2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 25-29, 2020, Las Vegas, NV, USA (Virtual)

Set-Membership Extrinsic Calibration of a 3D LiDAR and a Camera

Raphael Voges and Bernardo Wagner

***Abstract*— To fuse information from a 3D Light Detection and Ranging (LiDAR) sensor and a camera, the extrinsic transformation between the sensor coordinate systems needs to be known. Therefore, an extrinsic calibration must be performed, which is usually based on features extracted from sensor data. Naturally, sensor errors can affect the feature extraction process, and thus distort the calibration result. Unlike previous works, which do not consider the uncertainties of the sensors, we propose a set-membership approach that takes all sensor errors into account. Since the actual error distribution of off-the-shelf sensors is often unknown, we assume to only know bounds (or intervals) enclosing the sensor errors and accordingly introduce novel error models for both sensors. Next, we introduce interval-based approaches to extract corresponding features from images and point clouds. Due to the unknown but bounded sensor errors, we cannot determine the features exactly, but compute intervals guaranteed to enclose them. Subsequently, these feature intervals enable us to formulate a Constraint Satisfaction Problem (CSP). Finally, the CSP is solved to find a set of solutions that is guaranteed to contain the true solution and simultaneously reflects the accuracy of the calibration. Experiments using simulated and real data validate our approach and show its advantages over existing methods.**

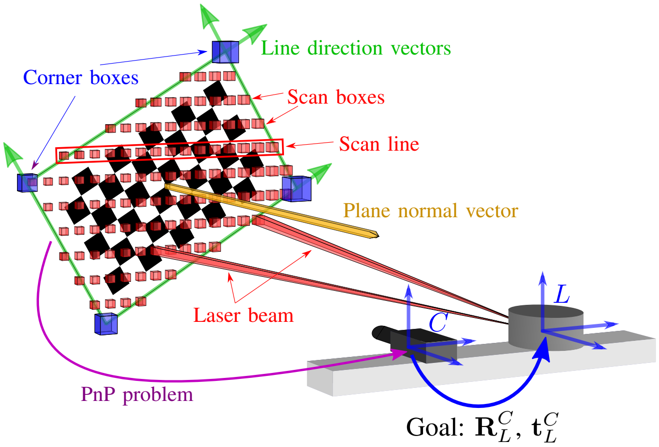


Fig. 1: Overview of our approach. To find the extrinsic transformation, i.e. the rotation matrix $\mathbf{R}_L^C$ and the translation vector $\mathbf{t}_L^C$, between camera and LiDAR **in a guaranteed way** we find plane, line and point features in both camera and LiDAR data **under interval uncertainty**.

I. INTRODUCTION

Two sensors commonly employed in mobile robotics are RGB cameras and 3D LiDARs (or laser scanner). While the camera is well suited to reidentify distinct features over time and space, the distance to those features cannot be determined from single images. In contrast, the LiDAR provides distance information, but does not allow to easily reidentify the same object. Consequently, it is beneficial to fuse information from these two complementary sensors.

Often, distinct features of a checkerboard are extracted from the data of both sensors and subsequently employed to compute the 6-DOF (Degrees Of Freedom) extrinsic calibration. The checkerboard is particularly useful as a calibration target since it is also required for the intrinsic calibration of a camera. However, these distinct features cannot be determined without error due to the measurement uncertainties of both sensors. Nevertheless, most existing approaches do not take possible errors of the LiDAR and the camera into account.

The disregard of sensor errors generates two problems. First, a zero mean error is implicitly assumed, which means that the sensor measurements should approximate the true value on average. However, this assumption is often invalid due to, for example, the distance measurement of the LiDAR depending on the incidence angle and the reflectivity of the surface [1], [2], or an incorrect intrinsic calibration of the camera or the LiDAR. Consequently, it may happen that the distance measurements of the scan points residing on the calibration target are biased (i.e. they exhibit a systematic error), and thus the extracted features are inevitably incorrect. Second, the accuracy of the extracted features cannot be assessed, and therefore cannot be employed to adequately weight the features during the computation of the extrinsic transformation parameters. In addition, the final calibration accuracy cannot be determined, thus leaving it unknown to the user whether the achieved accuracy is sufficient.

To overcome these drawbacks, we first introduce novel bounded-error sensor models that are compliant with nonzero mean errors (cf. Section IV). For example, as indicated in Fig. 1, we no longer assume to know which exact point is actually measured by the LiDAR, but only assume to know boxes (or interval vectors) that enclose the actually measured points (i.e. the red scan boxes). The size of these scan boxes is directly related to the distance uncertainty of the measurements and the divergence of the laser beams. Subsequent computations using intervals require a dedicated arithmetic, which is also known as *interval analysis* [3] and which constitutes a special case of computations on sets. Section III introduces basics and explains the choice of this error modeling technique over traditional stochastic methods.

Using the scan boxes, we determine the plane normal vector, the line direction vectors of the boundaries and the

The authors are with the Real Time Systems Group (RTS), Institute of Systems Engineering, Leibniz Universität Hannover, D-30167 Hannover, Germany. {voges,wagner}@rts.uni-hannover.de

This work was supported by the German Research Foundation (DFG) as part of the Research Training Group i.c.sens [RTG 2159].

four corner points of the checkerboard (cf. Section VI). Since we use an interval-based approach, the extracted features are unknown but bounded (i.e. intervals). For example, instead of computing the exact corner points, we are able to find corner boxes that are guaranteed to enclose the true corner points. To establish a connection between the checkerboard and the camera, we have to solve the Perspective-n-Point (PnP) problem, which is the problem of finding the pose of the camera relative to a coordinate system established for the checkerboard, under interval uncertainty. Using the resulting intervals enclosing the pose of the camera, we compute intervals enclosing the same features as for the LiDAR. Afterwards, we employ the corresponding feature intervals extracted from camera and LiDAR data to formulate a CSP that constrains the extrinsic transformation and for which constraint propagation methods can be used to compute the solution (cf. Section VII). Finally, an evaluation of the new extrinsic calibration approach using simulated as well as real data shows its applicability and its advantages over established methods (cf. Section VIII).

II. RELATED WORK

Existing approaches to find the relative transformation between camera and LiDAR can be divided into two categories: target-less or target-based. Target-less approaches do not depend on a dedicated calibration target and are particularly useful to compute the extrinsic transformation during the operation of the robot [4]. However, these approaches are generally less accurate or require manual operation by the user. Moreover, the transformation is static, and thus it suffices to perform the calibration once before operation.

