2021 IEEE International Conference on Robotics and Automation (ICRA 2021)
May 31 - June 4, 2021, Xi'an, China

Generic Hand–Eye Calibration of Uncertain Robots

Markus Ulrich and Markus Hillemann

Abstract— **We provide a generic framework for the hand–eye calibration of vision-guided industrial robots. In contrast to traditional methods, we explicitly model the uncertainty of the robot in a statistically sound manner. Albeit the precision of modern industrial robots is high, their absolute accuracy typically is much lower. This uncertainty — if not considered — deteriorates the result of the hand–eye calibration. Our proposed framework not only results in a high accuracy of the computed hand–eye pose but also provides reliable information about the uncertainty of the robot. It further provides corrected robot poses for a convenient and inexpensive robot calibration. Our framework is generic in several respects: It supports the use of a calibration target as well as self-calibration without the need for known 3D points. It optionally allows the simultaneous calibration of the interior camera parameters. The framework is also generic with regard to the robot type, and hence supports articulated as well as SCARA robots, for example. Simulated and real experiments show the validity of the proposed methods.**

I. INTRODUCTION

Hand–eye calibration is essential for applications that use vision-guided robots. It determines the 3D rigid transformation (pose) between the robot and the camera (hand–eye pose). This allows measurements that are performed in the camera coordinate system to be transformed into the robot coordinate system. In bin-picking applications, for example, the pose of an object is determined in the coordinate system of the camera by using 3D object recognition [1], [2], [3], [4]. To grasp the object, the object pose must be transformed into the coordinate system of the robot.

In general, there are two different scenarios of vision-guided robots [5, Chapter 3.12.2]: In the first scenario, the camera is mounted at the robot tool and is moved to different positions by the robot. In the second scenario, the camera is mounted stationary outside the robot without moving with respect to the robot base. The pose that must be determined by hand–eye calibration is the pose of the camera with respect to the robot tool in the first scenario, or the pose of the camera with respect to the robot base in the second scenario. To keep things simple, we will concentrate in the following on the first scenario with a moving camera. Our framework can be applied to the second scenario with a stationary camera in an analogous way.

Traditional approaches for hand–eye calibration assume that the pose of the robot tool with respect to the robot base is known accurately. When discussing the accuracy of robots, it is necessary to distinguish between the pose repeatability and the pose accuracy of the robot [6]: the repeatability is the ability of the robot to move the tool back to the same pose. The accuracy is the ability of the robot to move its tool to a specified pose in 3D space. Modern industrial robots typically offer a very high repeatability, which is in the range of 0.02–0.15 mm [7], [8], [9], [10]. For applications where the robot always moves to the exact same pose, which is trained in advance, a high repeatability is sufficient. However, for robots that are programmed offline and especially for vision-guided robots, obviously a high pose accuracy is important as well. Unfortunately, the pose accuracy of robots often is much lower than the pose repeatability. The position accuracy typically is in the range of 0.1–10.0 mm while the orientation accuracy ranges from about 0.2 deg up to several degrees [8], [9], [10].

Robot calibration allows to improve the accuracy by up to an order of magnitude, in rare cases even up to the level of repeatability [7]. Unfortunately, robot calibration often requires high-precision measurement instruments like lasertrackers [8], and hence is expensive and time consuming. In this paper, we combine all (remaining) errors of the robot under the term uncertainty, which include errors that have an impact on repeatability and on accuracy.

Almost all traditional methods for hand–eye calibration assume error-free robot poses. Our proposed framework explicitly models the uncertainty of the robot in a statistically sound manner resulting in a higher accuracy of the computed hand–eye pose. Furthermore, it provides reliable information about the uncertainty of the robot, which otherwise would require high-precision measurement instruments. Since the framework also provides error-corrected robot poses, it can support a convenient and inexpensive robot calibration.

Most existing approaches for hand–eye calibration require the acquisition of multiple images of a calibration target. A few more flexible and user-friendly solutions avoid the use of calibration targets. Our framework supports both variants, the target-based calibration as well as self-calibration without the need for known 3D points. Furthermore, for both variants, the framework optionally allows the simultaneous calibration of the interior camera parameters.

II. RELATED WORK

Fig. 1 shows the coordinate systems that are relevant for the hand–eye calibration of a robot with a moving camera (for a stationary camera, see [5, Chapter 3.13], for example). Let ${}^{\mathrm{s2}}\mathbf{H}_{\mathrm{s1}}$ denote the 4×4 homogeneous transformation matrix that represents a rigid 3D transformation of points from coordinate system s1 to s2. One of the most common problem formulations of hand–eye calibration is based on closing the following chain of transformations [11]:

$$ {}^{\mathrm{b}}\mathbf{H}_{\mathrm{w}} = {}^{\mathrm{b}}\mathbf{H}_{\mathrm{t}}\,{}^{\mathrm{t}}\mathbf{H}_{\mathrm{c}}\,{}^{\mathrm{c}}\mathbf{H}_{\mathrm{w}} \ , \qquad (1) $$

markus.ulrich@kit.edu, markus.hillemann@kit.edu; Institute of Photogrammetry and Remote Sensing, Karlsruhe Institute of Technology, 76131 Karlsruhe, Germany

978-1-7281-9077-8/21/$31.00 ©2021 IEEE

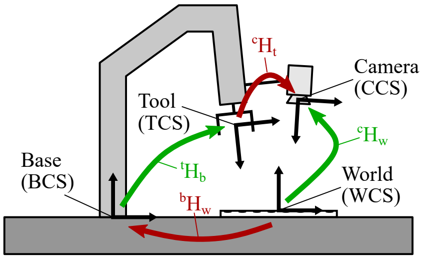

Fig. 1. Relevant 3D coordinate systems, known or observed transformations (green), and unknown (red) transformations for hand–eye calibration where a camera (eye) is rigidly mounted at the robot's end-effector (hand).

