

Model-based Offline Vehicle Tracking in Automotive Applications Using a Precise 3D Model

Egon Ye¹ and Matthias Althoff²

Abstract—Object tracking aims at estimating the state of moving objects based on remote measurements. To evaluate online algorithms in automotive systems, ground truth data must be acquired, which is a time-consuming and expensive approach. We propose a novel offline approach to generate ground truth data from existing sensor measurements using CAD models. In our approach, we provide error bounds for the localization of the objects based on the measurement noise of a single laser beam and the sensitivity of the point cloud registration. To estimate accurate kinematic states of the vehicle, we apply an extended Rauch-Tung-Striebel smoother on the stored measurements. In experiments with real sensor data, we demonstrate that the performance of the proposed approach is superior to DGPS within the near range.

I. INTRODUCTION

In order to realize higher levels of autonomy, autonomous vehicles need increasingly precise environment models. For the validation of these models, one requires ground truth data [1], whose quality obviously has to improve with the increased quality demands of environment models. It is foreseeable that classical methods for obtaining ground truth data are quickly reaching their limits, e.g., differential global positioning systems (DGPS) for localization in urban areas [2]. Thus, this paper investigates a new method: By tracking vehicles for which a precise 3D model is given, we can obtain ground truth data of unmatched accuracy. Intuitively, the high precision is achieved by excluding two sources of uncertainty. First, the tracking is not done using estimated 3D models; second, by matching many laser points on the surface of the 3D model, we drastically reduce the measurement error of a single laser beam. An application of precise 3D models generated from a CAD model to track vehicles has not yet been published to the best knowledge of the authors.

Online vs. Offline Tracking: In a robotic system like an autonomous vehicle, environment perception algorithms are constrained by real-time requirements and causal signal processing. However, these restrictions are not relevant for an offline-generated ground truth model using recorded sensor data. By considering future measurements for the state estimation at each time step, smoothing algorithms can be used for a more accurate environment representation. A further benefit is that iterative procedures can be involved, e.g., to subsequently improve object localization and dynamic state

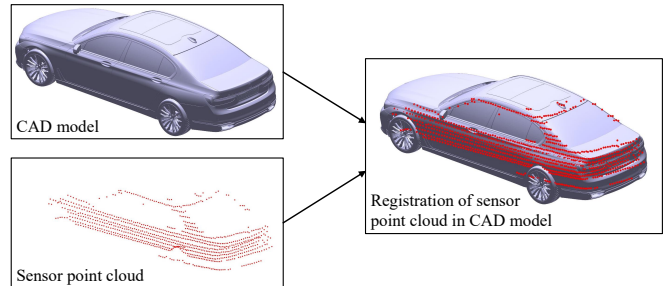


Fig. 1. Vehicle tracking using CAD model and laser scanner data.

estimation. The work in [3] presents an offline concept to extract vehicle trajectories from recorded lidar data. Thereby, objects are initialized based on the number of assigned points as well as the conformance to L-, I- and U-shapes and tracked in both the future and the past. An extended Rauch-Tung-Striebel smoother [4] is applied on the measurements of the entire track to obtain the final state estimates.

Tracking Using Laser Scanners: Various sensors like radars, laser scanners, cameras and ultrasonic sensors can be deployed for object tracking in automotive applications. Each of these exploits a specific sensory modality necessitating different signal processing algorithms. Regarding ground truth generation of object data, laser scanners are preferably deployed due to their precise and dense position measurements, which enable accurate pose estimations [3], [5]. In the literature, most laser-scanner-based tracking approaches are dedicated to online applications. These can be broadly subdivided into model-free and model-based tracking concepts.