Target-based approaches differ in the calibration target and the features they identify on it. While there exist some approaches employing spheres [5] or even more specialized calibration targets such as a polygonal planar board [1], most approaches rely on a checkerboard due to its availability and further usability for intrinsic camera calibration.

Unnikrishnan and Hebert proposed the first approach for the extrinsic calibration of a 3D LiDAR and a camera using a checkerboard [6]. They find the plane parameters of the checkerboard and subsequently minimize the distance of laser scan points to this plane. Their approach requires several checkerboard poses to constrain all transformation parameters. Thus, Zhou et al. do not only use the plane parameters of the checkerboard, but also identify the boundaries [7]. This allows them to compute the full extrinsic transformation from a single checkerboard pose.

All presented approaches disregard the sensor uncertainties and thus experience the problems previously mentioned. In contrast, Zhou and Deng introduce the only calibration approach that takes the uncertainty of the plane parameters into account [8]. However, they only employ the uncertainty to weight the different checkerboard poses during the optimization and do not propagate it to the final result. Consequently, none of the presented approaches are able to directly assess the accuracy of the computed extrinsic transformation. Instead, they employ ground truth information to compute the calibration error for specific test cases that cannot serve as a general statement for different sensors and calibration environments. Furthermore, as mentioned above, due to the zero mean error assumption, systematic errors of, for example, the distance measurements of the LiDAR, cannot be taken into account.

While the approach of Zhou et al. [7] produces the most accurate results, it is also the most error-prone because of the difficulty in identifying the checkerboard boundaries in a point cloud. Naturally, the resolution of the LiDAR limits the accuracy with which the boundaries can be determined. In addition, scan points close to the boundary often have large noise. Therefore and due to the reasons mentioned above, we introduce a novel approach that takes all these uncertainties of the feature extraction into account and propagates them to the result. To be able to deal with nonzero mean errors and unknown error distributions, we use interval analysis.

III. INTERVAL ANALYSIS

Intuitively, we as humans would claim that we can measure a distance with an accuracy of, for example, $\pm 1\,\mathrm{cm}$. This is the idea of interval analysis [3]. Rather than specifying an exact value or a stochastic distribution, it is assumed that measurements can be bounded using a lower and an upper bound, $\underline{x}$ and $\overline{x}$ respectively. However, no statement can be made about which value within the interval $[x] = [\underline{x}, \overline{x}]$ is most likely. We only assume that the true measurement value x^* is enclosed in the interval: $x^* \in [x]$. Formally defined, an interval is the set of all real numbers between its lower and upper bounds. Consequently, the uncertainty of an interval is its radius $r([x]) = (\overline{x} - \underline{x})/2$. Furthermore, an interval box $[\mathbf{x}]$ is defined as a vector of intervals.

The main idea of interval analysis is to extend the classical real arithmetic operators to set theory. We give an example for the addition of two intervals:

$$[-3, 1] + [4, 7] = [-3 + 4, 1 + 7] = [1, 8]. \qquad (1)$$

As can be seen, the computation takes the worst case (i.e. all possible combinations of values from both intervals) into account. As this is the case for all operations, the results of interval-based approaches are guaranteed to enclose the true solution if the initially assumed bounds are correct. However, to determine the most likely point-valued solution in the interval, a combination with stochastic approaches is required in the future.

Often, we require to characterize the set

$$\mathbb{X} = \{\, \mathbf{x} \in \mathbb{R}^n \mid \mathbf{f}(\mathbf{x}) \in \mathbb{Y} \,\} = \mathbf{f}^{-1}(\mathbb{Y}), \qquad (2)$$

where $\mathbf{f} \; : \; \mathbb{R}^n \rightarrow \mathbb{R}^m$ is a possibly nonlinear function and $\mathbb{Y}$ is a subset of $\mathbb{R}^m$. For example, $\mathbb{X}$ could be the desired set describing the position of a robot, $\mathbf{f}$ could be a measurement function and $\mathbb{Y}$ the measurement. To compute $\mathbb{X}$, the branch and bound algorithm Set Inversion Via Interval Analysis (SIVIA) can be employed [3].

The same problem can also be formulated as a CSP, where $\mathbf{f}$ is understood as a function that constrains the variables $\mathbf{x}$. In this context, so-called *contractors* [9] can be employed, which, starting from an initially arbitrary search space, use interval computations to remove inconsistent parts of the search space. Since no branching (i.e. bisections of the initial search space) is required, the solution set is computed more efficiently, but can be more pessimistic compared to using SIVIA. The classical contractor is the forward-backward contractor that decomposes all constraints into primitive constraints and considers them in isolation [3].

Finally, we want to summarize the advantages of interval-based over stochastic error modeling for our application:

- The true error distribution, which is generally unknown for off-the-shelf sensors, is not required.
- In particular, unknown systematic (i.e. nonzero mean) errors are compatible with interval analysis.
- Given correct sensor error bounds, all subsequent results are guaranteed to be correct. In the case of errors, the fault can be traced back to incorrect bounds.

Furthermore, no local minima can appear, and thus no initial values are required. Besides, the nonlinear functions do not need to be linearized, and thus no linearization errors occur.

IV. SENSOR ERROR MODELS

We assume unknown but bounded errors for both sensors. The error model for the camera was first introduced in our most recent paper [10], while the 3D LiDAR error model is an extension of an existing 2D model [11].

A. Camera Model

We employ the approach that is included in the OpenCV library [12] for the detection of checkerboard corners in camera images. Thus, the raw measurements we use are pixel points in the image. Afterwards, the pinhole camera model [13] is applied to find 3D vectors pointing in the direction of the checkerboard corners.

However, the corners cannot be detected perfectly for various reasons. First, the camera has a limited resolution, and thus the analog signal (i.e. the actual scene) has to be discretized into pixels. Second, image blur occurring, for example, due to the object not being in proper focus hampers the corner detection algorithm. Third, each image sensor is subject to noise corrupting the individual color channels of each image pixel, thus again leading to errors during the corner detection. We choose an interval-based over a stochastic error model since the discretization results in natural bounds. Moreover, the error distribution of the two remaining error sources is generally unknown [14] and depends on various circumstances (e.g. the environment), but can be bounded [15]. Fig. 2 shows the idea. Besides, the intrinsic parameters, which are required to apply the pinhole camera model, cannot be determined perfectly but can be assumed to be static. Thus, any error is systematic, and consequently the interval-based error model is preferable.