with the coordinate systems world (w, WCS), camera (c, CCS), robot tool (t, TCS), and robot base (b, BCS), as well as the unknown poses ${}^{\mathrm{t}}\mathbf{H}_{\mathrm{c}}$ (hand–eye pose) and ${}^{\mathrm{b}}\mathbf{H}_{\mathrm{w}}$. The unknown poses are typically determined by moving the robot to different poses and taking an image of a calibration target, which defines the WCS, at each pose. At each robot pose, ${}^{\mathrm{b}}\mathbf{H}_{\mathrm{t}}$ is queried from the robot controller and ${}^{\mathrm{c}}\mathbf{H}_{\mathrm{w}}$ is determined by PnP algorithms or camera calibration like [12]. Equation 1 is often written as

$$\mathbf{Y} = \mathbf{A}_i \mathbf{X} \mathbf{B}_i \tag{2}$$

with the unknown poses $\mathbf{X}$ and $\mathbf{Y}$ and the observed poses $\mathbf{A}_i$ and $\mathbf{B}_i$ for each of the n robot pose i $(i = 1, \ldots, n)$. Since the essential unknown is $\mathbf{X}$, we can eliminate $\mathbf{Y}$ by taking a pair of different robot poses i and j, yielding $\mathbf{A}_j^{-1}\mathbf{A}_i\mathbf{X} = \mathbf{X}\mathbf{B}_j\mathbf{B}_i^{-1}$. With $\mathbf{A} = \mathbf{A}_j^{-1}\mathbf{A}_i$ and $\mathbf{B} = \mathbf{B}_j\mathbf{B}_i^{-1}$ we obtain

$$\mathbf{A}\mathbf{X} = \mathbf{X}\mathbf{B}, \tag{3}$$

where the movement of the tool is represented by $\mathbf{A}$ and the movement of the camera is represented by $\mathbf{B}$, when moving the robot from pose i to j.

There are several linear approaches that basically solve Eq. (2) or Eq. (3), e.g., [13], [14], [15], [16], [17], [18], [19], [20], [21]. They either determine the rotation and translation parts of the hand–eye pose successively or simultaneously. Latter has the advantage that rotation errors do not propagate and increase translation errors. Approaches that solve Eq. (3) require to select suitable pairs of robot poses to compute $\mathbf{A}$ and $\mathbf{B}$. Criteria for the suitability are proposed in [13], [20], [22]. However, it is still difficult to ensure that the observed information ($\mathbf{A}_i$ and $\mathbf{B}_i$) is optimally exploited. Furthermore, these approaches assume error-free robot poses [13].

Because linear approaches typically minimize an algebraic error, their accuracy is limited. Therefore, they are often used to initialize a subsequent non-linear optimization to obtain a higher accuracy. Most non-linear approaches ([23], [16], [24], [17], [25], [18], [26], [21], [5, Chapter 3.13.5]), which either minimize an algebraic or geometric error, also assume error-free robot poses and often additionally face the problem of how to relatively weight error components of rotation and translation. In [11], a weighted sum of the rotation and translation error parts is minimized where the weights for the error components are statistically derived. While this allows to take errors in the robot poses ${}^{\mathrm{b}}\mathbf{H}_{\mathrm{t}}$ into account, error-free camera poses ${}^{\mathrm{c}}\mathbf{H}_{\mathrm{w}}$ are assumed as input. In [27], rotation and translation parts of Eq. (3) are also solved successively while errors in $\mathbf{A}$ and $\mathbf{B}$ are propagated to $\mathbf{X}$.

Another class of approaches minimizes the reprojection error of 3D world points on a calibration target (e.g., [28], [29]) similar to camera calibration approaches like [12]. The appealing advantages are that it is unnecessary to explicitly estimate the camera pose in each image in pre-processing, no pre-selection of pose pairs is required, and a meaningful geometric error is minimized in the space of the observed erroneous measurements, i.e., the image points. Minimizing the reprojection error also allows to simultaneously estimate the interior camera parameters. However, also these methods assume error-free robot poses. Recently, in [30] a pose graph optimization framework for hand–eye calibration was introduced, in which the errors of the robot poses are minimized in addition to the reprojection error of the points on a calibration target. Experiments showed the benefit of taking the uncertainty of robot poses into account. Unfortunately, no details are given about the stochastic model, the optimization, and how the different error components are weighted with respect to each other.

Self-calibration techniques are used to perform the hand–eye calibration without a calibration target. In [25], unknown 3D points are tracked in the image sequence obtained from a predefined robot motion. The interior camera parameters and the 3D points are simultaneously estimated together with the hand–eye pose. A structure-from-motion (SfM) approach is used in [19], where the unknown scale factor of the SfM result is integrated into the equations. The idea is picked up in [26], where the post-processing step of enforcing the orthogonality of the rotation matrix is avoided by introducing the unknown scale factor into the equations of [16] and [18]. Because these self-calibration techniques minimize algebraic errors, the obtained accuracy is limited.

Our proposed framework combines the mentioned benefits of minimizing a reprojection error, the advantage of stochastically modelling the uncertainty of all measured observations (image points and robot poses), the flexibility of either performing target-based calibration or self-calibration, and the possibility of using known interior camera parameters or estimating them simultaneously.

III. HAND–EYE CALIBRATION OF UNCERTAIN ROBOTS

We will first describe the camera model and the calibration model, i.e., the relation between 3D world points and their projection into the camera that is mounted on the end-effector of a robot, then introduce three alternative least-squares estimation models for hand–eye calibration, and finally suggest how to obtain initial values for the estimation.

A. Camera Model

We represent the camera by the perspective camera model described in [5, Chapter 3.9.1]: When using homogeneous coordinates, a 3D point $\boldsymbol{p}_{\mathrm{w}}$ in the WCS is transformed to a point $\boldsymbol{p}_\mathrm{c}$ in the CCS by

$$\boldsymbol{p}_\mathrm{c} = {}^\mathrm{c}\mathtt{H}_\mathrm{w}\boldsymbol{p}_\mathrm{w} = \begin{pmatrix} \mathtt{R} & \boldsymbol{t} \\ \boldsymbol{0}^\top & 1 \end{pmatrix} \boldsymbol{p}_\mathrm{w} \tag{4}$$

where $\boldsymbol{t} = (t_\mathrm{x}, t_\mathrm{y}, t_\mathrm{z})^\top$ is a translation vector and $\mathtt{R}$ is a rotation matrix parameterized by Euler angles: $\mathtt{R} = \mathtt{R}_x(\alpha)\mathtt{R}_y(\beta)\mathtt{R}_z(\gamma)$. The parameters ($t_\mathrm{x}$,$t_\mathrm{y}$,$t_\mathrm{z}$,$\alpha$,$\beta$,$\gamma$) describe the exterior orientation of the camera. The point $\boldsymbol{p}_\mathrm{c} = (x_\mathrm{c}, y_\mathrm{c}, z_\mathrm{c})^\top$ is then projected into the image plane by

$$\begin{pmatrix} x_\mathrm{u} \\ y_\mathrm{u} \end{pmatrix} = \frac{c}{z_\mathrm{c}} \begin{pmatrix} x_\mathrm{c} \\ y_\mathrm{c} \end{pmatrix} \ , \tag{5}$$

with the principal distance c of the lens. Even if we will concentrate on a perspective lens, other camera models (e.g., for telecentric lenses) can be included in a straightforward way [5, Chapter 3.9]. Next, the undistorted point $(x_\mathrm{u}, y_\mathrm{u})^\top$ is distorted to $(x_\mathrm{d}, y_\mathrm{d})^\top$, e.g., by the division model [31]

$$\begin{pmatrix} x_\mathrm{u} \\ y_\mathrm{u} \end{pmatrix} = \frac{1}{1+\kappa r_\mathrm{d}^2} \begin{pmatrix} x_\mathrm{d} \\ y_\mathrm{d} \end{pmatrix} \ , \tag{6}$$

which models radial distortions by κ, or the polynomial model [32], which models radial and decentering distortions by five parameters. Here, $r_\mathrm{d}^2 = x_\mathrm{d}^2 + y_\mathrm{d}^2$.

