Mutual Absolute Calibration of Lidar Mounting Positions on Vehicles using CAD Models

Egon Ye[1] and Matthias Althoff[2]

***Abstract*— Extrinsic calibration of a sensor mounting position on a vehicle is essential for a truthful mapping of the sensor data with respect to the vehicle. However, extrinsic calibration is usually cumbersome due to complex measurement setups or requirements on the environment, e.g., locally planar surfaces. We propose to use mutual detections of sensors mounted on different vehicles for calibrating their mounting positions. To maximize the accuracy of the detected vehicle poses, our developed method uses CAD models of the involved vehicles. Based on the accuracy of each detected vehicle pose, we further derive a formulation to estimate the calibration accuracy. The benefits of our methods are demonstrated using Monte Carlo simulations and real-world experiments with two BMW test vehicles.**

I. Introduction

The development of autonomous driving functions relies on precise environment perception for which lidars play an essential role due to their measurement accuracy. A correct interpretation of lidar data in turn requires an accurate calibration of the sensor mounting position. However, extrinsic calibration is a tedious process, which often involves complex setups.

A. Related Work

While intrinsic calibration provides sensor-specific transformation parameters, e.g., the distance offset and angular orientation of single laser beams for a lidar [1], extrinsic calibration tries to obtain the true position of the sensor with respect to a reference object. Therein, the extrinsic calibration can be further subdivided into a relative calibration between sensors and an absolute calibration with respect to the vehicle platform as aimed in our case.

The absolute calibration of a lidar relies on a repetitive detection of a specific target object or of the static environment at different relative poses. In [3], a target is manually positioned on the ground at a fixed lateral and longitudinal distance from the vehicle, which is taken as the reference to align sensor data relatively to the vehicle. To overcome errors from manual measurements, [4] detects a pole as a target over multiple sensor frames and exploits navigation data from IMU and differential GPS. Instead of using a target, some approaches make assumptions on the environment to align subsequent sensor frames. Such an approach is shown in [5], which maximizes the local planarity of surfaces in the environment to obtain relative transforms between sensor frames. Similarly, [6] uses a crispness criterion to increase the quality of superposed lidar frames.

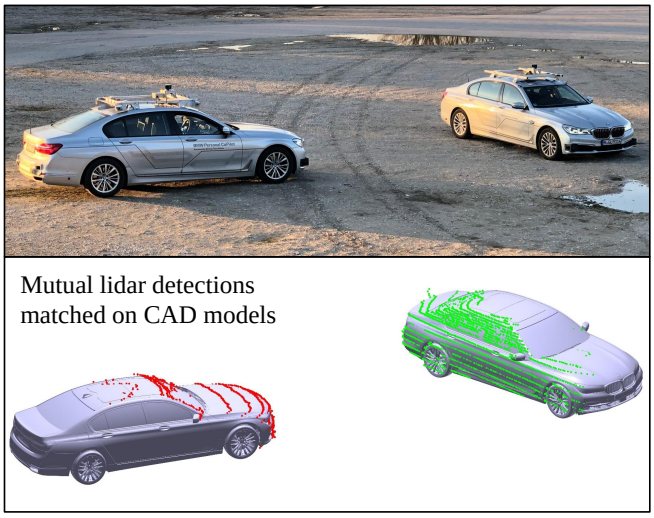


Fig. 1. Schematic depiction of the calibration setup, where lidar points corresponding to the roof racks are removed in the bottom illustration for clarity.

Approaches regarding the relative calibration of multiple lidars consider an overlapping field of view among the sensors. In [7], a sensor setup with multiple lidars and cameras was calibrated using a sphere target, where the cost function to be minimized includes the distances between pairs of sphere centers estimated by lidars. An approach based on matching geometric features of the environment in lidar data has been presented in [8]. Therein, a cost function comprising pairwise point, line, and plane feature correspondences between sensors is optimized for sensor data alignment.

The simultaneous computation of two unknown spatial relationships in a circle of spatial relationships is also referred to as the hand-eye calibration problem [9]. Different approaches have been proposed to optimize for the unknown spatial relationships [10], [11] by determining first the rotation and then the translation parameters consecutively. Since rotational errors propagate into higher translational errors, a method to solve the parameters simultaneously has been shown by [12]. The underlying idea incorporates the uncertainty of the known spatial relationships but assumes that their uncertainties are negligible for all but one of them. However, the uncertainties of the known spatial relationships are similar to each other in our concept, which is why we

[1]Egon Ye is with BMW Group, D-85716 Unterschleissheim, Germany. egon.ye@bmw.de

[2]Matthias Althoff is with the Department of Computer Science, Technical University of Munich, D-85748 Garching, Germany. althoff@in.tum.de

introduce a more generic cost function for the hand-eye calibration problem.

B. Contributions

This paper proposes a new approach to obtain the extrinsic calibration of lidars mounted on vehicles. By mutually matching CAD models to sensor point clouds, we obtain highly accurate vehicle poses relative to the sensors. With these relative pose pairs, we estimate the mounting positions of the sensors by closing a circle of transforms between each vehicle pair. We further derive the possible accuracy that we could achieve with our calibration method based on the registered pose errors. The main advantages of our approach are as follows:

- Since vehicles use each other as targets, no further targets or assumptions about the environment are required.
- Sensor data can be recorded at any place as long as the vehicles are not occluded.
- Concurrent calibration of the lidars on all vehicles.
- Recalibration is possible retrospectively for cooperative test drives with several test vehicles.
- Scalability – the more vehicles involved, the better the calibration.