Model-free Tracking: Model-free algorithms track objects independent of an object class. These generic approaches are designed for objects with arbitrary shapes and sizes. The work in [6] proposes an approach that tracks moving objects relying on motion cues between frames determined by a RANSAC operation with SHOT descriptors. An object-class-independent concept has been presented in [7] that is based on accumulating point cloud data over time for each object in combination with the ICP algorithm [8]. In [9], a joint state representation of the sensor pose, the static background, and the motion states of dynamic objects has been established for model-free object tracking. In general, due to the variations of trackable objects, model-free concepts typically exhibit lower tracking accuracies compared to model-based concepts.

Model-based Tracking: In contrast, model-based algorithms assume the classes of tracked objects to be known.

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

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

They exploit prior knowledge about the shape of tracked objects as well as the corresponding motion model for the filtering process. For vehicle tracking, rectangular shapes are assumed in [10] and L-, I- and U-shapes in [11]. In this work, we use precise CAD models. Matching 3D point measurements to CAD models is a task that is commonly applied in the manufacturing industry, e.g., for surface precision inspection [12] and pose estimation for robotic grasping [13], [14]. By sampling the surface of the CAD model, its solution can be reduced to that of a point cloud registration problem.

Point Cloud Registration: Point cloud registration is the process to find an appropriate spatial transformation between two point clouds. A registration algorithm is called rigid if the distances between any two points are assumed invariable, which holds for CAD models and laser scanner data. Most approaches for the registration problem iteratively match points and minimize a distance measure calculated from these point matches to determine a transformation estimate [15]. The two most popular algorithms are the point-to-point method, also called the iterative closest point (ICP) algorithm [8], and the point-to-plane method [16]. While the ICP algorithm applies the Euclidean distance measures, the point-to-plane method defines the distance measure by the perpendicular distance of a point to the local surface of the matched point. An extension aiming at the reduction of outliers in the correspondences is obtained by the minimization of a robust criterion function [17].

Contribution: Contrary to existing work, such as [3], we focus on the generation of highly precise object trajectories for the validation of online algorithms. Therefore, we propose a model-based offline tracking approach to localize and track surrounding vehicles using a precise 3D point cloud model (deduced from CAD models) and show its potential for ground truth generation. We further present a formulation to estimate the measurement accuracies for tracking, derived from the sensitivity of the point cloud registration algorithm. In our experiments with real sensor data, we show that within near range, our approach achieves very good performance in tracking single objects without occlusions.

Overview of the Paper: The remainder of the paper is organized as follows: Section II shows an overview of our offline vehicle tracking concept using a CAD model. A more detailed insight into the point cloud registration algorithm is given in section III. Section IV presents the measurement noise model for tracking that includes the properties of the local surface geometry. In section V, the proposed concept is evaluated on real sensor data.

II. OVERVIEW OF OFFLINE TRACKING PROCEDURE

Since the goal of this work is to investigate achievable accuracies of a model-based tracking approach, we focus on processing a single object. An overview of our tracking framework is depicted in Fig. 2. Below, we describe the components and main steps of our approach.

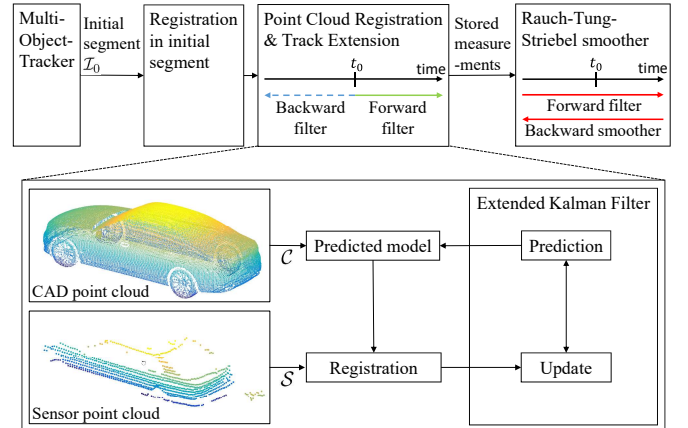


Fig. 2. Overview of single vehicle tracking framework with point clouds from CAD model and laser scanner sensor.