Finally, the distorted point $(x_\mathrm{d}, y_\mathrm{d})^\top$ is transformed into the image coordinate system:

$$\boldsymbol{p}_\mathrm{i} = \begin{pmatrix} x_\mathrm{i} \\ y_\mathrm{i} \end{pmatrix} = \begin{pmatrix} x_\mathrm{d}/s_x + c_x \\ y_\mathrm{d}/s_y + c_y \end{pmatrix} \ , \tag{7}$$

where $(c_x, c_y)^\top$ is the principal point and s_x and s_y denote the pixel pitches on the sensor.

The six parameters $\boldsymbol{i} = (c,\ \kappa,\ s_x,\ s_y,\ c_x,\ c_y)^\top$ for the division model (or ten parameters for the polynomial model) describe the interior orientation $\boldsymbol{i}$ of the camera.

B. Calibration Model

For hand–eye calibration, the tool of the robot is moved to n_r different poses and an image is acquired at each pose. For target-based calibration, the calibration target is placed at a fixed position within the workspace of the robot (see Fig. 1). In the self-calibration case, images are acquired from any sufficiently textured object or background scene instead. Let ${}^\mathrm{t}\mathtt{H}_{\mathrm{b},j}$ be the robot pose that is queried from the robot controller at pose j ($j = 1, \dots, n_\mathrm{r}$). Further let the 3D world points (given in the WCS) of the calibration target or in the scene be $\boldsymbol{p}_k$ ($k = 1, \dots, n_\mathrm{w}$) and their 2D projections into the image at robot pose j be $\boldsymbol{p}_{j,k}$. Then, we can describe the projection of a 3D point into an image by:

$$\boldsymbol{p}_{j,k} = \pi({}^\mathrm{c}\mathtt{H}_\mathrm{t}{}^\mathrm{t}\mathtt{H}_{\mathrm{b},j}{}^\mathrm{b}\mathtt{H}_\mathrm{w}\boldsymbol{p}_k, \boldsymbol{i}) \ , \tag{8}$$

where $\pi(\boldsymbol{p}_\mathrm{c}, \boldsymbol{i})$ is the projection of the point $\boldsymbol{p}_\mathrm{c}$, which is given in the CCS, into the image by successive execution of Eqs. (5), the inverse of (6), and (7), by using the parameters of the interior orientation $\boldsymbol{i}$. Finally, we denote the six transformation parameters of the robot poses ${}^\mathrm{t}\mathtt{H}_{\mathrm{b},j}$ by $\boldsymbol{e}_{\mathrm{t},j}$, that of the unknown hand–eye pose ${}^\mathrm{c}\mathtt{H}_\mathrm{t}$ by $\boldsymbol{e}_\mathrm{c}$, and that of the unknown pose ${}^\mathrm{b}\mathtt{H}_\mathrm{w}$ by $\boldsymbol{e}_\mathrm{b}$ (in analogy with ${}^\mathrm{c}\mathtt{H}_\mathrm{w}$ above).

C. Estimation in the Gauss-Markov Model

We distinguish between the functional model and the stochastic model [33]. The functional model describes the relations between the observations and the unknown parameters. In the stochastic model, the observations and the unknowns are treated as random variables with uncertainties, where the uncertainties are described by (co-)variances.

For error-free robot poses, the hand–eye calibration task can be formulated in the Gauss-Markov model [33, Chapter 4.4] because all uncertain observations $\boldsymbol{l}$ can be expressed as a function $\boldsymbol{l} = \boldsymbol{f}(\boldsymbol{x})$ of the unknown parameters $\boldsymbol{x}$, where $\boldsymbol{f}$ describes the functional model and corresponds to Eq. (8).

The vector $\boldsymbol{l}$ contains the measured image points $\boldsymbol{l} = (\boldsymbol{p}_{1,1}^\top$, ..., $\boldsymbol{p}_{1,n_\mathrm{w}}^\top$, ..., $\boldsymbol{p}_{n_\mathrm{r},n_\mathrm{w}}^\top)^\top$. If a 3D point is not visible in a certain image, the respective entry in $\boldsymbol{l}$ is simply omitted. Let n_i denote the sum of measured image points over all images, i.e., the number of observations n_l is $2n_\mathrm{i}$. For each observed image point, we set up two equations of the form (8) yielding $2n_\mathrm{i}$ equations. The vector $\boldsymbol{x}$ depends on the application scenario and is built with the following recipe:

- For all scenarios, $\boldsymbol{x}$ contains the hand–eye pose: $\boldsymbol{x} := \boldsymbol{e}_\mathrm{c}$. Note that for SCARA robots, t_z of the robot pose cannot be determined and is therefore left fixed at 0 [21].
- For target-based calibration, $\boldsymbol{x}$ must be extended by the unknown $\boldsymbol{e}_\mathrm{b}$: $\boldsymbol{x} := (\boldsymbol{x}^\top, \boldsymbol{e}_\mathrm{b}^\top)^\top$. Note that for self-calibration, we can reconstruct the 3D points directly in the BCS. Hence, we exclude $\boldsymbol{e}_\mathrm{b}$ from the unknowns and internally set $\boldsymbol{e}_\mathrm{b} = (0, 0, 0, 0, 0, 0)^\top$.
- If the interior orientation of the camera is unknown and shall be simultaneously estimated, we further extend $\boldsymbol{x} := (\boldsymbol{x}^\top, \boldsymbol{i}^\top)^\top$. Note that s_y is typically excluded from the calibration [5, Chapter 3.9.4.2].
- For self-calibration, $\boldsymbol{x}$ must be further extended by the coordinates of the 3D points: $\boldsymbol{x} := (\boldsymbol{x}^\top, \boldsymbol{p}_1^\top, \dots, \boldsymbol{p}_{n_\mathrm{w}}^\top)^\top$.

