Online Extrinsic Calibration based on Per-Sensor Ego-Motion Using Dual Quaternions

Markus Horn*, Thomas Wodtko*, Michael Buchholz and Klaus Dietmayer

***Abstract*—In this work, we propose an approach for extrinsic sensor calibration from per-sensor ego-motion estimates. Our problem formulation is based on dual quaternions, enabling two different online capable solving approaches. We provide a certifiable globally optimal and a fast local approach along with a method to verify the globality of the local approach. Additionally, means for integrating previous knowledge, for example, a common ground plane for planar sensor motion, are described. Our algorithms are evaluated on simulated data and on a publicly available dataset containing RGB-D camera images. Further, our online calibration approach is tested on the KITTI odometry dataset, which provides data of a lidar and two stereo camera systems mounted on a vehicle. Our evaluation confirms the short run time, state-of-the-art accuracy, as well as online capability of our approach while retaining the global optimality of the solution at any time.**

***Index Terms*—Calibration and Identification, Sensor Networks, Optimization and Optimal Control**

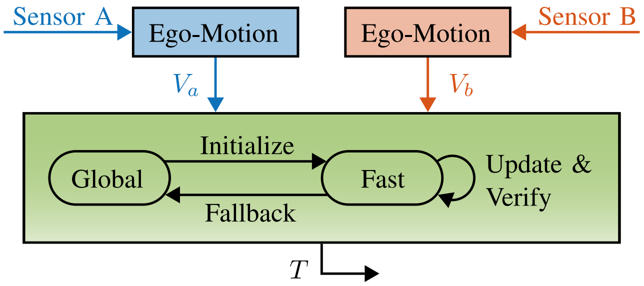

Fig. 1. Our proposed extrinsic calibration can be divided into distinct parts. First, ego-motions V_a and V_b are estimated separately for each sensor. The resulting motions are then passed to our calibration, which consists of a fast local and a global estimation part for estimating the calibration T. In the beginning, a global solution is calculated. This global solution is then used to initialize the fast optimization. When updating the fast solution, its globality is verified by evaluating the duality gap. In case of errors, the global solution is used as a fallback.

I. Introduction

With an evolving automation process, a growing number of sensors are embedded in robotic systems. Two exemplary reasons for this are increasing the field of view (FoV) or densifying certain areas. While intrinsic sensor parameters are often determined independently for each sensor, an extrinsic calibration is the key for fusing data from various sensors. Thus, extrinsic sensor calibration is a crucial step in the process of building any robotic system.

Several approaches for extrinsic sensor calibration have been proposed so far. While following the same objective, they distinctively vary in strategy. Most available algorithms belong to one of the following three categories: target-based, registration-based, and motion-based. Target-based algorithms [1]–[3] aim to detect engineered calibration targets within the FoV of multiple sensors at different positions or points in time. The extrinsic calibration is then estimated by matching the estimated target poses within each sensor's coordinate system. In contrast, registration-based approaches [4]–[6] directly use overlapping FoVs of multiple sensors and estimate the calibration by aligning their measurements. For example, for lidar-camera calibration [5], lidar depth discontinuities are aligned with image edges. However, target- and registration-based methods are subject to severe limitations, since they either require a specific calibration procedure, involving capturing the target at different positions, or can only be used on sensors with overlapping FoVs. Furthermore, they are restricted to specific sensor types and cannot be used on motion sensors like Inertial Measurement Units (IMUs). In contrast, motion-based calibration approaches [7], [8], including our proposed algorithm, only require per-sensor ego-motion estimates. This makes them suitable for many sensor types such as cameras, lidars, radars, or IMUs. Although per-sensor ego-motion is not directly available from most sensors, algorithms used for registration and mapping are effective tools to estimate it [9]–[11]. Since we want to offer a universal calibration tool that can be deployed on many different sensor setups, our work is focused on motion-based extrinsic calibration.

While some calibration approaches are designed for online use [5], [12], most algorithms are developed with regard to offline calibration. However, using online calibration, new sensors can be easily integrated into the system without extensive calibration procedures. Further, online calibration can be used to verify and correct a given calibration. This is an important aspect since sensors of robotic systems can be displaced, for example, after maintenance. Fig. 1 gives an overview of our online calibration approach.

One of the most diverse aspects when dealing with calibration is the representation of transformations and poses in 3D space. They have 6 degrees of freedom (DOF) [13], though different representations are often either highly over-represented or cumbersome to calculate with. Mostly, homogeneous matrices or dual quaternions (DQs) are used [13],

Manuscript received: October 10, 2020; Revised January 8, 2021; Accepted January 18, 2021.

This paper was recommended for publication by Editor Lucia Pallottino upon evaluation of the Associate Editor and Reviewers' comments.

All authors are with the Institute of Measurement, Control and Microtechnology, Ulm University, Albert-Einstein-Allee 41, 89081 Ulm, Germany {firstname}.{lastname}@uni-ulm.de

* *Markus Horn and Thomas Wodtko are co-first authors. Corresponding author: Markus Horn.*

both over-representing 6 DOF. While Euler angles are a more compact representation for orientations, without further definitions, their representation has many ambiguities. Additionally, they are not directly feasible for error calculation since, for example, L^2 errors can differ when changing the defined rotation axes order. Therefore, Euler angle representations should be avoided whenever possible. Homogeneous matrices for rigid body displacements are another way for representing 3D transformations. Although they are better for calculation, as they form a group w.r.t. multiplications, they are still subject to constraints and ambiguities. Those need close attention when working with them. Especially for optimization approaches, these constraints must always be enforced, leading to cumbersome problem definitions.

In contrast, DQs provide a rather compact representation and still form a group w.r.t. multiplications. Furthermore, only a single ambiguity concerning the sign exists for DQs, and only two constraints are necessary to ensure a valid transformation representation. Hence, we formulate our problem based on DQs, since they are more feasible and compact than homogeneous matrices and over-represent rigid displacements by only two parameters.

Our main contributions are:

- a problem formulation allowing integration of previous knowledge in Section III, e.g., common ground planes for planar motion,
- a globally optimal online extrinsic calibration approach from per-sensor ego-motion, based on a DQ optimization problem in Section IV,
- an extensive evaluation based on simulated and on real-world data from public datasets in Section V,
- a Python-based open-source library for ego-motion simulation and extrinsic calibration[1].