In practice, we empirically determine an interval $[\Delta_{px}]$ that encompasses all different error sources and results in an interval box that can be seen in Fig. 2. Since the interval should consider the worst case, we choose the maximum reprojection error occurring during the intrinsic calibration.

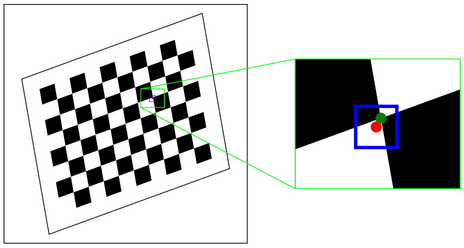


Fig. 2: Exemplary image of a checkerboard for which the corner features (green dot) have to be detected. However, various error sources lead to an inaccurate detection (red dot). Thus, we apply our bounded-error model which results in the blue interval box that encloses the true feature.

B. LiDAR Model

The raw measurements of a 3D LiDAR are the radial distance r, the polar (vertical) angle θ and the azimuthal (horizontal) angle φ. To obtain the 3D point corresponding to this measurement, these spherical coordinates can be transformed into Cartesian coordinates.

As explained previously, the distance measurement is often biased due to the incidence angle between the laser beam and the surface [1], [2], and the environment of the sensor (temperature, humidity, etc.) [16]. Since this systematic error cannot be predicted without additional information about the environment, it cannot be modeled using an established stochastic error model. However, using information from the manufacturer, the error can be bounded. Similarly, the LiDAR does not emit a perfect but a diverging beam, and thus measures a small surface instead of a point. Since the power distribution inside the beam is generally unknown, the actual location of the measured point is uncertain and could be anywhere within the beam footprint [17]. Consequently, an interval-based model is again preferable to consider this uncertainty. Fig. 3 shows the idea.

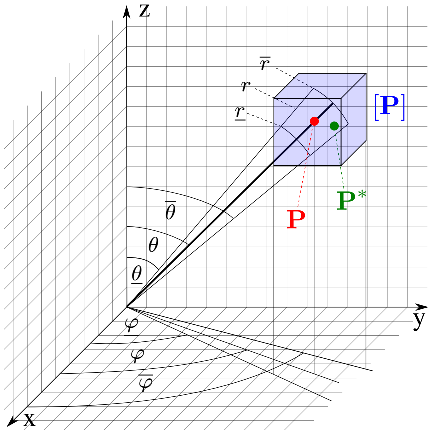


Fig. 3: Visualization of the 3D box $[\mathbf{P}]$ resulting from the unknown but bounded errors for the spherical coordinates r, θ and φ. This box is guaranteed to enclose the actually measured point $\mathbf{P}^*$, which is different from the point $\mathbf{P}$ corresponding to the raw measurement.

V. Problem Definition and Notations

The extrinsic calibration is the problem of determining the transformation from the LiDAR coordinate system L into the camera coordinate system C. This transformation consists of the rotation matrix $\mathbf{R}_L^C \in SO(3)$ and the translation vector $\mathbf{t}_L^C$. To find this transformation, we identify features on a checkerboard as shown in Fig. 1. Here, we assume the sensor clocks to be synchronized [18]. Different poses of the checkerboard help to constrain the transformation further. Thus, the following procedure, which focuses on a single pose of the checkerboard, can be repeated several times.

The LiDAR points on the checkerboard are denoted as $\mathbf{P}_l^L$, with $l \in \{1, \dots, N_p\}$. N_p is the number of scan points on the plane. The first feature we identify are the plane parameters consisting of the plane normal vector, denoted as $\mathbf{n}^L$ and $\mathbf{n}^C$, and the distance to the plane, denoted as d^L and d^C. These parameters are linked by the general plane equation

$$\mathbf{n}^A \cdot \mathbf{P}^A + d^A = 0. \tag{3}$$

The superscript $A \in \{C, L\}$ indicates the sensor coordinate system. Next, we identify the boundary lines $i \in \{1, \dots, 4\}$ of the checkerboard. The line direction vectors are denoted as $\mathbf{d}_i^L$ and $\mathbf{d}_i^C$. Corresponding points on the line are denoted as $\mathbf{Q}_{ij}^L$ and $\mathbf{Q}_{ik}^C$ with $j \in \{1, \dots, N_i\}$ and $k \in \{1, 2\}$. N_i is the total number of points on the line i which we extract from LiDAR data. In contrast, we determine only two points on every line i for the camera - namely the two adjacent corner points. These corner points are the third feature we identify and are also denoted as $\mathbf{C}_m^L$ and $\mathbf{C}_m^C$ with $m \in \{1, \dots, 4\}$.

VI. Checkerboard Feature Extraction

This section details our approach to extract the checkerboard features depicted in Fig. 1 under interval uncertainty.

A. Camera Feature Extraction

First, we solve the PnP problem under interval uncertainty as detailed in [10]. Here, we assume an unknown but bounded error for the detections of the checkerboard pattern in the image (cf. Fig. 2) and the 3D world coordinates of these corners of the checkerboard pattern: $[\mathbf{\Delta}_W]$. This results in domains for the pose of the camera in the checkerboard coordinate system W: $[\mathbf{R}_W^C]$ and $[\mathbf{t}_W^C]$.

1) Plane feature extraction: The checkerboard coordinate system W is defined such that the plane intersects the origin and $\mathbf{n}^W = (\,0\ \ 0\ \ 1\,)^\intercal$ is the normal vector of the plane. Thus, the plane normal vector in the camera coordinate system is $[\mathbf{n}^C] := [\mathbf{R}_3]$, where $\mathbf{R}_3$ is the third column of $\mathbf{R}_W^C$. Next, we know that $\mathbf{t}_W^C$ can also be interpreted as a point $\mathbf{P}^C$ on the checkerboard. Substituting $\mathbf{t}_W^C$ into (3) allows us to compute $[d^C] := -([\mathbf{n}^C] \cdot [\mathbf{t}_W^C])$.

2) Point feature extraction: Taking advantage of the known checkerboard dimensions, we are able to immediately determine the four corner points $\mathbf{C}_m^W$, $m \in \{1, \dots, 4\}$, in the checkerboard coordinate system W. Since the checkerboard cannot be manufactured perfectly, we again assume a bounded error: $[\mathbf{C}_m^W] = \mathbf{C}_m^W + [\mathbf{\Delta}_W]$. Subsequently, using $[\mathbf{R}_W^C]$ and $[\mathbf{t}_W^C]$ they are transformed into C.