A. Point Cloud from CAD Model

For the point cloud registration process [16], the 3D model data must be given in the form of sampled surface points. In particular, given the stereolithography (STL) file format describing a triangle mesh, one can obtain surface points by calculating intersections of single rays with surface triangles [18]. The set of points sampled from the CAD model in vehicle-fixed coordinates relative to the center of the rear axle is denoted by \mathcal{C} .

B. Multi-Object-Tracker and Object Initialization

To obtain a rough pose and velocity estimation before using a precise 3D model, we apply a baseline multi-object-tracking (MOT) approach as a preprocessing step that tracks the centroids of segmented lidar point clouds (see Fig. 2). For segmentation we use a distance-based clustering algorithm as shown in [19]. However, 3D point clouds acquired by a multilayer laser scanner contain a great number of ground reflections, which impedes an appropriate segmentation. Since there exist many approaches addressing this problem (e.g., [20], [21]), we assume that the ground points are already removed in this work. The centroids computed from the clustered point clouds are associated to a simple MOT following the global nearest neighbor principle [22], whereby we use the object management from [23]. The outcome of the MOT is a list with object hypotheses consisting of their assigned segment points and dynamic states over time.

We select the time step t_0 in the measurement sequence, where a favorable observation of the real object is assumed. In addition, we define the assigned segment points of the object hypothesis (corresponding to the tracked object) at t_0 as the initial segment \mathcal{I}_0 (see Fig. 2). Due to the spatial proximity of \mathcal{I}_0 to the ego vehicle, \mathcal{I}_0 provides a dense point cloud of the object allowing for high accuracies in the subsequent matching step. The initial object pose is established by fitting \mathcal{I}_0 in \mathcal{C} using prior knowledge from the MOT for choosing the initial transformation and robust point cloud registration (see Section III).

C. Point Cloud Registration and State Estimation

Given the initial object pose, we estimate its dynamic states separately in a forward and a backward filter (see Fig. 2) using an extended Kalman filter (EKF), which predicts the object state in each time step to provide an initial transformation for the point cloud registration.

We choose the coordinated turn (aka kinematic single-track [24]) model as our motion model, which is commonly used to represent a ground-based vehicle in tracking applications [25]. It describes a 2D motion on a horizontal plane, where the yaw (or heading) angle ψ equals the kinematic turn angle since tire slip is neglected. The discrete-time model for constant acceleration and yaw rate is

$$\begin{bmatrix} x \\ y \\ v \\ a \\ \psi \\ \omega \end{bmatrix}_{k+1} = \begin{bmatrix} x \\ y \\ v \\ a \\ \psi \\ \omega \end{bmatrix}_k + \int_{kT}^{(k+1)T} \begin{bmatrix} v(t) \cos(\psi(t)) \\ v(t) \sin(\psi(t)) \\ a(t) \\ 0 \\ \omega(t) \\ 0 \end{bmatrix} dt + \begin{bmatrix} 0 \\ 0 \\ 0 \\ w_a \\ 0 \\ w_\omega \end{bmatrix}_k, \quad (1)$$

where $w_a \sim \mathcal{N}(0, \sigma_a^2)$ and $w_\omega \sim \mathcal{N}(0, \sigma_\omega^2)$ are white noise signals accounting for unknown inputs and somewhat model errors, ω represents the yaw rate, x and y the mid-position of the vehicle rear axle (world coordinates), v the velocity, and a the longitudinal (vehicle-fixed coordinates) acceleration. The EKF is initialized by the initial object pose and the velocity estimate from the MOT.

The predicted object state of the EKF is used to transform the CAD model point cloud \mathcal{C} in every time step (see Fig. 2). Point cloud registration is then applied on the predicted \mathcal{C} and the acquired lidar data to obtain the transformation function between them (see Section III). We thereby restrict the point cloud from the laser scanner to a gated sphere around the predicted position to reduce computational costs. The 2D position and orientation measurement for updating the object state is computed from the predicted position and the transformation function. An updated object state contributes to the extension of an object track. The track extension is terminated if the likelihood of the measured 2D pose, given the previous estimated motion states, falls below a threshold for several time steps.