We demonstrate the accuracy of our approach using Monte Carlo simulations as well as real-world experiments recorded with test vehicles.

The remainder of the paper is organized as follows: Section II gives an outline of our multi-vehicle extrinsic calibration approach. We derive its accuracy in section III. Our calibration concept is evaluated in section IV on simulated and real sensor data. We finish with conclusions in section V.

II. Proposed Approach for Extrinsic Calibration

Our approach requires the CAD model of each vehicle and coarse global localization information. We further assume that the intrinsic calibration parameters [1] have negligible errors. In subsequent derivations, we consider N vehicles, each equipped with a single sensor.

A. Homogeneous Transforms

Let the rigid transformation of a point $\boldsymbol{p} \in \mathbb{R}^3$ to $\boldsymbol{p}' \in \mathbb{R}^3$ be given by $\boldsymbol{p}' = \boldsymbol{R}\boldsymbol{p} + \boldsymbol{t}$, where $\boldsymbol{R} \in \mathbb{R}^{3x3}$ represents an orthonormal rotation matrix with $\det(\boldsymbol{R}) = 1$ and $\boldsymbol{t} \in \mathbb{R}^3$ is a translation vector. The homogeneous transform $\boldsymbol{T}$ relates the homogeneous coordinates $[\boldsymbol{p}, 1]^T$ and $[\boldsymbol{p}', 1]^T$ linearly by

$$\begin{bmatrix} \boldsymbol{p}' \\ 1 \end{bmatrix} = \begin{bmatrix} \boldsymbol{R} & \boldsymbol{t} \\ \boldsymbol{0} & 1 \end{bmatrix} \begin{bmatrix} \boldsymbol{p} \\ 1 \end{bmatrix} = \boldsymbol{T} \begin{bmatrix} \boldsymbol{p} \\ 1 \end{bmatrix}. \tag{1}$$

Thus, the homogeneous transforms $\boldsymbol{T}_{mnt,i}$ to represent the i-th sensor mounting position with respect to the i-th vehicle coordinate system as well as the transforms $\boldsymbol{T}_{fit,ij}$ to describe the pose of the i-th detected vehicle with respect to the j-th sensor coordinate system are given by

$$\boldsymbol{T}_{mnt,i} = \begin{bmatrix} \boldsymbol{R}_{mnt,i} & \boldsymbol{t}_{mnt,i} \\ \boldsymbol{0} & 1 \end{bmatrix} \tag{2}$$

and

$$\boldsymbol{T}_{fit,ij} = \begin{bmatrix} \boldsymbol{R}_{fit,ij} & \boldsymbol{t}_{fit,ij} \\ \boldsymbol{0} & 1 \end{bmatrix}, \tag{3}$$

where the components are defined as (with indices dropped for clarity)

$$\begin{aligned} \boldsymbol{R}_{fit} &= \boldsymbol{R}_x(\phi_{fit})\boldsymbol{R}_y(\theta_{fit})\boldsymbol{R}_z(\psi_{fit}), \\ \boldsymbol{t}_{fit} &= [x_{fit}, y_{fit}, z_{fit}], \\ \boldsymbol{R}_{mnt} &= \boldsymbol{R}_x(\phi_{mnt})\boldsymbol{R}_y(\theta_{mnt})\boldsymbol{R}_z(\psi_{mnt}), \\ \boldsymbol{t}_{mnt} &= [x_{mnt}, y_{mnt}, z_{mnt}]. \end{aligned} \tag{4}$$

We further introduce the vectors

$$\begin{aligned} \boldsymbol{h}_{mnt,i} &= [\psi_{mnt,i}, \theta_{mnt,i}, \phi_{mnt,i}, x_{mnt,i}, y_{mnt,i}, z_{mnt,i}]^T, \\ \boldsymbol{h}_{fit,ij} &= [\psi_{fit,ij}, \theta_{fit,ij}, \phi_{fit,ij}, x_{fit,ij}, y_{fit,ij}, z_{fit,ij}]^T \end{aligned}$$

to simplify expressions in later derivations. Fig. 2 visualizes the transformation parameters with respect to the coordinate systems of sensor S and vehicle V .

B. Registration of Object Poses

In our previous work [13], we have shown an offline tracking approach using CAD models. To determine the relative pose of the detected vehicle with respect to the sensor, we match a precise 3D point set $\mathcal{C}$ sampled from the CAD model to the sensor point cloud $\mathcal{S}$ using point cloud registration algorithms (like ICP [14], [15]). To handle outliers in matching correspondences, point cloud registration is further augmented with robust criterion functions as shown in [16].

We reuse the registration method of our previous work to obtain the transforms $\boldsymbol{T}_{fit,ij}$ and $\boldsymbol{T}_{fit,ji}$. As initial transformation, we use global localization data from differential GPS. Low-cost localization methods would also be sufficient if corresponding techniques for finding an appropriate initial transformation are realized [17, pp. 21–23].