II. Related Work

For estimating the required per-sensor ego-motion, registration algorithms [4], [11] or online capable SLAM algorithms like LOAM for lidars [9] or OpenVSLAM for cameras [10] can be used. Additionally, SLAM algorithms are often directly used to estimate the sensor calibration by registering the individual maps of multiple per-sensor SLAM algorithms [14]–[16]. Nevertheless, overlaps between the maps are crucial in this case. This creates unwanted requirements on the ego-motion, for example, large rotations within a small position range [14].

However, many other algorithms only rely on per-sensor ego-motion. Most of them use the rotation matrix or the quaternion representation for the rotation and handle the translation separately as a vector. This also means, similar to [17] or [18], the rotation is often optimized first, and afterward, the translation in a consecutive step. In [18], the quaternion representation for rotations is used, and quaternion products are transformed into matrix-vector products for defining the optimization problem. Eigenvalue decomposition is then applied for obtaining the solution.

In contrast, DQs provide a combined representation for rotation and translation. DQs are more widely used in hand-eye calibration tasks [19], although approaches also exist for regular extrinsic calibration [7]. In [19], the problem is solved analytically by applying the Singular Value Decomposition (SVD). Assuming the two smallest singular values are almost zero, the associated right-singular vectors are then projected onto the constrained manifold. However, as our experiments indicate, this approach fails to find an appropriate solution in case this assumptions is violated, i.e., for planar motion with higher noise. In [7], the optimization problem is defined based on the Lie algebra of the Lie group of DQs and the calibration is solved by a non-linear least squares optimization.

A major drawback of the previously mentioned works is their local optimization approach, which leads to dependencies of the result on the initialization. Addressing this issue, based on [20], a global optimization for extrinsic and hand-eye calibration with additional scale estimation is proposed by [8], [21]. The rotation matrix and the translation vector are estimated using Lagrange duality. However, due to the rotation matrix representation, the optimization has 22 constraints in total, which increases the computation time. Similarly, another global approach proposed in [22] suffers from a high computational burden, more precisely, 6 s for only 1630 transformation pairs. In contrast to the previously mentioned approaches, we provide online capable global and local strategies based on the DQ representation and an additional globality verification for the local approach.

For the special case of indoor robotics and autonomous vehicles with planar motion, not all calibration parameters are observable since the rotation axes of all transformations are parallel [7]. Therefore, the non-observable parameter z as well as the roll and pitch angle are often estimated from a common ground plane [15], [18], [23] and only the remaining parameters x, y, and yaw angle are estimated from per-sensor motion. Since our approach is based on DQs, we can restrict the optimization on planar calibration only, i.e., x, y, and yaw angle, for the cost of only two additional constraints.

To summarize this section, to the best of our knowledge, we propose the first online calibration with certifiably global solutions, which can be used for non-planar and planar motion-based calibration.

III. Problem Formulation

In this section, we derive the optimization problem similar to [8], [20], but represent rigid displacements with DQs instead of homogeneous matrices. This results in a simplified problem formulation involving fewer constraints. Thus, DQs are generally introduced before the actual objective function is derived.

For our work, we assume synchronized sensor pairs and known intrinsic calibrations. This also includes the scaling for monocular camera motion estimation. If sensors do not provide simultaneous measurements but provide their data in a common time frame, the motion estimations can be interpolated, for example, similar to [23]. Further, all noise,

[1] https://github.com/uulm-mrm/motion_based_calibration

if not stated differently, is assumed to originate from additive white Gaussian processes.

In the following, transformations are generally referred to as functions, denoted with uppercase letters, meaning no representation is implicitly specified. A composition of two transformations T_a and T_b is denoted by $T_{ab} = T_a \circ T_b$, which refers to the usual function composition. (Dual) quaternions are denoted with lowercase letters and the dot multiplication "$\cdot$" refers to the respective (dual) quaternion multiplication. Vectorized representations and matrices are always denoted with lowercase and uppercase bold letters, respectively. For example, if q is a (dual) quaternion, $\boldsymbol{q} = \mathrm{vec}(q)$ represents q as a vector. Thus, the transformation $T_{ab} = T_a \circ T_b$ can be described using homogeneous matrices $\mathbf{T}_{ab} = \mathbf{T_b T_a}$ or DQs $q_{ab} = q_b \cdot q_a$.

A. Dual Quaternions

DQs, which combine the dual number theory with ordinary quaternions, are briefly introduced in this section. For a more detailed explanation, we refer to [13]. Given two quaternions q_r and q_d, a DQ q is noted by

$$q = q_r + \epsilon\, q_d\,, \tag{1}$$

where ϵ is the dual number with $\epsilon^2 = 0$ and $\epsilon \neq 0$, which commutes with all other elements of the algebra. q_r is called the real part and q_d the dual part. The conjugate of a DQ q^* is defined as

$$q^* = q_r^* + \epsilon\, q_d^*\,, \tag{2}$$

with q_r^* and q_d^* being the respective conjugated quaternions. Similar to quaternions, a DQ q is a unit DQ if and only if

$$\|q\|^2 = q \cdot q^* \overset{!}{=} 1\,. \tag{3}$$

Carrying out the multiplication of (3) leads to

$$\begin{aligned}\|q\|^2 = q \cdot q^* &= (q_r + \epsilon\, q_d) \cdot (q_r^* + \epsilon\, q_d^*)\\ &= \|q_r\|^2 + \epsilon(q_r \cdot q_d^* + q_d \cdot q_r^*) \overset{!}{=} 1\,.\end{aligned} \tag{4}$$

Thus, a DQ $q = q_r + \epsilon\, q_d$ is unit if and only if

$$\|q_r\|^2 = 1 \quad \wedge \quad q_r \cdot q_d^* + q_d \cdot q_r^* = 0\,. \tag{5}$$

The group of unit DQs is of special interest for this work, since it surjectively represents rigid body displacements, as explained in the following. By restricting the real part of DQs to be positive, rigid body displacements are represented unambiguously in our work. Given an arbitrary rotation with axis $\boldsymbol{n} \in \mathbb{R}^3$ and magnitude $\theta \in \mathbb{R}$, as well as a translation $\boldsymbol{t} \in \mathbb{R}^3$, the quaternions r and t, representing rotation and translation, are created first by

$$r = \left(\cos\frac{\theta}{2}, \boldsymbol{n}\sin\frac{\theta}{2}\right)\,, \quad t = (0, \boldsymbol{t})\,, \tag{6}$$

using the scalar-vector representation of quaternions. The DQ q is subsequently defined by

$$q = r + \epsilon \cdot \frac{1}{2} \cdot t \cdot r\,, \tag{7}$$

representing the respective 3D transformation of r and t. Such a transformation can be applied to a point $\boldsymbol{v} \in \mathbb{R}^3$ similar to ordinary quaternions. To do so, the respective DQ $q_v = 1 + \epsilon\,(0, \boldsymbol{v})$ is transformed by