Similar to [3], we apply the extended Rauch-Tung-Striebel smoother (ERTS) [4] on the stored pose measurements from both the forward and the backward filter (see Fig. 2).

III. REGISTRATION OF 3D MODEL POINT CLOUD IN LASER SCANNER DATA

In this section, we present the considered robust point cloud registration algorithm [17] for our application to track vehicles in laser scanner data. We refer to the sensor point cloud \mathcal{S} as the point cloud to be registered and the CAD point cloud \mathcal{C} as the target. We outline the two main steps of the iteratively solved registration problem, namely the corresponding point search and the estimation of the rigid (body) transformation. We prefer to use the point-to-plane [16] over the ICP [8] method due to faster convergence

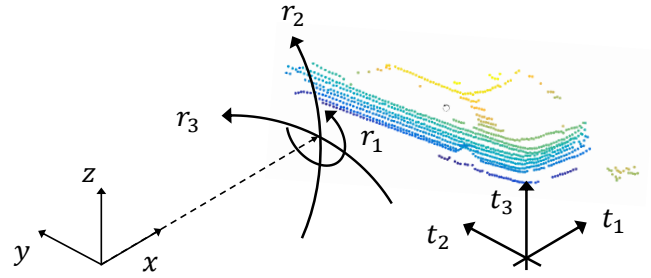


Fig. 3. Variables for rigid transformation.

and since it is more appropriate for matching point clouds representing object surfaces.

A. Correspondence Search

In each iteration of the registration algorithm, point pairs must be determined between the point clouds \mathcal{C} and \mathcal{S} to compute an error measure. Let $\{\mathbf{p}_i\}_{i=1}^N$ be the set of N points $\mathbf{p}_i \in \mathbb{R}^3$ from \mathcal{S} . We assign to each data point \mathbf{p}_i a corresponding point $\mathbf{y}_i \in \mathbb{R}^3$ from \mathcal{C} according to their Euclidean distance to each other. This results in the set of correspondences $\mathcal{K} = \{(\mathbf{p}_i, \mathbf{y}_i) | \mathbf{p}_i \in \mathcal{S}, \mathbf{y}_i = \arg \min_{\mathbf{y} \in \mathcal{C}} \|\mathbf{p}_i - \mathbf{y}\|\}$.

B. Estimation of Rigid Transformation

The estimation of the rigid transformation using the point-to-plane method is an established procedure. However, we briefly review its basic ideas for completeness and to provide preliminaries for the derivation of the accuracy model in Section IV.

1) *Rigid Transformation*: The rigid (aka Euclidean) transformation of a point \mathbf{p}_i is given by $\mathbf{R}\mathbf{p}_i + \mathbf{t}$, where $\mathbf{R} \in \mathbb{R}^{3 \times 3}$ represents an orthonormal rotation matrix with $\det(\mathbf{R}) = 1$ and $\mathbf{t} \in \mathbb{R}^3$ is a translation vector. Let the parameterization of the rigid transformation be represented by the vector (see Fig. 3)

$$\mathbf{z} = [r_1, r_2, r_3, t_1, t_2, t_3]^T. \quad (2)$$

Using \mathbf{z} , the corresponding rotation matrix is defined as:

$$\mathbf{R}(\mathbf{z}) = \begin{bmatrix} c_2 c_3 & s_1 s_2 c_3 - c_1 s_3 & c_1 s_2 c_3 + s_1 s_3 \\ c_2 s_3 & s_1 s_2 s_3 + c_1 c_3 & c_1 s_2 s_3 - s_1 c_3 \\ -s_2 & s_1 c_2 & c_1 c_2 \end{bmatrix}, \quad (3)$$

where the sin and cos of angles r_1 , r_2 and r_3 are abbreviated by s , c with index 1, 2, 3. These angles correspond to a sequential rotation around the x -, y - and z -axes and are limited by $-\pi < r_1, r_3 \leq \pi$, and $-\pi/2 \leq r_2 \leq \pi/2$. The translation vector is parameterized as $\mathbf{t}(\mathbf{z}) = [t_1, t_2, t_3]^T$, where t_i stands for the translations in x , y and z directions.