3) Line feature extraction: To compute an arbitrarily scaled line direction interval vector $[\hat{\mathbf{d}}_i^W]$ in W, we subtract two adjacent corner points for which we assume the same uncertainties as before. Subsequently, we normalize this vector to obtain a unit vector and transform it into C.

B. LiDAR Feature Extraction

Identifying the same features in point clouds is more difficult since the transformation between W and L cannot be calculated as conveniently. First, we detect all scan boxes $[\mathbf{P}_l^L]$, $l \in \{1, \dots, N_p\}$ on the checkerboard. We perform the calibration in a laboratory, and can thus assume no objects to be close to the checkerboard. Still, outliers (e.g. scan points on the tripod holding the checkerboard) can occur and will be considered as detailed in the following.

1) Plane feature extraction: We use (3) to formulate a CSP for the plane normal vector. Besides, we introduce a constraint to restrict the normal vector to a unit vector:

$$\mathcal{P}: \begin{cases} \textbf{Variables: } \mathbf{n}^L,\ d^L,\ \mathbf{P}_l^L \\ \textbf{Constraints:} \\ \quad 1.\ \ \mathbf{n}^L \cdot \mathbf{P}_l^L + d^L = 0 \qquad \forall l \in \{1, \dots, N_p\} \\ \quad 2.\ \ \left\|\mathbf{n}^L\right\|_2 = 1 \\ \textbf{Domains: } [\mathbf{n}^L], [d^L], [\mathbf{P}_l^L] \end{cases}$$

The domains are initialized with $[\mathbf{n}^L] = ([-1, 1]_{\times 3})^\intercal$ and $[d^L] = [0, \infty]$. In words, we consider the normal vector to be unknown, but restrict $d^L \geq 0$ to remove the ambiguity of positive/negative unit vectors. To solve the CSP, we use a forward-backward contractor and SIVIA.

To account for outliers, we employ a relaxed intersection [19]. This means that not all scan boxes $[\mathbf{P}_l^L]$ have to meet the first constraint, but that a given number of points are potential outliers. Afterwards, we detect and remove definite outliers by checking for every point whether the constraints of $\mathcal{P}$ uphold with the computed domains $[\mathbf{n}^L]$ and $[d^L]$.

2) Line feature extraction: To determine the checkerboard boundaries, we compute so-called *hypothetical laser rays* originating from the LiDAR and passing through the border of the checkerboard. In order to do that, we iterate over each scan line (cf. Fig. 1). Such scan lines are typical for 3D LiDARs that achieve 3D vision by combining several line scanners (e.g. Velodyne VLP-16). In each scan line, we detect **adjacent** scan boxes of which one does not lie on the checkerboard but in the background and one lies on the checkerboard. Thus, we can conclude that the border of the checkerboard must be in between these two adjacent scan boxes, which allows us to determine an interval for the horizontal opening angle of the hypothetical laser ray. Fig. 4 shows a scan line in top view and visualizes our idea. Since a scan line is defined as the set of points having the same vertical opening angle, an interval for the vertical opening angle of the hypothetical laser ray is directly given.

Subsequently, the hypothetical laser rays of each scan line are intersected with the checkerboard plane to find boxes enclosing the boundary points $[\mathbf{Q}_{ij}^L]$. Moreover, multiple hypothetical rays of the same boundary span a plane, which

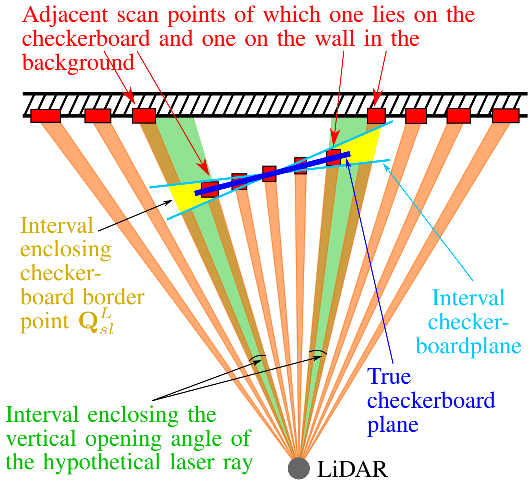

Adjacent scan points of which one lies on the checkerboard and one on the wall in the background
Interval enclosing checker-board border point $\mathbf{Q}^L_{sl}$
Interval checker-boardplane
True checkerboard plane
Interval enclosing the vertical opening angle of the hypothetical laser ray
LiDAR

Fig. 4: Visualization of the idea to find intervals that enclose the horizontal angle of the hypothetical laser rays.

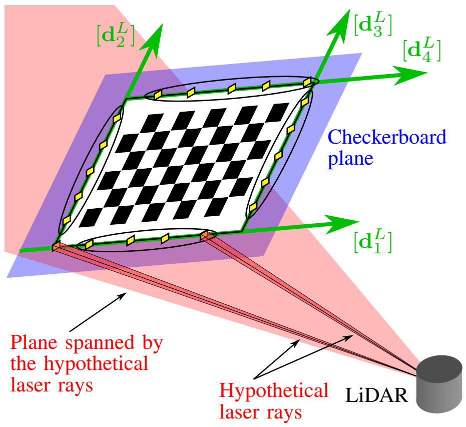


Fig. 5: The hypothetical laser rays are split into sets such that each set corresponds to one boundary of the checkerboard. A ray that could belong to multiple sets is omitted (e.g. in the bottom left). Then, the plane spanned by a set of rays is intersected with the checkerboard plane to find the boundary.

is intersected with the checkerboard plane to find an interval enclosing the unit direction vector $[\mathbf{d}^L_i]$ of the boundary line. Fig. 5 shows the idea. Naturally, if the scan lines are more or less parallel to the boundaries of the checkerboard, we cannot detect boundary points on each boundary of the checkerboard. In this case, we are not able to determine all four direction vectors.