C. Calibration Cost Function

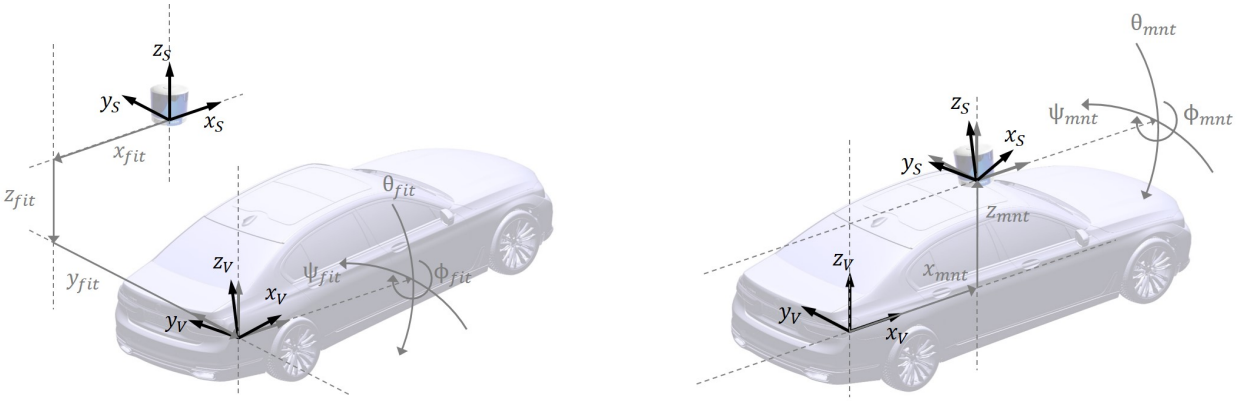
The cost function for the calibration is derived by applying the transforms for sensor mounting positions and mutually registered poses of each vehicle pair, which closes a circle of transforms as shown in Fig. 3. Since one should not have moved after completing all transformations, we introduce the errors

$$\boldsymbol{e}_{ij} = \boldsymbol{T}_{mnt,i}\,\boldsymbol{T}_{fit,ji}\,\boldsymbol{T}_{mnt,j}\,\boldsymbol{T}_{fit,ij}\,[0,0,0,1]^T, \tag{5}$$

which correspond to the consecutive application of the transforms on the origin vector. To simultaneously compute the mounting positions, we propose to minimize the errors from N_{ij} relative pose pairs between each pair of vehicles i and j (out of N vehicles) by solving

$$\boldsymbol{z}^* = \arg\min_{\boldsymbol{z}} \sum_{i=1}^{N-1} \sum_{j>i}^{N} \sum_{l=1}^{N_{ij}} \|\boldsymbol{e}_{ijl}\|^2 + \|\boldsymbol{e}_{jil}\|^2 \tag{6}$$

with $\boldsymbol{z} = [\boldsymbol{h}_{mnt,1}^T, \ldots, \boldsymbol{h}_{mnt,N}^T]^T$ and additional index l to denote the relative pose pair. By including the error



(a) Vehicle registration parameters

(b) Mounting position parameters

Fig. 2. Transformation parameters for mounting position and registered vehicle pose.

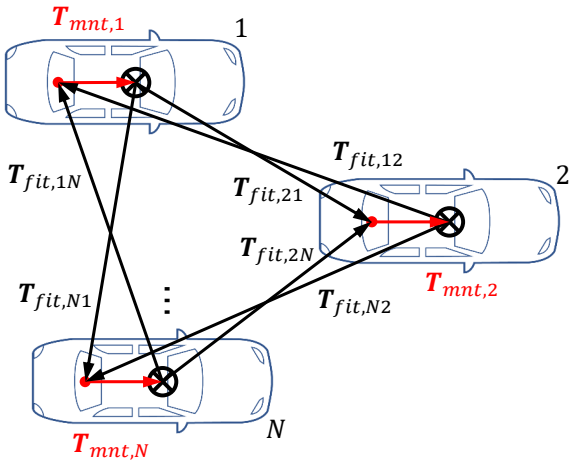


Fig. 3. Transformation relations between vehicles and their sensors.

expressions $\boldsymbol{e}_{ij}$ and $\boldsymbol{e}_{ji}$ from (5) obtained from two different permutations of transforms in our cost function, we treat the errors of both measured transforms $\boldsymbol{T}_{fit,ij}$ and $\boldsymbol{T}_{fit,ji}$ equivalently for the symmetric calibration problem.

D. Nonlinear Least Squares Optimization

The nonlinear least squares problem in (6) is solved by the `lsqnonlin` function in Matlab using the Levenberg–Marquardt algorithm [18], which is an extension to the Gauss–Newton algorithm. As input to the optimization, each entry in $\boldsymbol{e}_{ijl}$ and $\boldsymbol{e}_{jil}$ (except for the constant fourth entry) from (6) is considered individually such that each relative pose pair in (6) provides 6 equations to the nonlinear least squares problem. To increase the robustness with respect to the convergence into local minima, we repetitively perform the optimization task with randomly initialized calibration parameters. Other global optimization strategies might be possible as well, which is not the scope of this work.