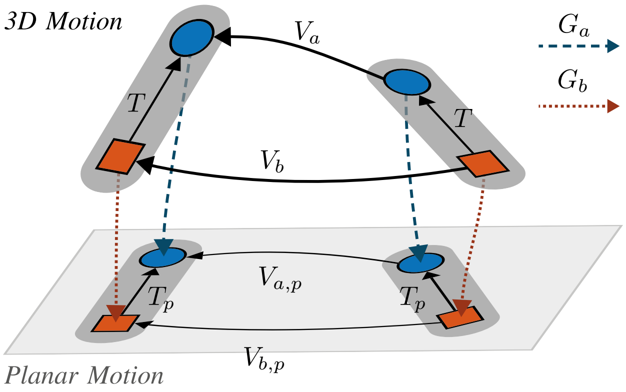

Fig. 2. Motions V_a and V_b of two rigidly mounted sensors with calibration T: The upper part describes general motion in 3D space, while the lower part represents the respective projection onto a plane using G_a and G_b.

$$\tilde{q_v} = q \cdot q_v \cdot q^*\,. \tag{8}$$

The transformed point $\tilde{\boldsymbol{v}}$ can be retrieved from $\tilde{q_v}$ since the structure is preserved.

A key feature of DQs, which we will take advantage of later, is that multiplications can be expressed using matrices. Given two DQs p and q, their multiplication is defined by

$$p \cdot q = p_r \cdot q_r + \epsilon\,(p_r \cdot q_d + p_d \cdot q_r)\,. \tag{9}$$

Let $\mathbf{M}^l_{p_r}, \mathbf{M}^l_{p_d}$ and $\mathbf{M}^r_{q_r}, \mathbf{M}^r_{q_d}$ represent the left and right side multiplications of the respective ordinary quaternions [13]. Now, $\mathbf{Q}^l_p$ and $\mathbf{Q}^r_q$ defined by

$$\mathbf{Q}^l_p = \begin{bmatrix}\mathbf{M}^l_{p_r} & \mathbf{0}\\ \mathbf{M}^l_{p_d} & \mathbf{M}^l_{p_r}\end{bmatrix}\,, \quad \mathbf{Q}^r_q = \begin{bmatrix}\mathbf{M}^r_{q_r} & \mathbf{0}\\ \mathbf{M}^r_{q_d} & \mathbf{M}^r_{q_r}\end{bmatrix}\,, \tag{10}$$

represent the left and right side multiplication of p and q, respectively. Given $\mathbf{Q}^l_p$ and $\mathbf{Q}^r_q$, the multiplication (9) can be expressed with

$$\mathrm{vec}(p \cdot q) = \mathbf{Q}^l_p \cdot \mathrm{vec}(q) = \mathbf{Q}^r_q \cdot \mathrm{vec}(p)\,, \tag{11}$$

where $\mathrm{vec}(\cdot)$ denotes the representation of a DQ as eight dimensional vector. Generally, quaternions are sequentialized by the order (w, x, y, z), where w is the real part and (x, y, z) the respective vector part. For DQs, the real and the dual vectors are concatenated.

B. Deriving a Cost Function

Similarly to [8], our first objective is to formulate a quadratic cost function. Given an arbitrary graph of transformations, the loop-closing condition requires the composition of all transformations along any path with identical start and end point to result in an identity transformation. Fig. 2 shows two rigidly connected sensors, which are freely moved in 3D space. This motion of two sensors can be conceived as a graph, where T is the transformation from sensor B to sensor A, and V_a and V_b are respective sensor ego-motion transformations. The general loop-closing condition is then given by

$$V_b \circ T \circ V_a^{-1} \circ T^{-1} = I\,, \tag{12}$$

where I denotes the identity transformation. Based on this condition, the authors of [8] formulate a quadratic function using homogeneous transformation matrices. In this work, we use DQs instead. Thus, given DQs q_T, q_a, and q_b representing the respective transformations, the condition (12) can be written as

$$q_a \cdot q_T = q_T \cdot q_b \,. \tag{13}$$

Subsequently, a quadratic representation is derived by

$$\begin{aligned} & q_T \cdot q_b - q_a \cdot q_T && = 0 \\ \iff \quad & (\mathbf{Q}^r_{q_b} - \mathbf{Q}^l_{q_a}) \ \mathrm{vec}(q_T) && = \mathbf{0} \\ \overset{\|\cdot\|_2^2}{\iff} \quad & \mathrm{vec}(q_T)^{\mathrm{T}} \, \mathbf{Q} \, \mathrm{vec}(q_T) && = 0 \,, \end{aligned} \tag{14}$$

where the cost matrix $\mathbf{Q} = (\mathbf{Q}^r_{q_b} - \mathbf{Q}^l_{q_a})^{\mathrm{T}}(\mathbf{Q}^r_{q_b} - \mathbf{Q}^l_{q_a})$ is used. In order to apply weights, a weight matrix $\mathbf{W} = \mathrm{diag}(\sqrt{w_1}, ..., \sqrt{w_8})$ can be introduced, leading to a weighted $\mathbf{Q}_w$ given by

$$\mathbf{Q}_w = (\mathbf{Q}^r_{q_b} - \mathbf{Q}^l_{q_a})^{\mathrm{T}} \mathbf{W}^{\mathrm{T}} \mathbf{W} (\mathbf{Q}^r_{q_b} - \mathbf{Q}^l_{q_a}) \,. \tag{15}$$

The previous equations are only derived for a single pair of ego-motion estimates. For multiple steps n, the weighted cost matrix of step i is defined as $\mathbf{Q}_{w,i}$. Let now the cost function $J : D \to \mathbb{R}$ be of the form

$$J(\boldsymbol{q}) = \sum_{i=1}^{n} \eta_i \, \boldsymbol{q}^{\mathrm{T}} \mathbf{Q}_{w,i} \boldsymbol{q} \,, \tag{16}$$

with weights η_i and $\boldsymbol{q} \in \mathbb{R}^8$ representing a vectorized DQ. J represents the sum of the left side of (14) for n steps and, thus, can be used as quadratic cost function for the following DQ based optimization procedures. The step specific weights η_i must satisfy $\sum_{i=1}^{n} \eta_i = 1$. This normalization is required for a consistent globality verification, as described later in Section III-F. For $\eta_i = \frac{1}{n}$, weights are equally distributed.

