., Yu, J., 2023. High precision calibration for three-dimensional vision-guided robot system. IEEE Trans. Ind. Electron. 70 (1), 624–634.

Liu, Y., Yuan, P., Chen, D., Su, F., Xue, L., 2017. Simultaneous calibration of hand-eye relationship, robot-world relationship and robot geometric parameters with stereo vision. In: Sun, F., Liu, H., Hu, D. (Eds.), Third International Conference on Cognitive Systems and Signal Processing. In: Communications in Computer and Information Science, vol. 710, Springer Verlag, Singapore, pp. 462–475.

Liu, H., Zhu, W., Dong, H., Ke, Y., 2018. An improved kinematic model for serial robot calibration based on local POE formula using position measurement. Ind. Robot: Int. J. 45 (5), 573–584.

Lou, Y., Chen, T., Wu, Y., Li, Z., Jiang, S., 2009. Improved and modified geometric formulation of POE based kinematic calibration of serial robots. In: International Conference on Intelligent Robots and Systems. pp. 5261–5266.

Luhmann, T., Robson, S., Kyle, S., Boehm, J., 2020. Close-Range Photogrammetry and 3D Imaging, third ed. De Gruyter, Berlin/Boston.
Luo, J., Chen, S., Zhang, C., Chen, C.-Y., Yang, G., 2023. Efficient kinematic calibration for articulated robot based on unit dual quaternion. IEEE Trans. Ind. Inform. 19 (12), 11898–11909.
Luo, R., Gao, W., Huang, Q., Zhang, Y., 2022. An improved minimal error model for the robotic kinematic calibration based on the POE formula. Robotica 40 (5), 1607–1626.
Lynch, K.M., Park, F.C., 2017. Modern Robotics: Mechanics, Planning, and Control. Cambridge University Press, Cambridge.
Ma, L., Bazzoli, P., Sammons, P.M., Landers, R.G., Bristow, D.A., 2018. Modeling and calibration of high-order joint-dependent kinematic errors for industrial robots. Robot. Comput. Integr. Manuf. 50, 153–167.
Madhusudanan, H., Liu, X., Chen, W., Li, D., Du, L., Li, J., Ge, J., Sun, Y., 2020. Automated eye-in-hand robot-3D scanner calibration for low stitching errors. In: International Conference on Robotics and Automation. pp. 8906–8912.
Meng, Y., Zhuang, H., 2007. Autonomous robot calibration using vision technology. Robot. Comput.-Integr. Manuf. 23 (4), 436–446.
Motta, J.M.S.T., de Carvalho, G.C., McMaster, R.S., 2001. Robot calibration using a 3D vision-based measurement system with a single camera. Robot. Comput. Integr. Manuf. 17 (6), 487–497.
Murray, R.M., Li, Z., Sastry, S.S., 1994. A Mathematical Introduction to Robotic Manipulation. CRC Press, Boca Raton, FL.
MVTec Software GmbH, 2024. HALCON Solution Guide III-C — 3D Vision. HALCON Version 24.05.
Niemeier, W., 2008. Ausgleichungsrechnung — Statistische Auswertemethoden, second ed. Walter de Gruyter, Berlin.
Okamura, K., Park, F.C., 1996. Kinematic calibration using the product of exponentials formula. Robotica 14 (4), 415–421.
Pashkevich, A.P., 2001. Computer-aided generation of complete irreducible models for robotic manipulators. In: 3^{e} Conférence Francophone de Modélisation et Simulation. pp. 293–298.
Peters, A., Knoll, A.C., 2024. Robot self-calibration using actuated 3D sensors. J. Field Robotics 41 (2), 327–346.
Rousseau, P., Desrochers, A., Krouglicof, N., 2001. Machine vision system for the automatic identification of robot kinematic parameters. IEEE Trans. Robot. Autom. 17 (6), 972–978.
Schröer, K., Albright, S.L., Grethlein, M., 1997. Complete, minimal and model-continuous kinematic models for robot calibration. Robot. Comput. Integr. Manuf. 13 (1), 73–85.