3) Point feature extraction: The corner points $\mathbf{C}^L_m$ of the checkerboard can be computed by performing a line intersection in 3D. Let $[\mathbf{d}^L_i]$ and $[\mathbf{d}^L_j]$ be two intersecting line direction vectors (i.e. their cross product must not be 0). Accordingly, $[\mathbf{Q}^L_{ik}]$ and $[\mathbf{Q}^L_{jp}]$ are boxes enclosing points on the respective lines. This allows us to formulate the CSP:

$$\mathcal{L}: \begin{cases} \textbf{Variables: } \mathbf{C}^L_m, r_{ik}, r_{jp}, \mathbf{d}^L_i, \mathbf{d}^L_j, \mathbf{Q}^L_{ik}, \mathbf{Q}^L_{jp} \\ \textbf{Constraints:} \\ \quad 1. \;\; \mathbf{C}^L_m = r_{ik}\mathbf{d}^L_i + \mathbf{Q}^L_{ik} = r_{jp}\mathbf{d}^L_j + \mathbf{Q}^L_{jp} \\ \textbf{Domains: } [\mathbf{C}^L_m], [r_{ik}], [r_{jp}], [\mathbf{d}^L_i], [\mathbf{d}^L_j], [\mathbf{Q}^L_{ik}], [\mathbf{Q}^L_{jp}] \end{cases}$$

To compute the box $[\mathbf{C}^L_m]$, we have to find r_{ik} and r_{jp}, which are unknown constants, and thus we set their initial domains to $[r_{ik}] = [r_{jp}] = [-\infty, \infty]$. Afterwards, we employ a forward-backward contractor and SIVIA to solve the CSP.

To find the tightest possible box enclosing $\mathbf{C}^L_m$, we formulate the CSP $\mathcal{L}$ for every combination of points $\mathbf{Q}^L_{ik}$ and $\mathbf{Q}^L_{jp}$ (i.e. $\forall k \in \{1, \dots, N_i\}$, $\forall p \in \{1, \dots, N_j\}$) and intersect the resulting intervals for $\mathbf{C}^L_m$. Generally, we determine all four corner points (i.e. $m = \{1, \dots, 4\}$) of the checkerboard. However, if not all boundaries can be detected, not all corner points can be computed.

VII. Resulting Constraint Satisfaction Problem

Subsequently, we are able to formulate the CSP $\mathcal{C}$ which imposes constraints on the desired rotation matrix $\mathbf{R}^C_L$ and the translation vector $\mathbf{t}^C_L$:

$$\mathcal{C}: \begin{cases} \textbf{Variables: } \begin{array}{l} \mathbf{R}^C_L, \mathbf{t}^C_L, \mathbf{n}^L, \mathbf{n}^C, \mathbf{d}^L_i, \mathbf{d}^C_i, \\ \mathbf{Q}^L_{ij}, \mathbf{Q}^C_{ik}, \mathbf{P}^L_l, d^C, \mathbf{C}^L_m, \mathbf{C}^C_m \end{array} \\ \textbf{Constraints:} \\ \forall i \in \{1, \dots, 4\},\ \forall j \in \{1, \dots, N_i\},\ \forall k \in \{1, 2\}, \\ \forall l \in \{1, \dots, N_p\},\ \forall m \in \{1, \dots, 4\}: \\ \;\; 1. \;\; \mathbf{R}^C_L \mathbf{n}^L = \mathbf{n}^C \\ \;\; 2. \;\; \mathbf{R}^C_L \mathbf{d}^L_i = \mathbf{d}^C_i \\ \;\; 3. \;\; \left(\mathbf{I} - \mathbf{d}^C_i \left(\mathbf{d}^C_i\right)^\intercal\right)\left(\mathbf{R}^C_L \mathbf{Q}^L_{ij} + \mathbf{t}^C_L - \mathbf{Q}^C_{ik}\right) = \mathbf{0} \\ \;\; 4. \;\; \mathbf{n}^C \cdot \left(\mathbf{R}^C_L \mathbf{P}^L_l + \mathbf{t}^C_L\right) + d^C = 0 \\ \;\; 5. \;\; \mathbf{R}^C_L \mathbf{C}^L_m + \mathbf{t}^C_L = \mathbf{C}^C_m \\ \textbf{Domains: } \begin{array}{l} [\mathbf{R}^C_L], [\mathbf{t}^C_L], [\mathbf{n}^L], [\mathbf{n}^C], [\mathbf{d}^L_i], [\mathbf{d}^C_i], \\ [\mathbf{Q}^L_{ij}], [\mathbf{Q}^C_{ik}], [\mathbf{P}^L_l], [d^C], [\mathbf{C}^L_m], [\mathbf{C}^C_m] \end{array} \end{cases}$$

The first two constraints establish a connection between unit direction vectors, and thus only involve the rotation matrix. The third constraint forces a boundary point $\mathbf{Q}^L_{ij}$, which is transformed into C, to also lie on the boundary line as computed in C. The fourth constraint ensures that all scan points lying on the checkerboard fulfill the plane equation after being transformed into C. Lastly, the fifth constraint states that a corner point $\mathbf{C}^L_m$, which is transformed into C, should coincide with the corresponding corner point computed from image data. To reduce the number of unknowns in the CSP $\mathcal{C}$ to six, we express the rotation matrix using three Euler angles $\boldsymbol{\xi}^C_L = (\,\varphi^C_L \;\; \theta^C_L \;\; \psi^C_L\,)^\intercal$ defined in the ZYX convention.

The domains $[\boldsymbol{\xi}^C_L]$ and $[\mathbf{t}^C_L]$ are initialized to either an initial estimate of the transformation or to $[\boldsymbol{\xi}^C_L] = ([-\pi, \pi]_{\times 3})^\intercal$ and $[\mathbf{t}^C_L] = ([-\infty, \infty]_{\times 3})^\intercal$. In the second case, we assume no initial information. Subsequently, we build a forward-backward contractor for every constraint of the CSP $\mathcal{C}$ and intersect them to contract the parameter domains.

Different checkerboard poses provide varying constraints on each transformation parameter. Thus, we repeat the feature extraction procedure for data from several checkerboard poses and combine all resulting constraints in one common CSP. Generally, we strive to contract the transformation parameter domains as much as possible while using as few checkerboard poses as needed, to limit the computational load. In this context, Zhou et al. [7] prove that rotating the checkerboard as close as possible to the sensors is sufficient and a movement of the checkerboard is not necessary.