2) *Point-to-plane Method*: The point-to-plane method [16] computes an error measure between the correspondences in \mathcal{K} by the perpendicular distance of \mathbf{p}_i to the local surface of the closest point \mathbf{y}_i . Using the unit normal vector \mathbf{n}_i corresponding to the surface at \mathbf{y}_i , we have to minimize

$$f(\mathbf{R}, \mathbf{t}) = \sum_{i=1}^N \rho(\mathbf{n}_i^T (\mathbf{R}\mathbf{p}_i + \mathbf{t} - \mathbf{y}_i)), \quad (4)$$

where $\rho(r)$ is a robust criterion function that is applied to reduce the influence of outliers in r as shown in [17]. The optimization of (4) can be further reduced to a nonlinear least squares problem, in which the function to be minimized is [17]

$$g(\mathbf{R}, \mathbf{t}) = \sum_{i=1}^N w_i (\mathbf{n}_i^T (\mathbf{R}\mathbf{p}_i + \mathbf{t} - \mathbf{y}_i))^2, \quad (5)$$

where the weight factors w_i are computed for each element of the sum in (4) by $w(r) = \rho(r)/r$. An approximated solution to the nonlinear least squares problem is obtained using the Gauss-Newton algorithm [26, pp. 67-68], which yields the approximated cost function

$$g(\mathbf{z}) \approx \sum_{i=1}^N w_i ((\mathbf{p}_i - \mathbf{y}_i)^T \mathbf{n}_i + [(\mathbf{p}_i \times \mathbf{n}_i)^T, \mathbf{n}_i^T] \mathbf{z})^2. \quad (6)$$

The minimization of (6) is performed according to a weighted linear least squares problem, which yields

$$\mathbf{z}^* = (\mathbf{A}_N^T \mathbf{W} \mathbf{A}_N)^{-1} \mathbf{A}_N^T \mathbf{W} \mathbf{b} \quad (7)$$

with

$$\mathbf{A}_N = \begin{bmatrix} (\mathbf{p}_1 \times \mathbf{n}_1)^T & \mathbf{n}_1^T \\ \vdots & \vdots \\ (\mathbf{p}_N \times \mathbf{n}_N)^T & \mathbf{n}_N^T \end{bmatrix}, \mathbf{b} = \begin{bmatrix} (\mathbf{y}_1 - \mathbf{p}_1)^T \mathbf{n}_1 \\ \vdots \\ (\mathbf{y}_N - \mathbf{p}_N)^T \mathbf{n}_N \end{bmatrix}, \quad (8)$$

and $\mathbf{W} = \text{diag}(w_1, \dots, w_N)$. It must be noted that due to the approximation of the cost function, the convergence of the iterative point-to-plane registration algorithm is not guaranteed.

For a more robust registration process, we further use the Levenberg-Marquardt optimization [27] to constrain the transformations of r_1 , r_2 and t_3 more restrictively since vehicle motion is mainly described in a 2D space.

C. Variation of Initial Transformation

The choice of the initial transformation has a significant influence on the probability to converge into the global minimum. For the offline processing procedure considered in this work, we distinguish between the registration for object initialization and for measurement updates.