Steger, C., 2017. A comprehensive and versatile camera model for cameras with tilt lenses. Int. J. Comput. Vis. 123 (2), 121–159.
Steger, C., Ulrich, M., Wiedemann, C., 2018. Machine Vision Algorithms and Applications, second ed. Wiley-VCH, Weinheim.
Sun, T., Lian, B., Yang, S., Song, Y., 2020. Kinematic calibration of serial and parallel robots based on finite and instantaneous screw theory. IEEE Trans. Robot. 36 (3), 816–834.
Sun, T., Liu, C., Lian, B., Wang, P., Song, Y., 2021. Calibration for precision kinematic control of an articulated serial robot. IEEE Trans. Ind. Electron. 68 (7), 6000–6009.
Ulrich, M., Hillemann, M., 2021. Generic hand–eye calibration of uncertain robots. In: IEEE International Conference on Robotics and Automation. pp. 11060–11066.
Ulrich, M., Hillemann, M., 2024. Uncertainty-aware hand–eye calibration. IEEE Trans. Robot. 40, 573–591.
Ulrich, M., Steger, C., 2016. Hand-eye calibration of SCARA robots using dual quaternions. Pattern Recognit. Image Anal. 26 (1), 231–239.
Ulrich, M., Wiedemann, C., Steger, C., 2012. Combining scale-space and similarity-based aspect graphs for fast 3D object recognition. IEEE Trans. Pattern Anal. Mach. Intell. 34 (10), 1902–1914.
Universal Robots, 2024. DH parameters for calculations of kinematics and dynamics. https://www.universal-robots.com/articles/ur/application-installation/dh-parameters-for-calculations-of-kinematics-and-dynamics/. (Accessed 06 June 2024).
Veitschegger, W.K., Wu, C.-H., 1986. Robot accuracy analysis based on kinematics. IEEE J. Robot. Autom. 2 (3), 171–179.
Villalobos, J., Sanchez, I.Y., Martell, F., 2021a. Alternative inverse kinematic solution of the UR5 robotic arm. In: Moreno, H.A., Carrera, I.G., Ramírez-Mendoza, R.A., Baca, J., Banfield, I.A. (Eds.), Advances in Automation and Robotics Research — Proceedings of the 3rd Latin American Congress on Automation and Robotics. In: Lecture Notes in Networks and Systems, vol. 347, Springer Verlag, Berlin, pp. 200–207.
Villalobos, J., Sanchez, I.Y., Martell, F., 2021b. Statistical comparison of Denavit-Hartenberg based inverse kinematic solutions of the UR5 robotic manipulator. In: International Conference on Electrical, Computer, Communications and Mechatronics Engineering.
Wu, L., Crawford, R., Roberts, J., 2017. An analytic approach to converting POE parameters into D–H parameters for serial-link robots. IEEE Robot. Autom. Lett. 2 (4), 2174–2179.
Wu, L., Crawford, R., Roberts, J., 2019. Geometric interpretation of the general POE model for a serial-link robot via conversion into D-H parameterization. In: International Conference on Robotics and Automation. pp. 7360–7366.
Wu, L., Yang, X., Chen, K., Ren, H., 2015. A minimal POE-based model for robotic kinematic calibration with only position measurements. IEEE Trans. Autom. Sci. Eng. 12 (2), 758–763.
Yang, X., Wu, L., Li, J., Chen, K., 2014. A minimal kinematic model for serial robot calibration using POE formula. Robot. Comput.-Integr. Manuf. 30 (3), 326–334.
Yu, C., Xi, J., 2018. Simultaneous and on-line calibration of a robot-based inspecting system. Robot. Comput. Integr. Manuf. 49, 349–360.
Zhang, X., Song, Y., Yang, Y., Pan, H., 2017. Stereo vision based autonomous robot calibration. Robot. Auton. Syst. 93, 43–51.
Zhuang, H., Roth, Z.S., 1996. Camera-Aided Robot Calibration. CRC Press, Boca Raton, FL.
Zhuang, H., Wang, K., Roth, Z.S., 1995. Simultaneous calibration of a robot and a hand-mounted camera. IEEE Trans. Robot. Autom. 11 (5), 649–660.
Zivid AS, 2023. Zivid Two M70 technical specification. version 1.3.