Let the final number of unknown variables in $\boldsymbol{x}$ be n_x. Note that the singularity in the Euler angle parameterization is uncritical as the optimum poses are close to the initial poses of $\boldsymbol{e}_\mathrm{c}$ and $\boldsymbol{e}_\mathrm{b}$. Latter are obtained by the dual quaternion based approach of [18] (cf. Sec. III-F). Regardless of this, our framework allows other parameterizations if desired.

The stochastic model specifies the statistical properties of the observation process. Assuming that the image points are uncorrelated and are measured with the same accuracy, we set the $n_\mathrm{l} \times n_\mathrm{l}$ weight coefficient matrix of the observations to the identity: $\mathtt{Q}_{ll} = \mathtt{I}$.

For linearization, we compute the $2n_\mathrm{i} \times n_\mathrm{x}$ Jacobian $\mathtt{A}$ of $\boldsymbol{f}$ at the initial values of the unknowns $\boldsymbol{x}^{(0)}$ (cf. Sec. III-F):

$$\mathtt{A} = \left.\frac{\partial \boldsymbol{f}(\boldsymbol{x})}{\partial \boldsymbol{x}}\right|_{\boldsymbol{x}=\boldsymbol{x}^{(0)}} \ . \tag{9}$$

With $\Delta \boldsymbol{l} = \boldsymbol{l} - \boldsymbol{f}(\boldsymbol{x}^{(0)})$ and the weight matrix $\mathtt{P}_{ll} = \mathtt{Q}_{ll}^{-1}$, we can calculate the corrections for the unknowns by solving

$$\mathtt{A}^\top \mathtt{P}_{ll} \mathtt{A} \Delta\hat{\boldsymbol{x}} = \mathtt{A}^\top \mathtt{P}_{ll} \Delta \boldsymbol{l} \tag{10}$$

for $\Delta\hat{\boldsymbol{x}}$ by sparse Cholesky decomposition, for example[1].

[1] In statistics, the hat operator ˆ denotes an estimated or fitted value.

Then, $\hat{\boldsymbol{x}}^{(1)} = \boldsymbol{x}^{(0)} + \Delta\hat{\boldsymbol{x}}$, and Eqs. (9) and (10) are repeatedly applied until convergence. This corresponds to minimizing the reprojection error. After convergence, the covariance matrix of the original observations is obtained by $\mathtt{C}_{ll} = \hat{\sigma}_0^2 \mathtt{Q}_{ll}$, with the variance factor $\hat{\sigma}_0^2 = \hat{\boldsymbol{v}}^\top \mathtt{P}_{ll} \hat{\boldsymbol{v}}/r$, the residuals $\hat{\boldsymbol{v}} = \mathtt{A}\Delta\hat{\boldsymbol{x}} - \Delta\boldsymbol{l}$, and the redundancy $r = n_\mathrm{l} - n_\mathrm{x}$. 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 $\mathtt{C}_{\hat{x}\hat{x}} = \hat{\sigma}_0^2(\mathtt{A}^\top \mathtt{P}_{ll}\mathtt{A})^{-1}$.

Note that this approach corresponds to the rp_1 method (fixed interior orientation) or rp_2 (simultaneously estimated interior orientation) method in [29] for the special case of target-based calibration and $\mathtt{Q}_{ll} = \mathtt{I}$.

D. Estimation in the Gauss-Helmert Model

To consider uncertain robot poses, they must be introduced as observations in addition to the image coordinates. Hence, the observations cannot be expressed as a function of the unknowns any longer. This requires performing the parameter estimation in the Gauss-Helmert model [33, Chapter 4.8], and our functional model becomes $\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{l}) = 0$.

In this model, the vector $\boldsymbol{l}$ contains the observed image points and the robot poses $\boldsymbol{l} = (\boldsymbol{p}_{1,1}^\top, \ldots, \boldsymbol{p}_{1,n_\mathrm{w}}^\top, \ldots, \boldsymbol{p}_{n_\mathrm{r},n_\mathrm{w}}^\top, \boldsymbol{e}_{\mathrm{t},1}^\top, \ldots, \boldsymbol{e}_{\mathrm{t},n_\mathrm{r}}^\top)^\top$. Therefore, the number of observation equations is still $2n_\mathrm{i}$. However, the number of observations now is $n_\mathrm{l} = 2n_\mathrm{i} + 6n_\mathrm{r}$. The vector $\boldsymbol{x}$ of unknowns is identical to that in the Gauss-Markov model.

In comparison to the Gauss-Markov model, the stochastic model must also consider the uncertainties of the robot poses. Tests on real systems showed that the errors in the robot poses are zero-mean Gaussian distributed [11]. But even if assuming uncorrelated observations, we still would have to know the relative accuracy between the three observation groups (image coordinates, Euler angles of the robot poses, translation components of the robot poses) to set up $\mathtt{Q}_{ll}$. Therefore, we initialize $\mathtt{Q}_{ll}$ with reasonable values and use the concept of variance components [33] to estimate the true variances of the observations (see below). We could also initialize $\mathtt{Q}_{ll} = \mathtt{I}$, but then the number of iterations to estimate the variance components would be unnecessarily high. Therefore, for a higher efficiency, in our implementations we initially set the standard deviations of the image points to $\sigma_\mathrm{i} = 0.1$ pixel, of the Euler angles to $\sigma_\mathrm{a} = 0.1\,\mathrm{deg}$, and of the translation components to $\sigma_\mathrm{t} = 1\,\mathrm{mm}$. Then, $\mathtt{Q}_{ll} = \mathrm{diag}(\mathrm{rep}(\sigma_\mathrm{i}^2, 2n_\mathrm{i}), \mathrm{rep}([\sigma_\mathrm{a}^2, \sigma_\mathrm{a}^2, \sigma_\mathrm{a}^2, \sigma_\mathrm{t}^2, \sigma_\mathrm{t}^2, \sigma_\mathrm{t}^2], n_\mathrm{r}))$, where the function $\mathrm{rep}(\boldsymbol{y}, n)$ generates a vector containing n copies of $\boldsymbol{y}$.