VIII. EXPERIMENTS AND DISCUSSION

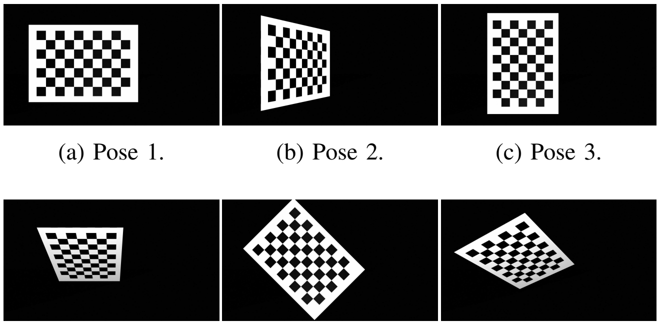
We evaluate our method using both simulated data generated using Gazebo [20] and real data. In addition, we compare our approach to the state-of-the-art algorithm of Zhou et al. [7]. However, a direct comparison is inappropriate because the objective of their approach is to find the best point-valued result, while our approach aims to enclose the true solution and simultaneously indicate the uncertainty of the computation. To depict this uncertainty, we state the radius of the intervals. Besides, we define the camera coordinate system C, which is the reference coordinate system for our results, as follows. The z-axis is pointing in the viewing direction, the x-axis is pointing to the right and the y-axis is pointing down.

A. Simulated Data

For our simulation environment, we place the checkerboard, which has a size of $100 \times 76\,\mathrm{cm}$, in a distance of roughly $2.5\,\mathrm{m}$ from the multi-sensor system. The pixel error of the simulated camera follows a Gaussian distribution with a mean value of 0 and a standard deviation of 0.01. This results in a maximum corner detection error of $0.3\,\mathrm{px}$ during camera calibration, and thus we set $[\Delta_{px}] = [-0.3, 0.3]\,\mathrm{px}$. The simulated LiDAR is a replication of the Velodyne VLP-16. The error of the spherical coordinates follows a uniform distribution without any outliers. The bounds of these uniform distribution are $[\Delta_r] = [-3, 3]\,\mathrm{cm}$ and $[\Delta_\theta] = [\Delta_\varphi] = [-0.03, 0.03]^\circ$. Furthermore, we set the following extrinsic transformation parameters to approximately replicate our real setup: $\phi_L^{C*} = 90^\circ$, $\theta_L^{C*} = 0^\circ$, $\psi_L^{C*} = 0^\circ$, ${}_x t_L^C = -27\,\mathrm{cm}$, ${}_y t_L^C = 15\,\mathrm{cm}$, ${}_z t_L^C = -12\,\mathrm{cm}$.

At first, we show results for the transformation parameters that are computed from one checkerboard pose only. We select six different checkerboard poses, which are depicted in Fig. 6, to show the influences on different extrinsic calibration parameters. Furthermore, the initial domains for the rotation are set to $[\theta_L^C] = [-90, 90]^\circ$ and $[\phi_L^C] = [\psi_L^C] = [-180, 180]^\circ$. Besides, we set $[{}_w t_L^C] = {}_w t_L^C + [-50, 50]\,\mathrm{cm}$ for $w \in \{x, y, z\}$. Thus, we assume no initial information about the extrinsic transformation other than a rough idea of the setup of the multi-sensor system.

For all six poses, our method encloses the true transformation parameters. However, the accuracy of the transformation parameters, which is depicted in Table I, varies. This can be explained by the corresponding pose of the checkerboard. For example, the first and fifth pose allow to accurately contract the translation along the z-axis. The reason is that the checkerboard plane is perpendicular to the z-axis for these poses, and thus the fourth constraint of the CSP $\mathcal{C}$, which forces scan points to lie on the checkerboard plane, is particularly suited to allow strong contractions in z-direction. In contrast, the first three (the fourth) poses of the checkerboard do not allow to contract the translation along the y-axis (z-axis) since no features can be extracted to constrain the corresponding translation parameters. This is because the scan lines are parallel to the top and bottom boundaries of the checkerboard. Similar reasons can be found for all results by linking the corresponding checkerboard pose to the CSP $\mathcal{C}$.


(a) Pose 1. (b) Pose 2. (c) Pose 3.

(d) Pose 4. (e) Pose 5. (f) Pose 6.

Fig. 6: Simulated images of six different checkerboard poses.

TABLE I: Interval radii for the poses in Fig. 6.

Pose	$r([\phi_L^C])$ (°)	$r([\theta_L^C])$ (°)	$r([\psi_L^C])$ (°)	$r([{}_x t_L^C])$ (cm)	$r([{}_y t_L^C])$ (cm)	$r([{}_z t_L^C])$ (cm)
1	0.7	0.6	0.7	2.7	50.0	1.6
2	1.2	0.5	0.8	2.8	50.0	2.0
3	0.8	0.9	0.5	4.6	50.0	1.9
4	0.4	0.6	0.7	3.2	32.2	50.0
5	1.1	1.1	0.2	5.7	5.8	1.5
6	0.8	1.0	1.2	5.4	3.7	2.4
all	0.4	0.5	0.2	2.4	2.3	1.1

In addition, Table I shows the result for the combination of the constraints from all six poses. As expected, the computed intervals resemble the best possible results of employing the different poses individually. However, some domains (e.g. the translation parameters) can be determined even more accurately. This can be explained by the fact that the combined constraints supplement each other. Thus, this experiment shows a textbook example of how different contractors can be combined to produce an even more powerful contractor.

TABLE II: Influence of different simulated error bounds.

	$r([\phi_L^C])$ (°)	$r([\theta_L^C])$ (°)	$r([\psi_L^C])$ (°)	$r([{}_x t_L^C])$ (cm)	$r([{}_y t_L^C])$ (cm)	$r([{}_z t_L^C])$ (cm)
reference	0.35	0.46	0.24	2.36	2.26	1.12
$r([\Delta_{px}])$ (px)						
0.4	0.38	0.47	0.26	2.42	2.45	1.35
0.5	0.41	0.48	0.27	2.51	2.62	1.50
0.6	0.41	0.53	0.28	2.73	2.65	1.60
$r([\Delta_r])$ (cm)						
6.0	0.38	0.55	0.36	2.76	2.37	1.15
$r([\Delta_\theta])$ (°)						
0.06	0.41	0.51	0.34	2.61	2.60	1.13
$r([\Delta_\varphi])$ (°)						
0.06	0.36	0.49	0.41	2.60	2.44	1.13

TABLE III: Results showing the influence of biased distance measurements on both our and the state-of-the-art approach.