In the first case, \mathcal{C} is translated such that the center points of both point clouds \mathcal{C} and $\mathcal{S} = \mathcal{I}_0$ coincide. \mathcal{C} is additionally rotated around the yaw axis by exploiting the direction of the object velocity vector estimate from the MOT. Taking this as the starting point \mathbf{z}_{base} , we first perform a systematic grid search with offset parameter vectors dt_1 , dt_2 , dt_3 , and dr_3 that span a grid around \mathbf{z}_{base} as presented in Algorithm 1. To reduce the grid dimensionality, offsets of the pitch and roll angles (dr_2 and dr_1) are neglected due to a less significant effect for the 2D tracking problem. A local search procedure based on stochastic hill climbing [28, p. 124] with sampling from a uniform distribution is applied on the best (with respect to f_i) results $\mathcal{Z} \subseteq \mathcal{A}$ to find the final pose estimation.

Algorithm 1 Grid Search

Input: \mathbf{z}_{base} , dt_1 , dt_2 , dt_3 , $dr_1 = 0$, $dr_2 = 0$, dr_3 , \mathcal{C} , \mathcal{S}
1: $G \leftarrow \text{buildNDimensionalGrid}(dt_1, dt_2, dt_3, dr_1, dr_2, dr_3)$
2: **for all** grid nodes $d\mathbf{z} \in G$ **do**
3: $\mathbf{z}_{init} \leftarrow \mathbf{z}_{base} + d\mathbf{z}$
4: $\mathbf{z}, f \leftarrow \text{PointCloudRegistration}(\mathcal{S}, \mathcal{C}, \mathbf{z}_{init})$
5: $\mathcal{A} \leftarrow \mathcal{A} \cup \{(\mathbf{z}, f)\}$
6: **end for**
7: **return** $\mathcal{A} = \{(\mathbf{z}, f)_i\}_{i=1}^{|G|}$

Regarding registrations for measurement updates, we only apply the local search based on a set of starting points uniformly sampled from the 3σ -range of the predicted object position and yaw angle variances.

IV. ACCURACY ESTIMATION OF PROPOSED APPROACH

In this section, we derive the achievable accuracy of the registration algorithm, which represents the measurement noise component in the tracking process. As motivated in [29, pp. 120-144], we estimate the registration accuracy based on the sensitivity measure of the cost function in (6) towards a change of \mathbf{z} . By assuming a perfect registration process (corresponding to the minimum of the cost function in (6)), the proposed approach uses the number of correspondences to refine the estimation accuracy. We further exclude outliers and limit \mathcal{K} to the set of $\tilde{N} \leq N$ actual correspondences in the final registration iteration. The resulting set is defined by $\tilde{\mathcal{K}} = \{(\mathbf{p}_i, \mathbf{y}_i) \in \mathcal{K} \mid d(\mathbf{p}_i, \mathbf{y}_i) < d_{min}\}$ with a distance threshold d_{min} .

A. Sensitivity to Transformation

Referring to the registration constraint analysis in [30, pp. 51-56], the sensitivity Ψ of the registration error towards \mathbf{z} is computed using the gradient of the distance measure (argument of the squared expression in cost function (6)) to \mathbf{z} :

$$\Psi = \mathbf{A}_{\tilde{N}}^T \mathbf{A}_{\tilde{N}} = \mathbf{V} \mathbf{\Lambda} \mathbf{V}^T, \quad (9)$$

where $\mathbf{A}_{\tilde{N}}$ is defined as in (8). $\mathbf{\Lambda}$ and \mathbf{V} consist of the eigenvalues and eigenvectors of Ψ , respectively. The eigenvalues represent the relative sensitivity of the estimated transformation in the directions given by the eigenvectors. The linear assumption for sensitivity analysis is justified since \mathbf{z} of the final registration iteration is presumed to be close to its true value. The work in [29, pp. 120-144] has further decoupled Ψ for translations and rotations to consider the registration errors separately yielding

$$\begin{aligned} \Psi_t &= \sum_i^{\tilde{N}} \mathbf{n}_i \mathbf{n}_i^T = \mathbf{V}_t \mathbf{\Lambda}_t \mathbf{V}_t^T, \\ \Psi_r &= \sum_i^{\tilde{N}} (\mathbf{p}_i \times \mathbf{n}_i) (\mathbf{p}_i \times \mathbf{n}_i)^T = \mathbf{V}_r \mathbf{\Lambda}_r \mathbf{V}_r^T. \end{aligned} \quad (10)$$