C. Unit Dual Quaternion Constraints

To ensure that the DQ q represents a rigid body displacement, certain constraints must be satisfied. Given the condition (5), two equality constraint functions g_1 and g_2, formally given by

$$g_1(\boldsymbol{q}) = 1 - \left\| \boldsymbol{q}_{\{1\dots4\}} \right\|^2 \overset{!}{=} 0 \,, \tag{17a}$$

$$g_2(\boldsymbol{q}) = 2(q_1 q_5 + q_2 q_6 + q_3 q_7 + q_4 q_8) \overset{!}{=} 0 \,, \tag{17b}$$

are defined. For $\boldsymbol{q} = \begin{bmatrix} q_1 & \dots & q_8 \end{bmatrix}^{\mathrm{T}}$, the subvector containing the first four elements is denoted by $\boldsymbol{q}_{\{1\dots4\}}$. The constraint g_1 requires the first half of $\boldsymbol{q}$ to be normalized, which reflects the left part of (5). The right part is represented by g_2 after carrying out the multiplication.

D. Optimization Problem

Given the cost function J and the two equality constraints g_1 and g_2, the optimization problem is defined by

$$\min_{\boldsymbol{q}} \quad J(\boldsymbol{q}) \tag{18a}$$

$$\text{w.r.t.} \ \boldsymbol{g}(\boldsymbol{q}) = \begin{pmatrix} 1 - \left\| \boldsymbol{q}_{\{1\dots4\}} \right\|^2 \\ 2(q_1 q_5 + q_2 q_6 + q_3 q_7 + q_4 q_8) \end{pmatrix} \overset{!}{=} \mathbf{0} \,. \tag{18b}$$

In comparison to the primal optimization problem of [8], the number of equality constraints is reduced from 22 down to 2. Additionally, its dimension is reduced from 10 to 8. Thus, starting from the same initial condition, the optimization problem (18) is a more compact representation due to the use of DQs.

A main feature of [8] is the certifiably global solution. In fact, the globality of a solution can only be guaranteed under certain conditions, which are outlined later. Otherwise, it is only possible to verify the globality of a given solution. Since the structure of both optimization problems using either homogeneous matrices or DQs is equal, the Lagrangian dual problem can be defined similarly to [20] for our problem (18). Furthermore, due to the small number of constraints, the respective Lagrangian function is less complicated. Formally, given the Langrange multiplier $\boldsymbol{\lambda} = \begin{bmatrix} \lambda_1 & \lambda_2 \end{bmatrix}^{\mathrm{T}}$, it is given by

$$L(\boldsymbol{q}, \boldsymbol{\lambda}) = \boldsymbol{q}^{\mathrm{T}} \mathbf{Q} \boldsymbol{q} + \boldsymbol{\lambda}^{\mathrm{T}} \boldsymbol{g}(\boldsymbol{q}) \,. \tag{19}$$

Using $\mathbf{P}_{||}(\lambda)$ and $\mathbf{P}_{\times}(\lambda)$ defined by

$$\mathbf{P}_{||}(\lambda) = \begin{bmatrix} -\lambda \mathbf{I}_4 & \mathbf{0}_4 \\ \mathbf{0}_4 & \mathbf{0}_4 \end{bmatrix} , \quad \mathbf{P}_{\times}(\lambda) = \begin{bmatrix} \mathbf{0}_4 & \lambda \mathbf{I}_4 \\ \lambda \mathbf{I}_4 & \mathbf{0}_4 \end{bmatrix} , \tag{20}$$

it can be further simplified to

$$\begin{aligned} L(\boldsymbol{q}, \boldsymbol{\lambda}) &= \boldsymbol{q}^{\mathrm{T}} \mathbf{Q} \boldsymbol{q} + \boldsymbol{q}^{\mathrm{T}} \mathbf{P}_{||}(\lambda_1) \boldsymbol{q} + \lambda_1 + \boldsymbol{q}^{\mathrm{T}} \mathbf{P}_{\times}(\lambda_2) \boldsymbol{q}, \\ &= \boldsymbol{q}^{\mathrm{T}} \mathbf{Z}(\boldsymbol{\lambda}) \boldsymbol{q} + \lambda_1 \,, \end{aligned} \tag{21}$$

where $\mathbf{Z}(\boldsymbol{\lambda})$ is given by $\mathbf{Z}(\boldsymbol{\lambda}) = \mathbf{Q} + \mathbf{P}_{||}(\lambda_1) + \mathbf{P}_{\times}(\lambda_2)$. Next, the Lagragian dual problem is specified. It is given by

$$\max_{\lambda_1} \quad \lambda_1 \tag{22a}$$

$$\text{w.r.t.} \quad \mathbf{Z}(\boldsymbol{\lambda}) \succeq \mathbf{0} \,. \tag{22b}$$

The dual problem results in a Semidefinite Programming (SDP) problem, for which solvers like [24] are available.

E. Planar Motion Constraints

In this section, we demonstrate how additional knowledge can be integrated into our proposed approaches by adding respective constraints. A planar-only motion hereby serves as an example. We assume that the common ground plane is already estimated for both sensors using concepts similar to [15], [18], [23]. Without loss of generality, we treat the ground plane as xy-plane. If a sensor's ground plane is given in Hesse normal form with normal vector $\boldsymbol{v}_G$ and distance d_G, we can calculate an aligning DQ q_G by

$$\boldsymbol{n}_G = \boldsymbol{v}_G \times \boldsymbol{e}_z \,, \tag{23a}$$

$$\theta_G = \arccos(\boldsymbol{v}_G^{\mathrm{T}} \, \boldsymbol{e}_z) \,, \tag{23b}$$

$$\boldsymbol{t}_G = d_G \, \boldsymbol{e}_z \,, \tag{23c}$$

with $\boldsymbol{e}_z = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^{\mathrm{T}}$ and by using (6) and (7). In general, the transformation q_G for sensors a and b is denoted by G_a and G_b, respectively. As illustrated in Fig 2, the planar components of estimated motion and calibration are calculated as

$$V_{a,p} = G_a^{-1} \circ V_a \circ G_a \,, \tag{24a}$$

$$V_{b,p} = G_b^{-1} \circ V_b \circ G_b \,, \tag{24b}$$

$$T_p = G_b^{-1} \circ T \circ G_a \,. \tag{24c}$$

In the following, we use $V_{a,p}$ and $V_{b,p}$ as new inputs for estimating the planar calibration T_p. Fig. 2 illustrates the transformation from 3D motion to planar motion.