The Jacobian $\mathtt{A}$ is the same as in the Gauss-Markov model. In the Gauss-Helmert model, we must linearize $\boldsymbol{f}$ also with respect to the observations yielding the $2n_\mathrm{i} \times n_\mathrm{l}$ Jacobian $\mathtt{B}$.

$$\mathtt{B} = \left. \frac{\partial \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{l})}{\partial \boldsymbol{l}} \right|_{\boldsymbol{x}=\boldsymbol{x}^{(0)}} . \tag{11}$$

Note that the upper left $2n_\mathrm{i} \times 2n_\mathrm{i}$ submatrix of $\mathtt{B}$ is a diagonal matrix with all diagonal elements set to -1. With $\boldsymbol{w} = \boldsymbol{f}(\boldsymbol{x}^{(0)}, \boldsymbol{l})$, we can calculate the corrections for the unknowns by solving

$$\mathtt{A}^\top(\mathtt{B}\mathtt{Q}_{ll}\mathtt{B}^\top)^{-1}\mathtt{A}\Delta\hat{\boldsymbol{x}} = -\mathtt{A}^\top(\mathtt{B}\mathtt{Q}_{ll}\mathtt{B}^\top)^{-1}\boldsymbol{w} \tag{12}$$

for $\Delta\hat{\boldsymbol{x}}$ [33]. After convergence, we estimate a variance component for each of the three observation groups. This yields corrected values for σ_i, σ_a, and σ_t, and thus a corrected $\mathtt{Q}_{ll}$ (for details about variance component estimation see [33, Chapter 4.2.4]). Then, the whole estimation process (which is an iterative process by itself) is repeated until the variance components converge (typically after 3 to 5 iterations). The covariance matrix of the observations is obtained by $\mathtt{C}_{ll} = \hat{\sigma}_0^2 \mathtt{Q}_{ll}$ with the redundancy $r = 2n_\mathrm{i} - n_\mathrm{x}$. The covariance matrix of the estimated unknowns is obtained by $\mathtt{C}_{\hat{x}\hat{x}} = \hat{\sigma}_0^2(\mathtt{A}^\top(\mathtt{B}\mathtt{Q}_{ll}\mathtt{B}^\top)^{-1}\mathtt{A})^{-1}$.

In essence, this model allows to simultaneously minimize the reprojection error and the error in the robot poses by automatically weighting the error terms with respect to each other based on the computed statistics.

E. Estimation in the Gauss-Markov Model with Fictitious Observations

Because of the expensive matrix computations in the Gauss-Helmert model, we also implemented a more efficient variant of the Gauss-Markov model that is equivalent to the Gauss-Helmert model ([34], [35]) by using the concept of fictitious observations for the robot poses.

The key idea is to introduce the uncertain robot poses simultaneously as observations and as unknowns. The first part of the functional model is still $\boldsymbol{l} = \boldsymbol{f}(\boldsymbol{x})$. In contrast to Sec. III-C, however, we add 6 additional observation equations of the form $\boldsymbol{e}_{\mathrm{t},j} = \boldsymbol{f}_2(\boldsymbol{e}_{\mathrm{t},j})$ for each robot pose j, where $\boldsymbol{f}_2$ is the identity in our case. This results in $2n_\mathrm{i}+6n_\mathrm{r}$ equations. Hence, the vector $\boldsymbol{l}$ contains the observed image points and the robot poses $\boldsymbol{l} = (\boldsymbol{p}_{1,1}^\top, \ldots, \boldsymbol{p}_{1,n_\mathrm{w}}^\top, \ldots, \boldsymbol{p}_{n_\mathrm{r},n_\mathrm{w}}^\top, \boldsymbol{e}_{\mathrm{t},1}^\top, \ldots, \boldsymbol{e}_{\mathrm{t},n_\mathrm{r}}^\top)^\top$ as in Sec. III-D. Simultaneously, we also introduce the robot poses as unknowns by extending $\boldsymbol{x} := (\boldsymbol{x}^\top, \boldsymbol{e}_{\mathrm{t},j}^\top, \ldots, \boldsymbol{e}_{\mathrm{t},n_\mathrm{r}}^\top)^\top$.

Because $\boldsymbol{l}$ is identical to that in the Gauss-Helmert model, we apply the same stochastic model and initialize $\mathtt{Q}_{ll} = \mathrm{diag}(\mathrm{rep}(\sigma_\mathrm{i}^2, 2n_\mathrm{i}), \mathrm{rep}([\sigma_\mathrm{a}^2, \sigma_\mathrm{a}^2, \sigma_\mathrm{a}^2, \sigma_\mathrm{t}^2, \sigma_\mathrm{t}^2, \sigma_\mathrm{t}^2], n_\mathrm{r}))$.

The computation of the $(2n_\mathrm{i} + 6n_\mathrm{r}) \times n_\mathrm{x}$ Jacobian $\mathtt{A}$ and of the corrections $\Delta\hat{\boldsymbol{x}}$ is done by Eqs. (9) and (10). Note that in this case the lower left part of $\mathtt{A}$ contains zeros and the lower right part $\mathtt{I}_{6n_\mathrm{r} \times 6n_\mathrm{r}}$. After convergence, the variance components are estimated as in Sec. III-D. Finally, $\mathtt{C}_{ll} = \hat{\sigma}_0^2 \mathtt{Q}_{ll}$, $\hat{\boldsymbol{l}}$, and $\mathtt{C}_{\hat{x}\hat{x}}$ are computed as described in Sec. III-C.

F. Initialization of the Unknowns

We propose the following initialization for $\boldsymbol{x}$:

- Initial values for the hand–eye pose $\boldsymbol{e}_\mathrm{c}$ and for $\boldsymbol{e}_\mathrm{b}$ are obtained from the linear approach of [18].
- Initial values for the interior orientation are obtained from the data sheets of the camera (s_x and s_y) and the lens (c). The principal point $(c_x, c_y)^\top$ is set to the image center and the distortion coefficients are set to 0.
- For self-calibration, we apply the SfM pipeline COLMAP ([36], [37]) to the acquired images. It returns

the camera parameters $\boldsymbol{i}$, the reconstructed 3D points $\boldsymbol{p}_k$, and for each image j the extracted 2D keypoints $\boldsymbol{p}_{j,k}$ and the exterior orientation ${}^{\mathrm{c}}\mathtt{H}_{\mathrm{w},j}$.

In the self-calibration case, we fix the inherent unknown scale factor of SfM in the initial values for $\boldsymbol{p}_k$ and ${}^{\mathrm{c}}\mathtt{H}_{\mathrm{w},j}$: For all pairs of robot poses $j1$ and $j2$, we compute the camera movement ${}^{\mathrm{c}2}\mathtt{H}_{\mathrm{c}1} = {}^{\mathrm{c}}\mathtt{H}_{\mathrm{w},j2}{}^{\mathrm{c}}\mathtt{H}_{\mathrm{w},j1}^{-1}$ and the tool movement ${}^{\mathrm{t}2}\mathtt{H}_{\mathrm{t}1} = {}^{\mathrm{t}}\mathtt{H}_{\mathrm{b},j2}{}^{\mathrm{t}}\mathtt{H}_{\mathrm{b},j1}^{-1}$ and convert both rigid 3D transformations to screw parameters [18]. From the screw congruency theorem [15] we known that the screw translation parameter d_{t} of the tool movement is identical to the screw translation parameter d_{c} of the camera movement. Therefore, the ratio $d_{\mathrm{t}}/d_{\mathrm{c}}$ reflects the unknown scale factor of the SfM. For a higher robustness, the median over the ratios of all pairs is computed while ignoring ratios for which d_{t} or d_{c} is below a noise threshold. Finally, the scale factor is used to correct the initial values for $\boldsymbol{p}_k$ and ${}^{\mathrm{c}}\mathtt{H}_{\mathrm{w},j}$.