B. Position and Orientation Accuracies

It is well-known that 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 [31, p. 233]:

$$\bar{X} \sim \mathcal{N}(\mu, \sigma^2/n). \quad (11)$$

Thus, every single measurement equally accounts for an increase of the estimation accuracy. However, regarding the transformation vector \mathbf{z} , this depends on the geometric properties of the underlying surfaces to be matched. From Fig. 4a, it is evident that correspondences between both surfaces mainly refine the distance parallel to \mathbf{n} but not a shift perpendicular to \mathbf{n} . Likewise, correspondences between both surfaces mainly refine rotational transformations (see Fig. 4b and 4c) if they cause a shift perpendicular to the local surface, like a rotation around the z -axis in Fig. 4c. These considerations correspond to the definitions of the sensitivity measures $\Psi_{\mathbf{t}}$ and Ψ_r .

Following the explanations in [29, pp. 120-144], we consider both point clouds \mathcal{C} and \mathcal{S} as sampled from the actual surface of an object including sampling errors represented by isotropic position error variances $\sigma_{\mathcal{C}}^2$ and $\sigma_{\mathcal{S}}^2$. Thereby, $\sigma_{\mathcal{C}}^2$ is modeled depending on the object shape, e.g., indentations, and the sampling resolution of the CAD model. $\sigma_{\mathcal{S}}^2$ refers to the measurement noise of a single laser beam.

For a special case as depicted in Fig. 5a, where $\Psi_{\mathbf{t}}$ is diagonal, [29, pp. 120-144] has derived the expressions for the translation errors that are scaled by the diagonal elements of $\Psi_{\mathbf{t}}$. This yields: $\sigma_x^2 = \sigma_{\mathcal{C}\mathcal{S}}^2 / \sum_i n_{x,i}^2$, $\sigma_y^2 = \sigma_{\mathcal{C}\mathcal{S}}^2 / \sum_i n_{y,i}^2$ and $\sigma_z^2 = \sigma_{\mathcal{C}\mathcal{S}}^2 / \sum_i n_{z,i}^2$, where $\sigma_{\mathcal{C}\mathcal{S}}^2$ is defined as the sum of $\sigma_{\mathcal{C}}^2$ and $\sigma_{\mathcal{S}}^2$. It is obvious in this special case that the diagonal elements of $\Psi_{\mathbf{t}}$ are the numbers of correspondences having an n_i pointing to the respective direction. For object surfaces with arbitrary n_i (see Fig. 5b), $\Psi_{\mathbf{t}}$ is not diagonal. Thus, the precision calculation is performed in the eigenspace of $\Psi_{\mathbf{t}}$: $\sigma_{\lambda_1}^2 = \sigma_{\mathcal{C}\mathcal{S}}^2 / \lambda_1$, $\sigma_{\lambda_2}^2 = \sigma_{\mathcal{C}\mathcal{S}}^2 / \lambda_2$ and $\sigma_{\lambda_3}^2 = \sigma_{\mathcal{C}\mathcal{S}}^2 / \lambda_3$, where λ_1 , λ_2 and λ_3 are the eigenvalues in $\Lambda_{\mathbf{t}}$.