After applying the previously described transformations, the values of z as well as of the roll and pitch angle are assumed to be zero. This means that r and t of (7) are given by

$$r \overset{!}{=} \left(\cos\frac{\theta}{2},\, \begin{bmatrix}0 & 0 & \sin\frac{\theta}{2}\end{bmatrix}^{\mathrm{T}}\right) , \tag{25a}$$

$$t \overset{!}{=} \left(0,\, \begin{bmatrix}x & y & 0\end{bmatrix}^{\mathrm{T}}\right) . \tag{25b}$$

From (7), we can see that $r = q_r$ and $t = 2q_d q_r^*$. In combination with (25), this results in the following constraints for planar-only calibration:

$$q_2^2 + q_3^2 = 0\,, \quad q_1 q_8 - q_4 q_5 = 0\,. \tag{26}$$

These constraints are included into the Lagrangian function using $\mathbf{P}_r(\lambda)$ and $\mathbf{P}_t(\lambda)$, which are defined similarly to (20).

After estimating the planar calibration T_p, the full calibration T is finally calculated as

$$T = G_b \circ T_p \circ G_a^{-1}\,. \tag{27}$$

F. Globality Verification

Solving our problem using a local optimization approach generally cannot guarantee to result in a global optimum. However, in our case, it is possible to verify the globality of a local solution using the verification proposed in [25].

Given a local solution $\hat{\boldsymbol{q}}$, the dual problem's first-order optimality condition must be satisfied for $\hat{\boldsymbol{q}}$ in order to be a global solution. This is formally given by

$$\left.\frac{\partial L}{\partial \boldsymbol{q}}\right|_{\substack{\boldsymbol{q}=\hat{\boldsymbol{q}}\\ \boldsymbol{\lambda}=\hat{\boldsymbol{\lambda}}}} = \mathbf{Z}(\hat{\boldsymbol{\lambda}})\,\hat{\boldsymbol{q}} \overset{!}{=} \mathbf{0}\,. \tag{28}$$

Since (28) is an over-determined linear system of equations, it can be solved using least squares algorithms w.r.t. $\hat{\boldsymbol{\lambda}}$, given the local solution $\hat{\boldsymbol{q}}$. If its solution is such that $\mathbf{Z}(\hat{\boldsymbol{\lambda}})\,\hat{\boldsymbol{q}} = \mathbf{0}$ and $\mathbf{Z}(\hat{\boldsymbol{\lambda}}) \succeq \mathbf{0}$, the solution $\hat{\boldsymbol{q}}$ is globally optimal. Otherwise, the duality gap is evaluated and serves as a measure for the distance between local and global solutions. Since the sum of the weights η_i in the cost function (16) is normalized to one, it is possible to set a threshold for the duality gap independent of the number of measurements n.

IV. Algorithms

In this section, we describe the fast local and the global optimization approach for the previously described problem formulation. Further, we propose an algorithm which combines both approaches for globally optimal online calibration.

A. Fast Optimization

Since the primal optimization problem (18) only has two constraints, a fast optimization can be achieved by applying standard constrained quadratic solving approaches. We have compared the performance of two approaches, i.e., Sequential Quadratic Programming (SQP) and Interior-Point (IP) [26], in preliminary investigations. Since SQP methods have performed better, they are used in the following.

Due to the quadratic structure of the cost function, its gradient $\nabla J(\boldsymbol{q}) = 2\mathbf{Q}\boldsymbol{q}$ can be passed to the optimizer analytically. Although the fast optimization only retains local solutions, their globality can still be verified using the concept described in Section III-F. Performance differences, especially during online calibration, are investigated later in Section V.

B. Global Optimization

While local optimization approaches are well suited for certain scenarios, a global solution is mostly favored when calibration is only performed once and with lower speed requirements. Thus, similar to [20], a global approach solving the Lagrangian dual problem is outlined here. As stated before, the optimization problem given in (22) can be solved using solvers like [24]. However, these solvers generally only deliver a global dual solution $\hat{\boldsymbol{\lambda}}$, based on which the respective primal solution has to be recovered. In general, the primal solution is given by

$$\hat{\boldsymbol{q}} = \arg\min_{\boldsymbol{q}} L(\boldsymbol{q}, \hat{\boldsymbol{\lambda}})\,, \tag{29}$$

which results in a quadratic subproblem given by

$$\hat{\boldsymbol{q}} = \arg\min_{\boldsymbol{q}} \boldsymbol{q}^{\mathrm{T}}\mathbf{Z}(\hat{\boldsymbol{\lambda}})\boldsymbol{q}\,. \tag{30}$$

Only if the duality gap is zero, the primal solution is guaranteed to be global. This is achieved if and only if the subproblem can be solved such that $\hat{\boldsymbol{q}}^{\mathrm{T}}\mathbf{Z}(\hat{\boldsymbol{\lambda}})\hat{\boldsymbol{q}} = 0$ holds. Since $\mathbf{Z} \succeq \mathbf{0}$ is required, $\hat{\boldsymbol{q}}$ has to be an element of the null space of $\mathbf{Z}(\hat{\boldsymbol{\lambda}})$. Assuming a one-dimensional null space similar to [7], we can use g_1 to retrieve a unique solution:

$$\begin{aligned} \mathrm{span}(\{\boldsymbol{v}\}) &= \mathrm{null}(\mathbf{Z}(\hat{\boldsymbol{\lambda}}))\,, \\ \hat{\boldsymbol{q}} &= \frac{\boldsymbol{v}}{\left\|\boldsymbol{v}_{\{1,\ldots,4\}}\right\|}\,. \end{aligned} \tag{31}$$

Since (31) represents all possible solutions and at least one solution must exist, other constraints, e.g. g_2, are always satisfied. As stated above, since the duality gap is zero, a solution recovered this way is guaranteed to be global.