	ϕ_L^C (°)	θ_L^C (°)	ψ_L^C (°)	${}_xt_L^C$ (cm)	${}_yt_L^C$ (cm)	${}_zt_L^C$ (cm)
True	90.0	0.0	0.0	-27.0	15.0	$\mathbf{-12.0}$
[7], no bias	90.0	0.0	0.0	-27.0	15.0	-11.9
Our, no bias	$[89.6, 90.3]$	$[-0.4, 0.3]$	$[-0.1, 0.3]$	$[-28.8, -25.0]$	$[13.1, 16.7]$	$[-13.1, -11.0]$
[7], bias	90.0	0.0	0.0	-27.1	14.9	$\mathbf{-13.0}$
Our, bias	$[89.7, 90.3]$	$[-0.4, 0.5]$	$[-0.4, 0.3]$	$[-29.5, -25.0]$	$[13.0, 16.8]$	$\mathbf{[-13.1, -10.9]}$

Finally, Table II shows results for different simulated errors for both camera and laser scanner. Naturally, the sensor error bounds are adjusted accordingly. For example, the second row shows the interval radii for $[\Delta_{px}] = [-0.4, 0.4]\,\mathrm{px}$ while the other sensor error bounds are kept as detailed above. As a reference, the first line shows the results for the error bounds introduced and evaluated above. Increasing the error results in wider intervals reflecting the increasing uncertainty. However, the uncertainty does not increase significantly. Thus, we can state that our approach can handle different errors for both camera and laser scan data without drastically increasing the uncertainty. Of course, this only applies if the error bounds are known and are neither over- nor underestimated.

1) Comparison to state-of-the-art approach: We depict exemplary results of the approach of Zhou et al. [7] for the fifth checkerboard pose: $\phi_L^C = 89.6°$, $\theta_L^C = -1.4°$, $\psi_L^C = 0.0°$, ${}_xt_L^C = -20.4\,\mathrm{cm}$, ${}_yt_L^C = 13.1\,\mathrm{cm}$, ${}_zt_L^C = -11.6\,\mathrm{cm}$. While the translation along the x-axis is off by $6.6\,\mathrm{cm}$, the translation along the z-axis is off by only $0.4\,\mathrm{cm}$ and the yaw angle is computed correctly. As can be seen from Table I, this complies with the accuracies we achieve. Our results show that the yaw angle and the translation along the z-axis can be computed most accurately. However, Zhou's approach does not allow these accuracies to be assessed solely on the basis of sensor data, and therefore the user cannot judge whether the extrinsic calibration is sufficiently accurate.

Next, we show the influence of systematic errors on both our and the established approach. We use a total of 27 checkerboard poses, including the six poses shown above, which are evenly distributed in the rotation space around the sensor setup. Originally, the error of the distance measurements of the LiDAR follows a uniform distribution in the interval $[-0.03, 0.03]\,\mathrm{m}$, and thus does not violate the zero-mean assumption of the state-of-the-art approach. The results for both approaches are depicted in the second and third row of Table III. The state-of-the-art approach performs reasonably well and is able to accurately estimate the transformation parameters. Furthermore, our own approach exhibits the same characteristics as before. Subsequently, we add a bias to the distance measurement such that the mean distance error is $1\,\mathrm{cm}$, but the error is still bounded in the interval $[-0.03, 0.03]\,\mathrm{m}$. The fourth and fifth row of Table III show the results. The approach of Zhou et al. is significantly influenced by the systematic error. As expected, the translation along the z-axis is off by approximately $1\,\mathrm{cm}$, as this is the forward facing axis, which is thus most affected by a distance error of the LiDAR. In contrast, our approach is not disturbed by systematic errors and the results are similar to the results for the unbiased data.

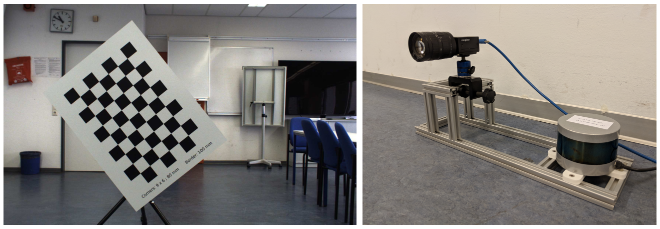
Although our experiment here might look staged, it is important to point out that such systematic errors can occur in reality due to, for example, the incidence angle of the laser beam or the surface of the measured calibration target. As shown, the stochastic approach cannot deal with such **unknown** systematic errors as the zero-mean error assumption is inherent and cannot be violated. Moreover, it is important to note that the width of the error distribution is not the deciding factor here as the stochastic approach performs well if the distance error is uniformly distributed in the same interval. In contrast, our interval-based approach can cope with unknown systematic errors as long as they are enclosed by the corresponding interval bounds. This also holds true for systematic errors that are larger than $1\,\mathrm{cm}$.

B. Real Data

Our setup that consists of a Velodyne VLP-16 LiDAR, a FLIR Grasshopper3 camera, and a $100 \times 76\,\mathrm{cm}$ checkerboard can be seen in Fig. 7. The camera has a resolution of $1920 \times 1200\,\mathrm{px}$. The LiDAR has a vertical angular resolution of $2°$. Besides, the rotation rate of the LiDAR is set to $5\,\mathrm{Hz}$, resulting in a horizontal angular resolution of $0.1°$.

The uncertainties of the LiDAR are specified by the manufacturer as $[\Delta_r] = [-3, 3]\,\mathrm{cm}$, $[\Delta_\varphi] = [-1.5, 1.5]\,\mathrm{mrad}$ and $[\Delta_\theta] = [-0.75, 0.75]\,\mathrm{mrad}$. Furthermore, we set an empirically determined maximum outlier percentage for the computation of the plane parameters of $0.5\,\%$. To find the maximum error of the checkerboard pattern detections in the camera image, we perform an intrinsic camera calibration and use the maximum reprojection error which is $[\Delta_{px}] = [-0.5, 0.5]\,\mathrm{px}$. In addition, we assume a maximum printing, manufacturing and bending accuracy for the corners and the pattern of the checkerboard of $[\mathbf{\Delta}_W] = ([-1, 1]\,\mathrm{mm}_{\times 3})$.

We gathered data from 26 different checkerboard poses. Table IV shows the results. The resulting interval radii are



(a) Checkerboard.

(b) Sensor setup.

Fig. 7: Overview of the equipment used for the evaluation.

TABLE IV: Results of our and the state-of-the-art approach for real data.