III. Estimation of Calibration Accuracy

We derive an accuracy measure for our calibration approach based on the point cloud registration accuracy and the solution of nonlinear least squares problems using the Gauss–Newton algorithm. We assume that the converged Levenberg–Marquardt solution is locally close to a Gauss–Newton solution, which makes our accuracy measure transferable to the Levenberg–Marquardt algorithm, while the Gauss-Newton algorithm is easier to analyze. To this end, we first introduce the Jacobian matrix $\boldsymbol{A}_k$ and residual vector $\boldsymbol{b}_k$ at iteration step k as

$$\boldsymbol{A}_k = \begin{bmatrix} \frac{\partial \boldsymbol{f}_1}{\partial \boldsymbol{z}}\Big|_{\boldsymbol{z}_k} \\ \vdots \\ \frac{\partial \boldsymbol{f}_N}{\partial \boldsymbol{z}}\Big|_{\boldsymbol{z}_k} \end{bmatrix}, \quad \boldsymbol{b}_k = \begin{bmatrix} -\boldsymbol{f}_1(\boldsymbol{z}_k) \\ \vdots \\ -\boldsymbol{f}_{N-1}(\boldsymbol{z}_k) \end{bmatrix}, \tag{7}$$

with

$$\begin{aligned} \boldsymbol{f}_i &= [\tilde{\boldsymbol{f}}_{i(i+1)}^T, \ldots, \tilde{\boldsymbol{f}}_{iN}^T]^T, \\ \tilde{\boldsymbol{f}}_{ij} &= [\hat{\boldsymbol{f}}_{ij1}^T, \ldots, \hat{\boldsymbol{f}}_{ijN_{ij}}^T]^T, \\ \hat{\boldsymbol{f}}_{ijl} &= [\boldsymbol{e}_{ijl}^T, \boldsymbol{e}_{jil}^T]^T, \end{aligned} \tag{8}$$

for which $i = 1, ..., N-1$, $i < j \leq N$, and $l = 1, ..., N_{ij}$. We further recall the Gauss–Newton algorithm [19, pp. 396–399], which can be reformulated to

$$\boldsymbol{z}_{k+1} = \boldsymbol{z}_k + (\boldsymbol{A}_k^T \boldsymbol{A}_k)^{-1} \boldsymbol{A}_k^T \boldsymbol{b}_k. \tag{9}$$

The optimization problem is solved by evaluating (9) iteratively. By assuming an optimal convergence of the algorithm, we estimate the accuracy of our calibration approach by quantifying how the optimum reacts to small changes in the data. Since errors from single data points have a small contribution and lead to small deviations in the vicinity of the optimum, we utilize the first-order approximation to the nonlinear error propagation [20]. This means, we use the local sensitivity of $\boldsymbol{z}^*$ with respect to the registered object poses $\boldsymbol{c}_{ijl} = [\boldsymbol{h}_{fit,ij}^T, \boldsymbol{h}_{fit,ji}^T]_l^T$ to propagate the errors in $\boldsymbol{c}_{ijl}$, yielding

$$\boldsymbol{\Sigma}_{\boldsymbol{z}^*} = \sum_{i=1}^{N-1} \sum_{j>i}^{N} \sum_{l=1}^{N_{ij}} \left(\frac{\partial \boldsymbol{z}^*}{\partial \boldsymbol{c}_{ijl}} \right) \boldsymbol{\Sigma}_{\boldsymbol{c}_{ijl}} \left(\frac{\partial \boldsymbol{z}^*}{\partial \boldsymbol{c}_{ijl}} \right)^T \tag{10}$$

for uncorrelated relative pose pairs $\boldsymbol{c}_{ijl}$, where $\boldsymbol{\Sigma}_{\boldsymbol{c}_{ijl}} = \mathbb{E}\left[(\boldsymbol{c}_{ijl} - \bar{\boldsymbol{c}}_{ijl})(\boldsymbol{c}_{ijl} - \bar{\boldsymbol{c}}_{ijl})^T\right]$ represents the covariance matrix of $\boldsymbol{c}_{ijl}$ and $\boldsymbol{\Sigma}_{\boldsymbol{z}^*}$ the covariance matrix of the calibration parameters.

IV. Evaluation

To validate our approach, we first evaluate the calibration performance using simulation results and later using real sensor data from two test vehicles each equipped with a Hesai Pandar laser scanner mounted on a roof rack. As a rotating sensor with 40 vertical channels, it provides a vertical resolution of 0.33° and 1° depending on the elevation angle and a horizontal resolution of 0.2° (at a rotation rate of 10Hz). The localization data are recorded using an RTK-supported differential GPS system from OxTS. As in [13], both vehicles are BMW 740Li, for which the CAD model is scanned from 5 sides with a resolution of $2cm$ to generate the model point cloud $\mathcal{C}$.

A. Results from Monte Carlo Simulations

Our Monte Carlo simulations are based on our real sensor setup and consider the real-world constraint that the vehicle poses mainly vary in 2D. To compare the results with those of real sensor data, we evaluate them for $N = 2$ vehicles, whereby more vehicles would obviously provide even more accurate results. The relative position of both test vehicles are parameterized according to the description in Fig. 4. Tab. I shows the distributions from which we randomly draw the relative pose parameters for the simulation. Therein, $\mathcal{U}(a, b)$ defines a uniform distribution with bounds a and b, $\mathcal{N}(\mu, \sigma^2)$ denotes a normal distribution with mean μ and variance σ^2, and w_t, w_r represent the translation and rotation errors, which are individually added to the relative pose parameters to account for object registration errors. The low level of assumed registration errors are reasonable since for real sensor data, multiple sensor data frames taken at the same position can be averaged to reduce statistical noise (cf. section IV-B).