C. Online Calibration

For online calibration, the previously described fast and global optimization approaches are combined. As shown in Fig. 1 and illustrated in Algorithm 1 for planar motion, both approaches rely on the same input data. In each iteration, the data matrix $\mathbf{Q}_i$ is updated and used to calculate the normalized $\mathbf{Q}$. If the fast optimization yields verified global solutions for at least t_{noFail}, the global optimization is not invoked. Nevertheless, the globality of the fast optimization solution is verified in every iteration and the global optimization is used as a fallback in case of discrepancies. By this, our verification ensures global solutions, even when only the local optimization is used. Additionally, due to the initialization with the solution of the previous iteration, our local approach converges faster, improving the execution time. $t_{lastLocalError}$ is initialized to the start time t_1 since the first global solution is used to initialize the fast optimization.

```
Algorithm 1 Online Extrinsic Calibration for Planar Motion
    Input: Sensor motion $\mathcal{Q}_i = \{q_{a,i}, q_{b,i}\}$, weights $W_i$
    Constant: Min. time w/o local error $t_{noFail}$
    Output: Calibration $\hat{\boldsymbol{q}}_i$
procedure CALIBRATE($\mathcal{Q}_i$, $W_i$, $t_i$, $t_{lastLocalError}$)
    Form $\mathcal{Q}_{i,p}$ from $\mathcal{Q}_i$                          ▷ planar motion (24)
    Form $\mathbf{Q}_{w,i}$ from $\mathcal{Q}_{i,p}$ and $W_i$             ▷ data matrix (15)
    $\mathbf{Q}_i \leftarrow \mathbf{Q}_{i-1} + \mathbf{Q}_{w,i}$          ▷ (16)
    $\mathbf{Q} \leftarrow \mathbf{Q}_i / i$                               ▷ equally distributed $\eta_i$
    $\hat{\boldsymbol{q}}_{i,p} \leftarrow$ Solution with SQP (init: $\hat{\boldsymbol{q}}_{i-1,p}$)   ▷ IV-A
    $isglobal \leftarrow$ Check globality of $\tilde{\boldsymbol{q}}$      ▷ III-F
    if $\neg isglobal$ then
        $t_{lastLocalError} \leftarrow t_i$                                ▷ store error time
    if $t_i - t_{lastLocalError} \leq t_{noFail}$ then
        $\hat{\boldsymbol{q}}_{i,p} \leftarrow$ Solution with SDP          ▷ global (22), (30)
    Form $\hat{\boldsymbol{q}}_i$ from $\hat{\boldsymbol{q}}_{i,p}$        ▷ 3D calibration (27)
    return $\hat{\boldsymbol{q}}_i$, $t_{lastLocalError}$
```

V. Experiments

In this section, we describe our experiments and discuss the results. First, motion data with artificial noise was simulated for investigating the influences of noise and different dataset sizes. Second, our work is compared to the approaches of [19], [7] and [8] on the publicly available dataset provided by the authors of [7]. Finally, the KITTI odometry dataset [27] is used to verify the influence of additional planar motion constraints and to evaluate the performance of our proposed online calibration approach.

The rotation and translation errors ε_r and ε_t between predicted calibration $\hat{q}$ and ground-truth calibration q_T are calculated as

$$\varepsilon_r = 2\arccos(\boldsymbol{q}_{\varepsilon,1}), \tag{32a}$$

$$\varepsilon_t = \left\| 2 \cdot q_{\varepsilon,d} \cdot q_{\varepsilon,r}^* \right\|, \tag{32b}$$

with $q_\varepsilon = q_T^{-1} \cdot \hat{q}$ and $\boldsymbol{q}_\varepsilon = \mathrm{vec}(q_\varepsilon)$. In contrast to other error metrics, these metrics describe physical quantities, i.e., rotation magnitude and translation offset. All evaluations were run on a computer containing an ADM Ryzen$^{\text{TM}}$ 7 3700X CPU and 64GB of DDR4 RAM. If not stated differently, optimization times are averaged over ten runs.

A. Simulated Data

During our work, we developed a framework for simulating sensor motion data. This framework can generate transformation matrices as well as DQs. Using this framework, extensive test data was generated, which is publicly available with our code. In contrast to real data, a simulation allows the configuration of parameters such as noise levels.

For simulation, a 2D path is projected onto a 3D surface as sketched in Fig. 3. The surface's normal vector in combination with the path's directions define the orientations along the path. The poses are then used to generate transformations for rigidly connected sensors following the 6D path.

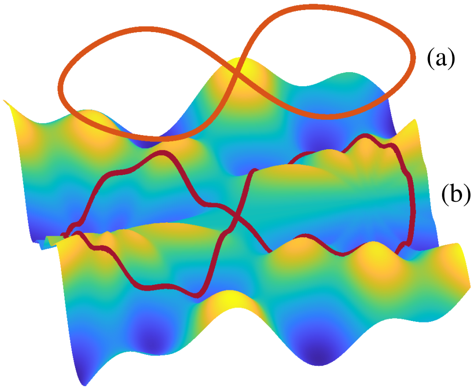

Fig. 3. An example of a simulated path: A 2D path (a) is projected onto a rough surface, resulting in a 3D path (b). The exemplary surface shown is a sinusoid mixture.

Further, unbiased Gaussian noise can be added to the transformations. Thereby, the standard deviation is referred to as noise level. It can either be configured with absolute values or relative ones, generated from the average sensor motion. For example, given a path with an average of $1\,\mathrm{m}$ and $0.1\,\mathrm{rad}$ per transformation, a noise level of $10\,\%$ leads to a respective Gaussian noise with $\sigma_{\text{trans}} = 0.1\,\mathrm{m}$, $\sigma_{\text{rot}} = 0.01\,\mathrm{rad}$.

B. General Optimization Behavior

To generally test our approach, calibrations were estimated based on simulated data with different parameters. More precisely, the data was simulated for six noise levels, 167 dataset sizes, and eight sensor poses with distances between $2\,\mathrm{m}$ and $8\,\mathrm{m}$. Each parameter set was used eight times in total, leading to a total amount of 224 448 sensor pair optimizations. The results for the global optimization are presented in Fig 4.

Generally, the results show that rotation and translation errors decrease for either an increasing amount of points or a decreasing noise level. Thus, we were able to verify that our approach is able to compensate for noise with an increasing amount of points.