IV. RESULTS

A. Validation on Simulated Data

We simulated an uncertain robot by adding zero-mean Gaussian noise to its robot poses ${}^{\mathrm{t}}\mathtt{H}_{\mathrm{b},j}$ with a standard deviation of 1 mm in the translation components and of 0.1 deg in the orientation components. To the image points, we applied noise with a standard deviation of 0.1 pixels.

For calibration, we simulated a planar calibration target with $n_{\mathrm{w}} = 40$ points and a diameter of about 1 m and placed it in the workspace of the robot. We set the hand–eye pose ${}^{\mathrm{c}}\mathtt{H}_{\mathrm{t}}$ to different values with about 10 cm distance and different relative rotations. For the camera of resolution 1280×1024 we assumed the division model and set $c = 8\,\mathrm{mm}$, $\kappa = 2000\,\mathrm{m}^{-2}$, $s_x = 5.21\,\mu\mathrm{m}$, $s_y = 5.2\,\mu\mathrm{m}$, and (c_x, c_y) = (645,502) pixels. We generated 40 random robot poses ${}^{\mathrm{t}}\mathtt{H}_{\mathrm{b},j}$ in its 1 m^3 workspace such that at least 90% of the calibration points were visible in the image.

First, we calibrated the interior orientation of the camera by using the acquired images and the approach of [5, Chapter 3.9.4]. Then, for hand–eye calibration we applied the linear approach of [18] (linear), the non-linear approach of [5, Chapter 3.13.5] (non-linear), the approach based on the Gauss-Markov model of Sec. III-C (GM), which is equivalent to the rp_1 method in [29], and the Gauss-Markov model with fictitious observations of Sec. III-E (GMF). Because in all experiments the results of the Gauss-Helmert model of Sec. III-D were identical to that of GMF (up to numerical inaccuracies), we omit the results. We repeated each experiment 30 times and display the mean values in the following.

Fig. 2 shows the translation and rotation error of the computed hand–eye poses ${}^{\mathrm{c}}\mathtt{H}_{\mathrm{t}}$. Let $\mathtt{H}$ represent the computed pose and $\mathtt{H}_{\mathrm{gt}}$ the ground truth pose. We compute the translation error as the length of the translation part of $\mathtt{H} - \mathtt{H}_{\mathrm{gt}}$. For the rotation error, we compute the magnitude of the Rodrigues rotation of $\mathtt{H}\mathtt{H}_{\mathrm{gt}}^{-1}$. As expected, the errors of all approaches generally decrease when increasing the number of robot poses. GMF performs best in all cases. Furthermore, the advantage of modeling the uncertainty of the robot can be clearly seen from the difference between GM and GMF. Note that the translation error might even exceed the uncertainty of the robot especially for fewer robot poses.

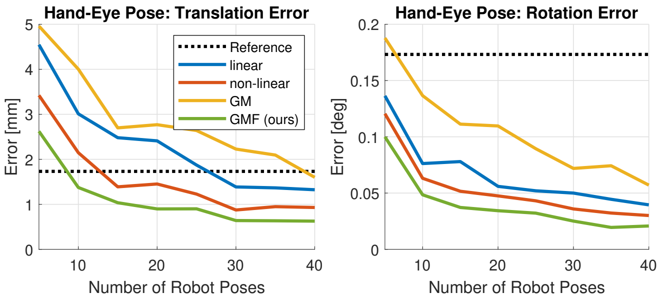

Fig. 2. Translation and rotation error of the hand–eye pose for different number of robot poses n_{r} and different approaches: linear [18], non-linear [5], GM (equivalent to [29]), GMF (ours). 'Reference' visualizes the uncertainty of the simulated robot.

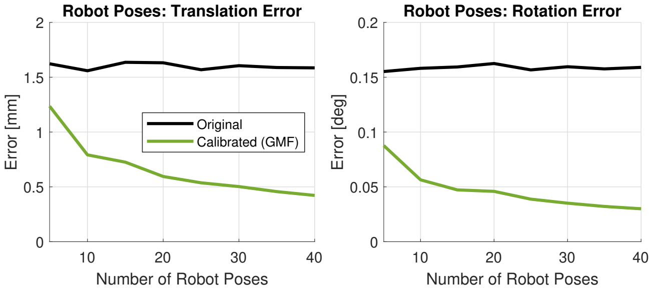

Fig. 3. Translation and rotation error of the robot poses ${}^{\mathrm{t}}\mathtt{H}_{\mathrm{b},j}$ before (black) and after (green) GMF calibration.

Fig. 3 shows the translation and rotation error of the robot poses before and after GMF calibration (the calibrated robot poses are contained in $\hat{\boldsymbol{l}}$). GMF is able to decrease the errors in the robot poses to about 20–25%. Thus, GMF not only returns an accurate hand–eye pose but also allows a convenient vision-based robot calibration.

To analyze the benefit of estimating the variance components, we fixed the number of robot poses to 40 and set the standard deviation of the rotation components of the simulated robot poses to 0.3 deg. Then, we varied the standard deviation of the translation components of the robot from 0 to 6 mm and repeated each experiment 30 times. The mean error of the hand–eye pose is show in Fig. 4. As before, we initialized $\mathbf{Q}_{ll}$ with $\sigma_{\mathrm{a}} = 0.1\,\mathrm{deg}$ and $\sigma_{\mathrm{t}} = 1\,\mathrm{mm}$. The positive effect of estimating the variance components on the accuracy is obvious and increases the more the assumed ratio deviates from the true ratio of standard deviations.

Another advantage of estimating the variance components is that a meaningful accuracy of the robot is returned in $\mathbf{C}_{ll}$, which otherwise would require costly robot calibration. On average, the accuracy (i.e., the standard deviations) of the robot was computed with an error of 61.1% in translation and 53.3% in rotation without estimating variance components, and with an error of only 0.8% and 1.0% when estimating variance components. The mean standard deviation of the image points was estimated as 0.14 pixels without, and correctly as 0.10 pixels with estimating variance components.