TABLE I
Parameter Selection for Monte Carlo Simulations.

Parameter	Variable	Distribution
Longitudinal distance	Δx	$\mathcal{U}(-15m, 15m)$
Lateral distance	Δy	$\mathcal{U}(-15m, 15m)$
Vertical distance	Δz	$\mathcal{U}(-0.2m, 0.2m)$
Pitch difference	$\Delta\theta$	$\mathcal{U}(-2°, 2°)$
Roll difference	$\Delta\phi$	$\mathcal{U}(-2°, 2°)$
Yaw difference	$\Delta\psi$	$\mathcal{U}(-180°, 180°)$
Additional translation error	w_t	$\mathcal{N}(0, \sigma_t^2 = (0.02m)^2)$
Additional orientation error	w_r	$\mathcal{N}(0, \sigma_r^2 = (0.2°)^2)$

We further initiate the calibration parameters randomly inside the parameter bounds such that $\psi_{init}, \phi_{init} \sim \mathcal{U}(-180°, 180°)$, $\theta_{init} \sim \mathcal{U}(-90°, 90°)$, and translations $x_{init}, y_{init}, z_{init} \sim \mathcal{U}(-1m, 1m)$. The optimization is restarted with a resampled set of parameters if the solution vector lies outside the parameter bounds.

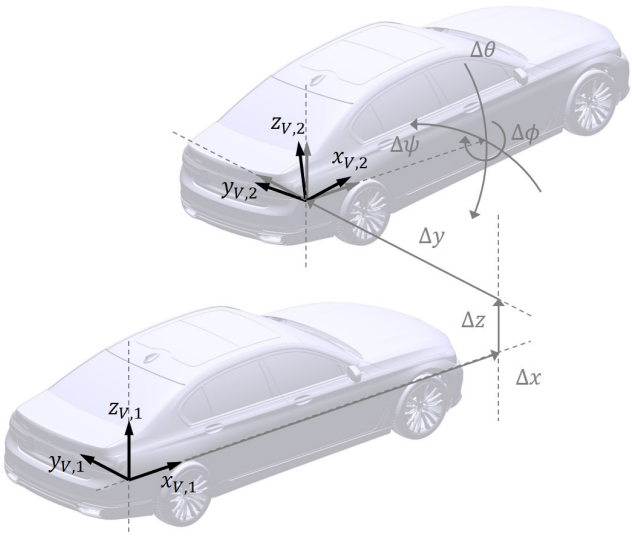


Fig. 4. Parameters for Monte Carlo simulations.

To compare between different simulation configurations, we introduce two commonly used joint error metrics (see [7]), which combines the single translation and rotation errors:

$$\begin{aligned} e_t &= \sqrt{(x_{mnt} - x^*_{mnt})^2 + (y_{mnt} - y^*_{mnt})^2}, \\ e_r &= \arccos((tr(\boldsymbol{R}^*_{mnt}\boldsymbol{R}^{-1}_{mnt}) - 1)/2), \end{aligned} \tag{11}$$

where the variables with * denote the estimated values from the optimization in (6). While e_t accounts for the Euclidean translation error, e_r corresponds to the scalar rotation error around a single rotation axis defined by a 3D vector (aka axis-angle representation). Note that z_{mnt} is not considered in e_t since the relative pose pairs mainly include variations in the 2D space. This leads to a low observability of z_{mnt} in a calibration setup with $N = 2$ vehicles and, thus, causes a high volatility in the z^*_{mnt}-estimation.

1) Accuracy of Calibration Approach: We first evaluate the absolute calibration errors based on our Monte Carlo simulation setup. To this end, we perform 1000 simulations, each with a set of 50 random relative pose pairs. Fig. 5 shows the boxplots of the absolute calibration errors accumulated over both sensors. While the median errors are at a low level, the worst cases are still limited at about 0.2° for rotation and $25mm$ for translation errors. In addition, a more accurate estimation of ψ_{mnt} compared to θ_{mnt} and ϕ_{mnt} can be observed, which might be a result of mainly varying relative pose pairs in 2D.

Besides the absolute errors, we also show the empirical noise (computed as standard deviation over all simulations) of all calibration parameters in Tab. II. As mentioned before, $\sigma_{emp,z}$ tends to be volatile due to its low observability. With the presented simulation setup, empirical standard deviations

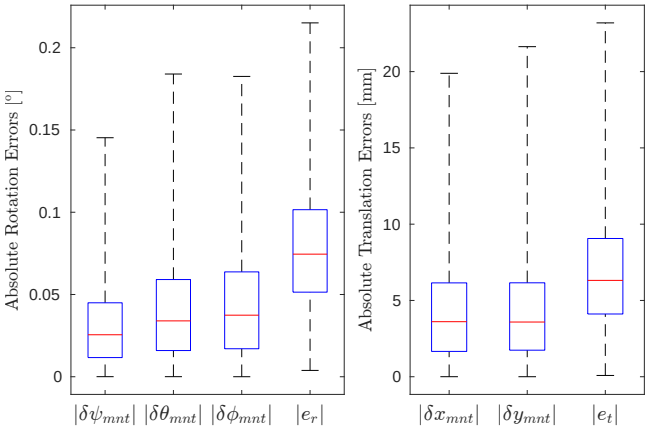


Fig. 5. Absolute translation and rotation errors from Monte Carlo simulation.