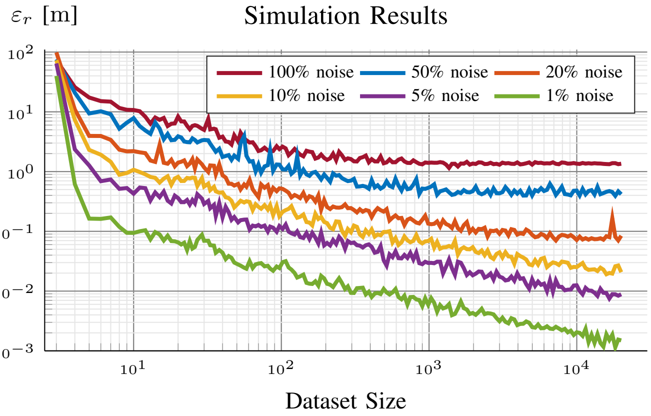

Fig. 4. Translation errors of calibrations based on simulated data are plotted over the number of points used for the global optimization. For each data point, the median over eight simulation runs was used. All results are grouped by their relative noise level. Respective rotation errors tend to follow the same general course and are therefore not depicted separately.

TABLE I
RESULTS ON THE RGB-D DATASET.

Approach	ε_t [cm]	ε_r [°]	Time [ms]
Analytic [19]	**1.08**	**0.94**	**0.2**
Local [7]	1.20	1.04	1900.0
Matrix [8]	1.28	1.06	61.5
Global (ours)	1.16	1.06	43.9
Fast (ours)	1.16	1.06	6.2

C. RGB-D Data

We have evaluated and compared our approach using the data provided by the authors of [7]. They provide 200 interpolated frames of per-sensor ego-motion estimates for two synchronized Xtion RGB-D cameras with full 6D-motion and known calibration. As shown in Table I, our approaches reach a similar translation accuracy while being considerably faster than [7], [8]. Although the local approach of [7] provides a better rotation accuracy, it takes more than 100 times longer than our local approach. Since all assumptions of [19] are met in this experiment, this approach outperforms all other methods. Furthermore, this experiment verifies that all methods are able to find appropriate solutions for well-conditioned data.

D. KITTI Odometry

The KITTI odometry dataset [27] provides calibrated data of a Velodyne HDL-64E lidar and two stereo camera setups mounted on a vehicle. Further, ground truth poses provided by an OXTS system are available for 11 sequences. An example is shown in Fig. 5. We evaluated our method on sequences 04-10, since all were recorded with the same calibration. For this purpose, we concatenated the sequences, which resulted in a total of 12 074 frames, covering approximately 20 minutes. The per-sensor ego-motions of lidar and stereo cameras were estimated using online capable implementations of LOAM[2] and OpenVSLAM[3], respectively. The ground plane for the planar motion extension was estimated in the first frame using a RANSAC based plane fitting on point clouds.

The calibration results between the lidar and the first stereo camera system are displayed in Table II, showing that a successful extrinsic sensor calibration of autonomous vehicle sensors based on per-sensor ego-motion can be achieved. Especially the rotation can be estimated precisely. For the general 3D case, our approaches deliver similar results to [8] while being considerably faster.

However, the analytic solution of [19] does not yield an appropriate solution. In contrast to the previous scenario V-C, the motion in this experiment almost only consists of planar motion, which violates the assumption of at least two non-parallel rotation axes in [19]. Furthermore, the sensor motion estimations are subject to relatively high noise.

Despite the use of a rather simple and unfiltered plane estimation, the results show that our planar extension leads to notable improvements. The translation error is decreased while still providing accurate rotation estimates and, at the same time, the execution time is further reduced.

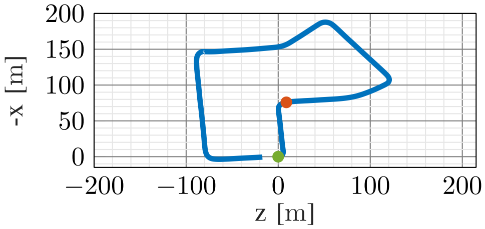

Fig. 5. The first 1000 ground-truth poses of sequence 07 of KITTI are shown. The first (green) and 150th (red) frame are marked with dots.

TABLE II
RESULTS ON THE KITTI ODOMETRY DATASET.

Approach	3D			Planar		
	ε_t[cm]	ε_r[°]	Time [ms]	ε_t[cm]	ε_r[°]	Time [ms]
Analytic [19]	$>10^5$	71.5	4.2	—	—	—
Matrix [8]	**17.84**	0.263	62.7	—	—	—
Global (ours)	20.76	**0.257**	45.1	**15.85**	0.355	53.1
Fast (ours)	20.76	**0.257**	**5.6**	**15.84**	**0.336**	**3.7**

E. Online Estimation

For evaluating our online calibration approach, we used sequence 07 of the KITTI dataset [27] and iteratively added the ego-motion estimates for planar motion as described in IV-C. Results for the first 150 frames are displayed in Fig. 6. It shall be noted that, if given more frames, errors converge to the same values as given in Section V-D. While the fast approach takes $3.7\,\mathrm{ms}$ for the planar case during offline calibration, the execution time further decreases to $1.8\,\mathrm{ms}$ during online calibration. This is comparable to the analytic solution of [19], which is not able to incorporate previous solutions. Average verification execution times of approximately $0.05\,\mathrm{ms}$ can be neglected compared to the over all processing time.

The duality gap indicates discrepancies between the local and the global solution within the first frames. Therefore, the global approach is used, leading to longer execution times. Nevertheless, except for the initial step, our online

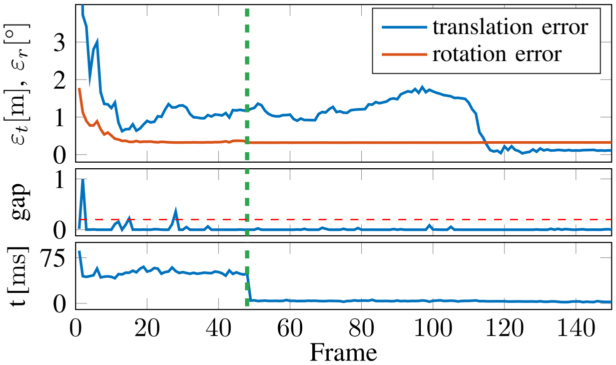

Fig. 6. Rotation and translation error, scaled duality gap, and execution time during an online calibration of sequence 07 of [27] are plotted. The red dashed line marks the verification threshold. The green dashed line at 48 frames marks the point from which only the fast approach is invoked, which decreases the execution time significantly.