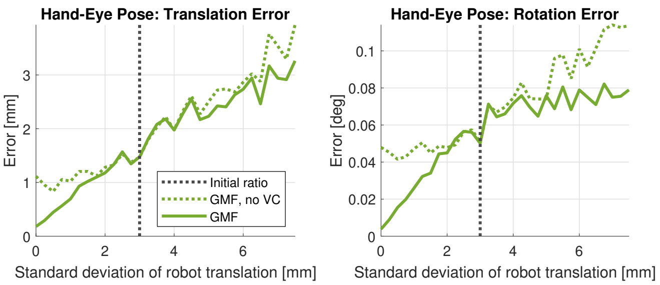

Fig. 4. Translation and rotation error of the hand–eye pose with (GMF) and without (GMF, no VC) estimating variance components. The standard deviation of the rotation components of the simulated robot was set to 0.3 deg and the standard deviation of the translation components was varied. The translation standard deviation that corresponds to the initial ratio $0.3 \cdot \sigma_\mathrm{t}/\sigma_\mathrm{a}$ mm is indicated by the vertical dotted line.

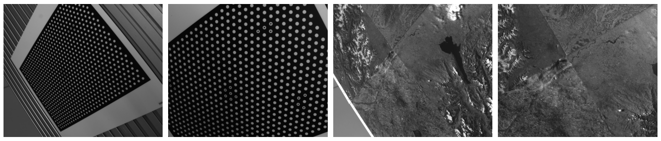

Fig. 5. Two example calibration images for the target-based hand–eye calibration (left) and for hand–eye self-calibration (right).

B. Experiments on Real Data

For our experiments, we attached a 2448 × 2048 IDS U3-3280SE camera with an 8 mm lens at the end-effector of a 6-axis UR3e robot arm (0.5 m working radius). To test the target-based calibration, we put a planar calibration target with known control points in the workspace of the robot and acquired images at 25 different robot poses (Fig. 5, left). To test the self-calibration capabilities of our model, we replaced the calibration target by a planar print and acquired 25 images by using the same robot poses (Fig. 5, right).

The computation time of the optimization in the Gauss-Markov model with fictitious observations was approximately 200 ms on an Intel Core i7-8565U with 1.8 GHz.

Table I displays the reprojection root mean squared error (RRMSE) and the reconstruction accuracy error (RAE), which is the average squared Euclidean error of the reconstructed calibration points, as proposed in [29]. 'fixed camera' means that camera parameters are calibrated separately by using [5, Chapter 3.9.4] and excluded from the hand–eye calibration. 'orig. poses' means that the errors are calculated with the original robot poses instead of the calibrated ones. 'self-calibration' means that no calibration target was used (obviously, no RAE can be computed in this case). For the calibrated robot poses, GMF returns the smallest errors for target-based calibration and for self-calibration.

It is also interesting to have a look at the standard deviations of the image points, the robot translations, and the robot rotations, which are returned by GMF. They are 0.23 pixels, 0.093 mm, and 0.029 deg in the target-based calibration and 0.32 pixels, 0.097 mm, and 0.025 deg in the self-calibration case. Thus, the robot uncertainty was estimated consistently. Furthermore, the image points of the calibration target are extracted with a slightly higher accuracy compared to the feature points in the prints, which is plausible. The resulting hand–eye poses for target-based calibration and self-calibration are shown in Table II.

TABLE I

RRMSE AND RAE ERRORS ON THE UR3E DATA

Method	RRMSE	RAE
	[pixel]	[mm^2]
linear [18]	0.708	0.00462
non-linear [5]	0.608	0.00212
GM, fixed camera	0.567	0.00194
GM	0.545	0.00212
GMF, orig. poses, fixed camera	0.784	0.02190
GMF, orig. poses	0.771	0.02103
GMF, fixed camera	0.196	0.00073
GMF	0.193	0.00073
GMF, self-calibration	0.178	-

TABLE II

RESULTING HAND–EYE POSES

Method	t_x	t_y	t_z	α	β	γ
	[mm]	[mm]	[mm]	[deg]	[deg]	[deg]
GMF	63.64	135.21	45.67	2.26	25.39	176.35
GMF, self-calib	63.79	135.18	45.76	2.24	25.35	176.35

We also applied our approach to four datasets published in [29]. The results are shown in Table III. For calibrated robot poses, GMF returns the smallest errors in almost all cases. When taking the original robot poses (GMF, orig. poses), the error is significantly higher.

TABLE III

RRMSE AND RAE ERRORS ON THE DATASET [29] (RESULTS OF [29] AND [30] ARE TAKEN FROM [30])

	Dataset 1		Dataset 2		Dataset 3		Dataset 4	
	RRMSE	RAE	RRMSE	RAE	RRMSE	RAE	RRMSE	RAE
Method	[pixel]	[mm^2]	[pixel]	[mm^2]	[pixel]	[mm^2]	[pixel]	[mm^2]
linear [18]	1.895	51.023	3.877	2.623	12.504	725.852	2.627	38.219
non-linear [5]	1.411	5.096	3.578	0.070	3.624	9.356	1.524	2.082
c2 [29]	1.672	5.820	3.960	0.086	3.553	1.927	2.388	1.715
rp1 [29]	1.569	0.238	3.741	0.030	2.890	1.454	1.603	1.437
rp2 [29]	1.363	0.524	3.181	0.450	2.680	1.588	1.505	1.481
Pose Graph [30]	1.562	0.214	3.751	0.038	2.879	1.337	1.619	1.380
GMF, orig. poses	1.567	7.715	3.835	3.519	11.744	584.722	1.661	9.090
GMF	0.076	0.113	0.176	0.031	0.479	1.188	0.389	1.380

V. CONCLUSIONS

Explicitly modelling the uncertainty of a robot is beneficial for robot hand–eye calibration. It improves the accuracy, returns calibrated robot poses, and gives information about the uncertainty of the robot. The proposed parameter estimation in the Gauss-Markov model with fictitious observations combined with the estimation of variance components allows a statistically sound representation of the problem. Different scenarios of hand–eye calibration (e.g., target-based calibration, self-calibration, calibration of articulated or SCARA robots, unknown or known interior orientation) can be easily represented by adding or removing the respective parameters to or from the parameter vectors. This allows a huge number of applications to benefit from our approach.

ACKNOWLEDGMENT

We thank the MVTec Software GmbH, where the first ideas on this research topic were developed and discussed.

References

[1] A. Hofhauser, C. Steger, and N. Navab, "Perspective planar shape matching," in *Image Processing: Machine Vision Applications II*, ser. Proc. SPIE 7251, K. S. Niel and D. Fofi, Eds., 2009.

[2] S. Hinterstoisser, V. Lepetit, S. Ilic, S. Holzer, G. Bradski, K. Konolige, and N. Navab, "Model based training, detection and pose estimation of texture-less 3d objects in heavily cluttered scenes," in *Computer Vision – ACCV 2012*, K. M. Lee, Y. Matsushita, J. M. Rehg, and Z. Hu, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 548–562.