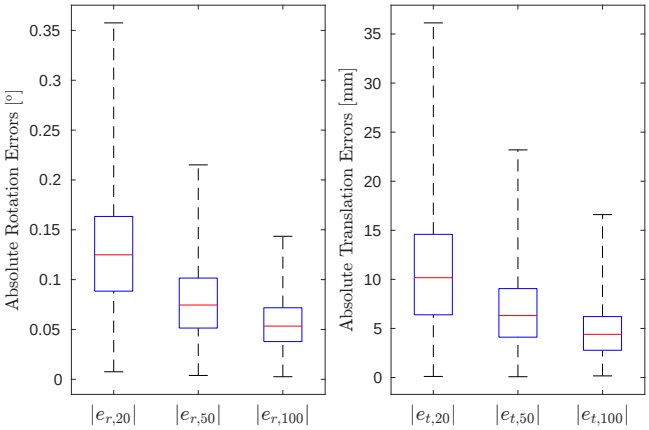


Fig. 6. Influence of number of relative pose pairs on joint error metrics.

of rotation estimations below $0.1°$ and of x_{mnt}-, y_{mnt}-estimations below $1cm$ are achieved.

TABLE II
EMPIRICAL CALIBRATION NOISE FROM MONTE CARLO SIMULATION.

	$\sigma_{\psi,emp}$	$\sigma_{\theta,emp}$	$\sigma_{\phi,emp}$	$\sigma_{x,emp}$	$\sigma_{y,emp}$	$\sigma_{z,emp}$
Vehicle 1	0.039 °	0.053 °	0.054 °	5.31 mm	5.56 mm	172 mm
Vehicle 2	0.039 °	0.054 °	0.055 °	5.57 mm	5.42 mm	172 mm

2) Influence of Number of Relative Pose Pairs: In our previous experiment, we used 50 random relative pose pairs for each simulation to compute the calibration parameters. Since the number of relative pose pairs is constrained for the evaluation with real sensor data, we also show the dependency of the calibration accuracy on this parameter. In Fig. 6, the boxplots for three configurations with 20, 50, and 100 random relative pose pairs are presented. The median and the maximum absolute errors both decrease with a higher number of data pairs. This confirms our intuition that the more relative pose pairs are available, the more accurate the calibration can be performed.

3) Validation of Accuracy Estimation Model: We validate our accuracy model in (10) with the same set of simulations as evaluated in section IV-A.1. Since all 1000 simulations are performed with different sets of relative pose pairs, we would also obtain 1000 different estimated accuracy models. Thus, we cannot simply compute the empirical covariance matrix of the solution vector $\boldsymbol{z}^*$ over all simulations and compare it with the estimated accuracy models. Instead, we normalize the calibration errors with the estimated standard deviation to make them comparable.

Let the standard normal distribution be given by $\mathcal{N}(0,1)$ [21, pp. 172–181]. Each element z_v^* in each sample $\boldsymbol{z}^*$ can be transformed to the standard normal distribution using $z'_v = (z_v^* - \mu_v)/\sigma_{v,est}$. Therein, v represents a calibration variable out of $\{\psi_j, \theta_j, \phi_j, x_j, y_j, z_j\}$ for each vehicle j,

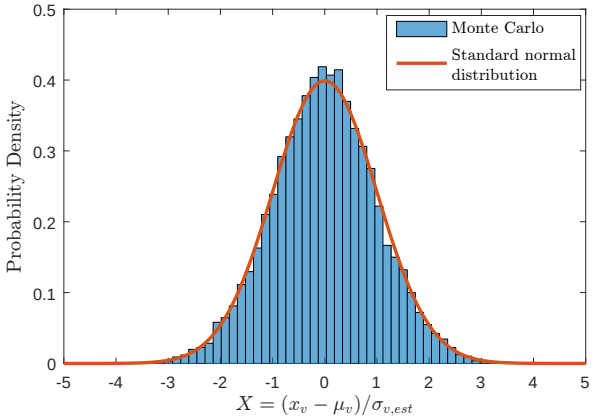


Fig. 7. Comparison of errors from Monte Carlo simulation normalized with true mean value μ_v and estimated standard deviation $\sigma_{v,est}$ to the standard normal distribution.

μ_v the true mean value, and $\sigma_{v,est}$ the estimated standard deviation, which is computed as the square root of the diagonal elements in $\boldsymbol{\Sigma}_{\boldsymbol{z}^*}$ (see (10)). The probability density obtained from the normalized values must fit to that from the standard normal distribution if the estimated standard deviations are correct.

Fig. 7 shows the probability densities of the simulation results considering the calibration parameters from all simulations individually. Since the results of the Monte Carlo simulation fits very well to the standard normal distribution, the applicability of the accuracy model is confirmed by simulations.