	$[\phi_L^C]$ (°)	$[\theta_L^C]$ (°)	$[\psi_L^C]$ (°)	$[{}_x t_L^C]$ (cm)	$[{}_y t_L^C]$ (cm)	$[{}_z t_L^C]$ (cm)
Our, parameter intervals	$[89.8, 90.5]$	$[-0.6, 0.4]$	$[0.4, 1.0]$	$[-28.3, -25.2]$	$[15.7, 18.8]$	$[-12.5, -10.3]$
Our, corresponding radii	0.35	0.5	0.3	1.55	1.55	1.1
State-of-the-art [7]	90.6	-0.6	0.9	-24.6	15.7	-13.2

similar to our simulation studies, and thus we conclude that our approach can also be applied to real data. Moreover, the intervals should be guaranteed to enclose the true parameters, since we were able to reliably enclose the true parameters using simulated data. Besides, the table shows the results of the established stochastic approach [7]. It can be seen that some of the computed parameters are not enclosed by our corresponding intervals. However, since our intervals are guaranteed to contain the true solution, we conclude that the parameters computed by the stochastic approach are erroneous (e.g. due to systematic errors). Since it is infeasible to obtain ground truth information using real data, we cannot make this statement with absolute certainty, but only with a strong reasoning using our simulation studies. Consequently, our approach can be used not only to perform the extrinsic calibration and assess its accuracy, but also to validate the results of a stochastic approach.

IX. CONCLUSIONS AND FUTURE WORK

We present a new approach for the extrinsic calibration of camera and LiDAR that takes sensor errors during the identification of checkerboard features into account and propagates them to the resulting calibration parameters. We use interval analysis to design new bounded-error models for both sensors, which allows us to compute intervals that are guaranteed to enclose the true solution under the assumption that the sensor error bounds are correct. Our evaluation shows that these guarantees can indeed be achieved. Moreover, the intervals allow us to individually assess the accuracy of each transformation parameter, and thus enables us to judge whether the calibration is sufficiently accurate. Finally, we are able to show that our approach is compatible with systematic errors, which is not the case for existing methods.

In future work, we plan to extend our approach to include an automatic process that informs the user of the best next checkerboard pose based on the uncertainty of the calibration parameters. In addition, we aim to combine interval-based and stochastic error modeling to find the most likely point-valued result. Finally, we plan to fuse data from camera and laser scanner to estimate the motion of a robot under interval uncertainty [21]. Here, the idea is to a use a set-based observer, which in contrast to Kalman filters can prove that the robot's state has to lie within a certain set.

REFERENCES

[1] Y. Park, S. Yun, C. Won, K. Cho, K. Um, and S. Sim, "Calibration between Color Camera and 3D LIDAR Instruments with a Polygonal Planar Board," *Sensors*, vol. 14, no. 3, pp. 5333–5353, Mar. 2014.

[2] C. Glennie, "Calibration and Kinematic Analysis of the Velodyne HDL-64E S2 Lidar Sensor," *Photogrammetric Engineering & Remote Sensing*, vol. 78, no. 4, pp. 339–347, Apr. 2012.

[3] L. Jaulin, M. Kieffer, O. Didrit, and É. Walter, *Applied Interval Analysis*. Springer London, 2001.

[4] J. Levinson and S. Thrun, "Automatic Online Calibration of Cameras and Lasers," in *Robotics: Science and Systems (RSS)*, Berlin, Germany, Jun. 2013.

[5] 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)*, Madrid, Spain, Oct. 2018.

[6] R. Unnikrishnan and M. Hebert, "Fast Extrinsic Calibration of a Laser Rangefinder to a Camera," Tech. Rep., 2005.

[7] L. Zhou, Z. Li, and M. Kaess, "Automatic Extrinsic Calibration of a Camera and a 3D LiDAR Using Line and Plane Correspondences," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Madrid, Spain, Oct. 2018.

[8] L. Zhou and Z. Deng, "Extrinsic calibration of a camera and a lidar based on decoupling the rotation from the translation," in *IEEE Intelligent Vehicles Symposium*, Alcala de Henares, Spain, Jun. 2012.

[9] G. Chabert and L. Jaulin, "Contractor programming," *Artificial Intelligence*, vol. 173, no. 11, pp. 1079–1100, Jul. 2009.

[10] R. Voges and B. Wagner, "Timestamp Offset Calibration for an IMU-Camera System Under Interval Uncertainty," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Madrid, Spain, Oct. 2018.

[11] M. Langerwisch and B. Wagner, "Building Variable Resolution Occupancy Maps Assuming Unknown But Bounded Sensor Errors ," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Tokyo, Japan, Nov. 2013.

[12] G. Bradski, "The OpenCV Library," *Dr. Dobbs Journal of Software Tools*, 2000.

[13] 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.

[14] B. T. Koik and H. Ibrahim, "A Literature Survey on Blur Detection Algorithms for Digital Imaging," in *1st International Conference on Artificial Intelligence, Modelling and Simulation*, Kota Kinabalu, Malaysia, Dec. 2013.

[15] C. Liu, W. T. Freeman, R. Szeliski, and S. B. Kang, "Noise Estimation from a Single Image," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, New York, NY, USA, Jun. 2006.

[16] Q. Wang, N. Zissler, and R. Holden, "Evaluate error sources and uncertainty in large scale measurement systems," *Robotics and Computer-Integrated Manufacturing*, vol. 29, no. 1, pp. 1–11, Feb. 2013.

[17] Q. Pentek, T. Allouis, O. Strauss, and C. Fiorio, "Developing And Validating a Predictive Model of Measurement Uncertainty For Multi-beam LiDARs: Application to the Velodyne VLP-16," in *International Conference on Image Processing Theory, Tools and Applications (IPTA)*, Xi'an, China, Nov. 2018.

[18] R. Voges, C. S. Wieghardt, and B. Wagner, "Finding Timestamp Offsets for a Multi-Sensor System Using Sensor Observations," *Photogrammetric Engineering & Remote Sensing*, vol. 84, no. 6, pp. 357–366, 2018.

[19] L. Jaulin and E. Walter, "Guaranteed robust nonlinear minimax estimation," *IEEE Transactions on Automatic Control*, vol. 47, no. 11, pp. 1857–1864, Nov. 2002.

[20] N. Koenig and A. Howard, "Design and use paradigms for Gazebo, an open-source multi-robot simulator," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Sendai, Japan, Sep. 2004.

[21] R. Voges, B. Wagner, and V. Kreinovich, "Odometry under Interval Uncertainty: Towards Optimal Algorithms, with Potential Application to Self-Driving Cars and Mobile Robots," *Reliable Computing (Interval Computations)*, vol. 27, no. 1, pp. 12–20, 2020.