[2] https://github.com/HKUST-Aerial-Robotics/A-LOAM
[3] https://github.com/xdspacelab/openvslam

calibration always complies with the sensor update rate of $10\,\mathrm{Hz}$. As soon as the local approach retrieves verified global results, the global optimization is no longer processed. This point is marked with a green dashed line in Fig. 6. While the rotation error decreases in an exponential manner, the translation error slightly increases first before it rapidly drops. The motion path displayed in Fig. 5 shows that, due to a turn of the vehicle, motion changes from almost pure translation to additional rotation at this point. Thereby, the optimization problem gets conditioned more homogeneously, leading to an immediate decrease of translation errors. Please note that global optimality does not necessarily imply high accuracy.

F. Globality Verification

Lastly, we evaluated the globality verification of our local solution. For this, the global solution of an offline calibration was used and falsified intentionally to different degrees. Comparing the duality gap with the error's magnitude, we observed a strong relation. Even at small changes of about 0.1° or $0.1\,\mathrm{m}$, discrepancies are reliably indicated. For online calibration, the duality gap is only related to the respective error magnitude and independent of the frame number since the costs are normalized using the weights η_i.

VI. Conclusion

To conclude, we have described two optimization approaches for extrinsic calibration based on per-sensor egomotion. Both, the fast local and the global approach are combined for an online calibration estimation with certifiable global solutions. Our formulation allows to add additional constraints, e.g., for planar motion. The evaluation on simulated and real-world data demonstrates that our approach achieves state-of-the-art accuracy while simultaneously obtaining the lowest run time on typical data from autonomous vehicles.

For future work, we are planning on adding the scaling factor for monocular visual odometry to our estimation, similar to [21], [23]. Furthermore, we intend to extend our online approach to track sensor calibration drifts and to determine the estimation consistency over time.

Acknowledgment

This research was accomplished within the projects UNICARagil (FKZ 16EMO0290) and U-Shift II (AK 3-433.62 -DLR/60). We acknowledge the funding by the Federal Ministry of Education and Research (BMBF) and the State Ministry of Economic Affairs Baden-Württemberg, respectively.

References

[1] A. Geiger, F. Moosmann, Ö. Car, and B. Schuster, "Automatic camera and range sensor calibration using a single shot," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2012, pp. 3936–3943.

[2] J. Kümmerle, T. Kühner, and M. Lauer, "Automatic calibration of multiple cameras and depth sensors with a spherical target," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018, pp. 1–8.

[3] J. Domhof, K. J. FP, *et al.*, "An extrinsic calibration tool for radar, camera and lidar," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2019, pp. 8107–8113.

[4] P. J. Besl and N. D. McKay, "A method for registration of 3-d shapes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, pp. 239–256, 2 1992.

[5] J. Levinson and S. Thrun, "Automatic online calibration of cameras and lasers," in *Robotics: Science and Systems IX*, 2013.

[6] N. Schneider, F. Piewak, C. Stiller, and U. Franke, "RegNet: Multimodal sensor registration using deep neural networks," in *IEEE Intelligent Vehicles Symposium (IV)*, 2017, pp. 1803–1810.

[7] J. Brookshire and S. Teller, "Extrinsic calibration from per-sensor egomotion," in *Robotics: Science and Systems VIII*, 2013.

[8] M. Giamou, Z. Ma, V. Peretroukhin, and J. Kelly, "Certifiably globally optimal extrinsic calibration from per-sensor egomotion," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 367–374, 2019.

[9] J. Zhang and S. Singh, "LOAM: Lidar odometry and mapping in real-time," in *Robotics: Science and Systems X*, 2014.

[10] S. Sumikura, M. Shibuya, and K. Sakurada, "OpenVSLAM: A versatile visual slam framework," in *ACM International Conference on Multimedia*, ser. MM '19, ACM, 2019, pp. 2292–2295.

[11] M. Horn, N. Engel, V. Belagiannis, M. Buchholz, and K. Dietmayer, "DeepCLR: Correspondence-less architecture for deep end-to-end point cloud registration," in *IEEE International Conference on Intelligent Transportation Systems (ITSC)*, 2020.

[12] S. Schneider, T. Luettel, and H.-J. Wuensche, "Odometry-based online extrinsic sensor calibration," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2013, pp. 1287–1292.

[13] J. M. McCarthy, *Introduction to Theoretical Kinematics*. Cambridge, MA, USA: MIT Press, 1990.

[14] G. Carrera, A. Angeli, and A. J. Davison, "Slam-based automatic extrinsic calibration of a multi-camera rig," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2011, pp. 2652–2659.

[15] L. Heng, B. Li, and M. Pollefeys, "CamOdoCal: Automatic intrinsic and extrinsic calibration of a rig with multiple generic cameras and odometry," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2013, pp. 1793–1800.

[16] X. Wang, H. Chen, and Y. Li, "Online calibration for monocular vision and odometry fusion," in *IEEE International Conference on Unmanned Systems (ICUS)*, 2017, pp. 602–607.

[17] Z. Taylor and J. Nieto, "Motion-based calibration of multimodal sensor arrays," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2015, pp. 4843–4850.

[18] J. Holtz and J. Biswas, "Automatic extrinsic calibration of depth sensors with ambiguous environments and restricted motion," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017, pp. 2235–2240.

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

[20] J. Briales and J. Gonzalez-Jimenez, "Convex global 3d registration with lagrangian duality," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 4960–4969.

[21] E. Wise, M. Giamou, S. Khoubyarian, A. Grover, and J. Kelly, "Certifiably optimal monocular hand-eye calibration," in *International Conference on Multisensor Fusion and Integration (MFI)*, 2020.

[22] K. Huang and C. Stachniss, "Extrinsic multi-sensor calibration for mobile robots using the gauss-helmert model," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017, pp. 1490–1496.

[23] D. Zuñiga-Noël, J.-R. Ruiz-Sarmiento, R. Gomez-Ojeda, and J. Gonzalez-Jimenez, "Automatic multi-sensor extrinsic calibration for mobile robots," *IEEE Robotics and Automation Letters*, vol. 4, no. 3, pp. 2862–2869, 2019.

[24] M. Grant and S. Boyd, *CVX: Matlab software for disciplined convex programming, version 2.2*, http://cvxr.com/cvx, Jan. 2020.

[25] L. Carlone, D. M. Rosen, G. Calafiore, J. J. Leonard, and F. Dellaert, "Lagrangian duality in 3d slam: Verification techniques and optimal solutions," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2015, pp. 125–132.

[26] J. Nocedal and S. Wright, *Numerical Optimization*. New York, NY, USA: Springer, 2006.

[27] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012, pp. 3354–3361.