B. Results with Real-World Data

Experimental evaluation was performed by a dataset with 80 different relative pose pairs, each consisting of about 100 lidar frames in a row. From each lidar frame, we obtain the vehicle poses as described in section II-B. To increase the robustness against outliers from erroneous registration, we applied RANSAC [22] on the registered poses $\boldsymbol{h}_{fit,ij}$. We further averaged the registered object poses over all n lidar

frames belonging to the same relative pose to reduce the influence of registration noise.

Taking the average of random variables benefits from the reduction of the corresponding noise. The mean $\bar{X}$ of n observations of a scalar normally distributed random variable $X \sim \mathcal{N}(\mu, \sigma^2)$ has a normal distribution [21, p. 233]: $\bar{X} \sim \mathcal{N}(\mu, \sigma^2/n)$. However, for calibration with real sensor data, we further have to include a bias noise term to account for unknown intrinsic calibration errors. While random errors decrease by averaging over a high number of samples, the bias error remains unaffected. Thus, we compute the scalar noise of the registered pose parameters by combining the random and bias noise [23, pp. 15–18]:

$$\sigma^2_{v,avg} = \sigma^2_v/n + \sigma^2_{v,b}, \tag{12}$$

where σ_v is the standard deviation computed from the registered pose variable v (out of $\{\psi_j, \theta_j, \phi_j, x_j, y_j, z_j\}$ for each vehicle j), and $\sigma_{v,b}$ to account for the bias error. As proposed in [23, pp. 15–18], a bias error can be assumed as uniformly distributed from $\mathcal{U}(-b_v, b_v)$ with corresponding standard deviation $\sigma_{v,b} = b_v/\sqrt{3}$. In our computations we use $b_v = 0.2°$ for rotation and $b_v = 2cm$ for translation parameters, which is a reasonable choice for common lidars.

1) Calibration using Subsets of Relative Pose Pairs: The validation of calibration with real sensor data is a challenging task, since commonly, no ground truth is available. Thus, we perform a repeatability analysis, where we separate our dataset into 5 subsets (each with 16 relative pose pairs) to evaluate the calibration statistics. The corresponding empirical calibration noise are listed in Tab. III. Compared to the results of the Monte Carlo simulation in Tab. II, higher standard errors can be observed in general. This can be attributed to the fact that less relative pose pairs were available leading to a smaller number of subsets and a smaller number of relative pose pairs in each subset for calibration. Despite this limitation we achieve angle standard errors around $0.1°$, except for the roll angle of vehicle 1, and standard errors of the x- and y-positions between 5 and $11mm$.

TABLE III
EMPIRICAL CALIBRATION NOISE FROM REAL-WORLD EXPERIMENTS COMPUTED FROM 5 SUBSETS OF RELATIVE POSE PAIRS.

	$\sigma_{\psi,emp}$	$\sigma_{\theta,emp}$	$\sigma_{\phi,emp}$	$\sigma_{x,emp}$	$\sigma_{y,emp}$	$\sigma_{z,emp}$
Vehicle 1	0.051 °	0.103 °	0.305 °	6.72 mm	10.6 mm	623 mm
Vehicle 2	0.096 °	0.108 °	0.127 °	5.46 mm	7.15 mm	617 mm

Analogously to section IV-A.3, we compare the probability density of the normalized errors to that of the standard normal distribution in Fig. 8. Except for the outlier on the left side of the plot, the probabilty density of the normalized errors matches approximately that of the standard normal distribution. Thus, we assume that the use of our accuracy model as well as the choice of our bias error parameters are justified.

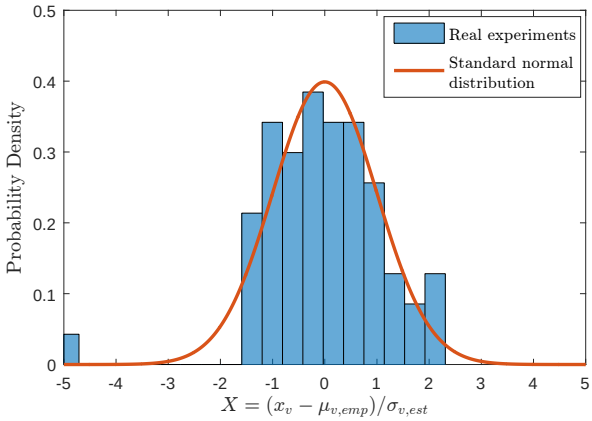


Fig. 8. Comparison of errors from real-world experiments normalized with empirical mean value $\mu_{v,emp}$ and estimated standard deviation $\sigma_{v,est}$ to the standard normal distribution.

2) Calibration Error Estimation with Accuracy Model: To determine the final calibration parameters, we exploit all 80 relative pose pairs at once for the optimization, for which we expect the highest calibration accuracy. Since no empirical noise is available, we estimate the performance using our accuracy model leading to the estimated calibration noise presented in Tab. IV. The results show that our calibration framework provides accurate calibration parameters for the absolute mounting position of a lidar with respect to the vehicle.

TABLE IV
ESTIMATED CALIBRATION NOISE FROM REAL-WORLD EXPERIMENTS.