[3] M. Ulrich, C. Wiedemann, and C. Steger, "Combining scale-space and similarity-based aspect graphs for fast 3D object recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 10, pp. 1902–1914, Oct. 2012.

[4] Y. Labbe, J. Carpentier, M. Aubry, and J. Sivic, "CosyPose: Consistent multi-view multi-object 6D pose estimation," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020.

[5] C. Steger, M. Ulrich, and C. Wiedemann, *Machine Vision Algorithms and Applications*, 2nd ed. Weinheim: Wiley-VCH, 2018.

[6] ISO 9283:1998, "Manipulating industrial robots — performance criteria and related test methods," 1998.

[7] P. Shiakolas, K. Conrad, and T. Yih, "On the accuracy, repeatability, and degree of influence of kinematics parameters for industrial robots," *International Journal of Modelling and Simulation*, vol. 22, no. 4, pp. 245–254, 2002.

[8] M. Abderrahim, A. Khamis, S. Garrido, and L. Moreno, "Accuracy and calibration issues of industrial manipulators," in *Industrial Robotics: Programming, Simulation and Application*. IntechOpen, 2006, pp. 131–146.

[9] M. Morozov, J. Riise, R. Summan, S. G. Pierce, C. Mineo, C. N. MacLeod, and R. H. Brown, "Assessing the accuracy of industrial robots through metrology for the enhancement of automated non-destructive testing," in *2016 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI)*, 2016, pp. 335–340.

[10] M. Płaczek and Ł. Piszczek, "Testing of an industrial robot's accuracy and repeatability in off and online environments," *Eksploatacja i Niezawodnosc — Maintenance and Reliability*, vol. 20, no. 3, pp. 455–464, 2018.

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

[12] Z. Zhang, "A flexible new technique for camera calibration," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 11, pp. 1330–1334, Nov. 2000.

[13] R. Y. Tsai and R. K. Lenz, "A new technique for fully autonomous and efficient 3D robotics hand/eye calibration," *IEEE Transactions on Robotics and Automation*, vol. 5, no. 3, pp. 345–358, June 1989.

[14] J. C. K. Chou and M. Kamel, "Finding the position and orientation of a sensor on a robot manipulator using quaternions," *The International Journal of Robotics Research*, vol. 10, no. 3, pp. 240–254, 1991.

[15] H. H. Chen, "A screw motion approach to uniqueness analysis of head-eye geometry," in *Computer Vision and Pattern Recognition*, 1991, pp. 145–151.

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

[17] F. Dornaika and R. Horaud, "Simultaneous robot-world and hand-eye calibration," *IEEE Transactions on Robotics and Automation*, vol. 14, no. 4, pp. 617–622, 1998.

[18] K. Daniilidis, "Hand-eye calibration using dual quaternions," *International Journal of Robotics Research*, vol. 18, no. 3, pp. 286–298, 1999.

[19] N. Andreff, R. Horaud, and B. Espiau, "Robot hand-eye calibration using structure-from-motion," *The International Journal of Robotics Research*, vol. 20, no. 3, pp. 228–248, 2001.

[20] J. Schmidt, F. Vogt, and H. Niemann, "Robust hand–eye calibration of an endoscopic surgery robot using dual quaternions," in *Pattern Recognition*, B. Michaelis and G. Krell, Eds. Springer Berlin Heidelberg, 2003, pp. 548–556.

[21] M. Ulrich and C. Steger, "Hand-eye calibration of SCARA robots using dual quaternions," *Pattern Recognition and Image Analysis*, vol. 26, no. 1, pp. 231–239, 2016.

[22] J. Schmidt and H. Niemann, "Data selection for hand-eye calibration: A vector quantization approach," *The International Journal of Robotics Research*, vol. 27, no. 9, pp. 1027–1053, 2008.

[23] F. C. Park and B. J. Martin, "Robot sensor calibration: solving AX=XB on the Euclidean group," *IEEE Transactions on Robotics and Automation*, vol. 10, no. 5, pp. 717–721, 1994.

[24] S. Rémy, M. Dhome, J. M. Lavest, and N. Daucher, "Hand-eye calibration," in *International Conference on Intelligent Robots and Systems*, 1997, pp. 1057–1065.

[25] G.-Q. Wei, K. Arbter, and G. Hirzinger, "Active self-calibration of robotic eyes and hand-eye relationships with model identification," *IEEE Transactions on Robotics and Automation*, vol. 14, no. 1, pp. 158–166, Feb. 1998.

[26] J. Schmidt, F. Vogt, and H. Niemann, "Calibration–free hand–eye calibration: A structure–from–motion approach," in *Pattern Recognition*, W. G. Kropatsch, R. Sablatnig, and A. Hanbury, Eds. Springer Berlin Heidelberg, 2005, pp. 67–74.

[27] H. Nguyen and Q. Pham, "On the covariance of $\boldsymbol{X}$ in $\boldsymbol{AX} = \boldsymbol{XB}$," *IEEE Transactions on Robotics*, vol. 34, no. 6, pp. 1651–1658, 2018.

[28] A. Malti, "Hand-eye calibration with epipolar contraints: Application to endoscopy," *Robotics and Autonomous Systems*, vol. 61, no. 2, pp. 161–169, 2013.

[29] A. Tabb and K. M. A. Yousef, "Solving the robot-world hand-eye(s) calibration problem with iterative methods," *Machine Vision and Applications*, vol. 28, no. 5, pp. 569–590, Aug. 2017.

[30] K. Koide and E. Menegatti, "General hand–eye calibration based on reprojection error minimization," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 1021–1028, Apr. 2019.

[31] R. Lenz and D. Fritsch, "Accuracy of videometry with CCD sensors," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 45, no. 2, pp. 90–110, 1990.

[32] D. C. Brown, "Close-range camera calibration," *Photogrammetric Engineering*, vol. 37, no. 8, pp. 855–866, Aug. 1971.

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

[34] K.-R. Koch, *Parameter Estimation and Hypothesis Testing in Linear Models*. Berlin Heidelberg: Springer, 1999.

[35] ——, *Introduction to Bayesian Statistics*, 2nd ed. Heidelberg: Springer, 2007.

[36] J. L. Schönberger and J.-M. Frahm, "Structure-from-motion revisited," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[37] J. L. Schönberger, E. Zheng, M. Pollefeys, and J.-M. Frahm, "Pixelwise view selection for unstructured multi-view stereo," in *European Conference on Computer Vision (ECCV)*, 2016.