	$\sigma_{\psi,est}$	$\sigma_{\theta,est}$	$\sigma_{\phi,est}$	$\sigma_{x,est}$	$\sigma_{y,est}$	$\sigma_{z,est}$
Vehicle 1	0.032 °	0.043 °	0.046 °	4.00 mm	4.17 mm	222 mm
Vehicle 2	0.032 °	0.045 °	0.045 °	4.07 mm	4.10 mm	221 mm

V. CONCLUSIONS

This paper proposes an extrinsic calibration method for automotive lidars by using mutually detected vehicle poses. To this end, we match sensor data to CAD models to obtain precise vehicle poses. In addition, we provide an accuracy model to estimate the precision of our calibration method. Both Monte Carlo simulations and real-world experiments demonstrate the benefits of our calibration approach and the accuracy model.

Our proposed methods allow developers to recalibrate sensor mounting positions simultaneously for multiple vehicles. This is particularly beneficial for test vehicles since mounting positions can change frequently and are more prone to slight slips. In the future, we would like to integrate a further transform into the cost function, which is obtained by matching the sensor points originating from the static

environment. This would make z_{mnt} observable even if the calibration setup only includes two vehicles.

REFERENCES

[1] F. M. Mirzaei, D. G. Kottas, and S. I. Roumeliotis, "3D LIDAR-camera intrinsic and extrinsic calibration: Identifiability and analytical least-squares-based initialization," *The International Journal of Robotics Research*, vol. 31, no. 4, pp. 452–467, 2012.

[2] M. He, H. Zhao, J. Cui, and H. Zha, "Calibration method for multiple 2D LIDARs system," in *Proc. of the IEEE International Conference on Robotics and Automation*, 2014, pp. 3034–3041.

[3] O. Khan, R. Bergelt, and W. Hardt, "On-field mounting position estimation of a lidar sensor," in *Proc. of SPIE 10431, Remote Sensing Technologies and Applications in Urban Environments II*, 2017.

[4] J. Underwood, A. Hill, and S. Scheding, "Calibration of range sensor pose on mobile platforms," in *Proc. of the IEEE International Conference on Intelligent Robots and Systems*, 2007, pp. 3866–3871.

[5] J. Levinson and S. Thrun, "Unsupervised calibration for multi-beam lasers," in *Experimental Robotics*, ser. Springer Tracts in Advanced Robotics. Springer, 2014, vol. 79, pp. 179–193.

[6] M. Sheehan, A. Harrison, and P. Newman, "Self-calibration for a 3D laser," *The International Journal of Robotics Research*, vol. 31, no. 5, pp. 675–687, 2012.

[7] J. Kümmerle, T. Kühner, and M. Lauer, "Automatic calibration of multiple cameras and depth sensors with a spherical target," in *Proc. of the IEEE International Conference on Intelligent Robots and Systems*, 2018, pp. 5584–5591.

[8] M. He, H. Zhao, F. Davoine, J. Cui, and H. Zha, "Pairwise LIDAR calibration using multi-type 3D geometric features in natural scene," in *Proc. of the IEEE International Conference on Intelligent Robots and Systems*, 2013, pp. 1828–1835.

[9] M. Feuerstein. Hand-eye calibration. [Online]. Available: http://campar.in.tum.de/Chair/HandEyeCalibration [Accessed: 17.06.2020]

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

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

[12] K. Strobl and G. Hirzinger, "Optimal hand-eye calibration," in *Proc. of the IEEE International Conference on Intelligent Robots and Systems*, 2006, pp. 4647–4653.

[13] E. Ye and M. Althoff, "Model-based offline vehicle tracking in automotive applications using a precise 3D model," in *Proc. of the 22nd IEEE International Conference on Intelligent Transportation Systems*, 2019, pp. 1128–1135.

[14] 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, no. 2, pp. 239–256, 1992.

[15] Y. Chen and G. Medioni, "Object modelling by registration of multiple range images," *Image and Vision Computing*, vol. 10, no. 3, pp. 145–155, 1992.

[16] P. Bergström and O. Edlund, "Robust registration of point sets using iteratively reweighted least squares," *Computational Optimization and Applications*, vol. 58, no. 3, pp. 543–561, 2014.

[17] F. Pomerleau, F. Colas, and R. Siegwart, "A review of point cloud registration algorithms for mobile robotics," *Foundations and Trends in Robotics*, vol. 4, no. 1, pp. 1–104, 2015.

[18] J. J. Moré, "The Levenberg-Marquardt algorithm: Implementation and theory," in *Numerical Analysis*, ser. Lecture Notes in Mathematics, G. A. Watson, Ed. Springer, 1978, vol. 630, pp. 105–116.

[19] Björck Ake, *Numerical Methods in Matrix Computations*, ser. Texts in Applied Mathematics. Springer, 2015, vol. 59.

[20] K. O. Arras, "An introduction to error propagation: Derivation, meaning and examples of equation Cy = Fx Cx Fx'," 1998.

[21] R. E. Walpole, R. H. Myers, S. L. Myers, and K. Ye, *Probability & statistics for engineers & scientists: MyStatLab update*, 9th ed. Boston: Pearson, 2017.

[22] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[23] H. W. Coleman and W. G. Steele, *Experimentation, validation, and uncertainty analysis for engineers*, 4th ed. John Wiley & Sons, 2018.