Additionally to [29], where the focus lies on a general error bound established on $\sigma_{\lambda_3}^2$, we further transform the noise expressions back to the ego vehicle fixed coordinate system to obtain the position measurement noise component for the EKF:

$$\begin{aligned} \mathbf{Q}_{\mathbf{t}} &= \begin{bmatrix} \sigma_x^2 & \sigma_{xy}^2 & \sigma_{xz}^2 \\ \sigma_{xy}^2 & \sigma_y^2 & \sigma_{yz}^2 \\ \sigma_{xz}^2 & \sigma_{yz}^2 & \sigma_z^2 \end{bmatrix} \\ &= \mathbf{V}_{\mathbf{t}} \text{diag}(\sigma_{\lambda_1}^2, \sigma_{\lambda_2}^2, \sigma_{\lambda_3}^2) \mathbf{V}_{\mathbf{t}}^T \\ &= \mathbf{V}_{\mathbf{t}} \sigma_{\mathcal{C}\mathcal{S}}^2 \Lambda_{\mathbf{t}}^{-1} \mathbf{V}_{\mathbf{t}}^T \\ &= \sigma_{\mathcal{C}\mathcal{S}}^2 \Psi_{\mathbf{t}}^{-1}. \end{aligned} \quad (12)$$

Concerning the rotational components, we are only interested in the noise component of the yaw angle for the 2D problem. Being a scalar component, no transformation into the eigenspace is considered. Thus, we directly adopt the expression as derived in [29, p. 138]:

$$\sigma_{\psi}^2 = \frac{\sigma_{\mathcal{C}\mathcal{S}}^2}{\sum_{i=1}^{\bar{N}} (y_{i,1} n_{i,2} - y_{i,2} n_{i,1})^2}. \quad (13)$$

As can be observed, the denominator is deduced from the third diagonal element of Ψ_r . This reflects that a higher sensitivity leads to a higher precision. However, point clouds with larger distances from the origin result in higher accuracies according to (13). To reduce this distance dependency, we propose to refer to a fixed reference point (e.g., the mid of the minimum enclosing box of the CAD model) as the origin for the computation of (13).

V. EVALUATION

For real data acquisition we use a Hesai Pandar laser scanner mounted on a roof rack of the ego vehicle. The sensor provides 40 vertical channels, which have a resolution of 0.33° and 1° depending on the angle ranges. By sensor rotation around the vertical axis, a 360° view is provided with a horizontal resolution of 0.2° (at 10Hz rotation rate). For validation and ego-motion compensation, the motion data of ego and object vehicle are both acquired using a RTK-supported DGPS system from OxTS.

As the object to be tracked, we consider a BMW 740Li vehicle, for which the CAD model is given. To extract point cloud data, the CAD model surface is scanned from 5 sides (from left, right, above, front, and back), each by parallel rays in a grid pattern with a fixed spacing of 2cm. As a result, a point cloud with around 60000 surface points is generated.

A. Theoretic Registration Accuracies

To show the theoretic accuracies provided by our approach, we compute (12) and (13) for simulated sensor data. Based on the mounting position and the scanning properties of the real sensor, simulated sensor data are generated for different static pose setups of the object relative to the ego vehicle (see Fig. 6). Thereby, points referring to window panes are removed because glass does not sufficiently reflect laser beams. Since the exact position of the object is known, point cloud registration is not considered. We further choose $\sigma_{\mathcal{C}} = 2\text{cm}$ to account for the discretization (based on sample spacing) of the CAD model and $\sigma_{\mathcal{S}} = 5\text{cm}$ as the upper bound for the measurement noise of a single laser beam.

The computed theoretic accuracies are presented in Fig. 7. Obviously, the accuracies increase with a smaller x_{obj} , which relates to a decrease of reflected points. Exceptions are visible for $x_{obj} = 5\text{m}$, which originate from the high mounting position of the laser scanner combined with its limited vertical field of view. σ_x exhibits generally higher accuracies compared to σ_y since in our simulation setup, most points have a larger normal vector component in x - than in y -direction (in sensor coordinate system). While σ_x provides higher accuracies when ψ_{obj} is around 90° (transverse position of the object vehicle), σ_y benefits from ψ_{obj} around 45° or 135° (diagonal position of the object vehicle) due to a higher portion of normal vector components in y -direction. Regarding σ_{ψ} , higher accuracies originate from a higher sensitivity to a yaw rotation, which prevails if the sensor detects the longer vehicle sides compared to the front or rear.

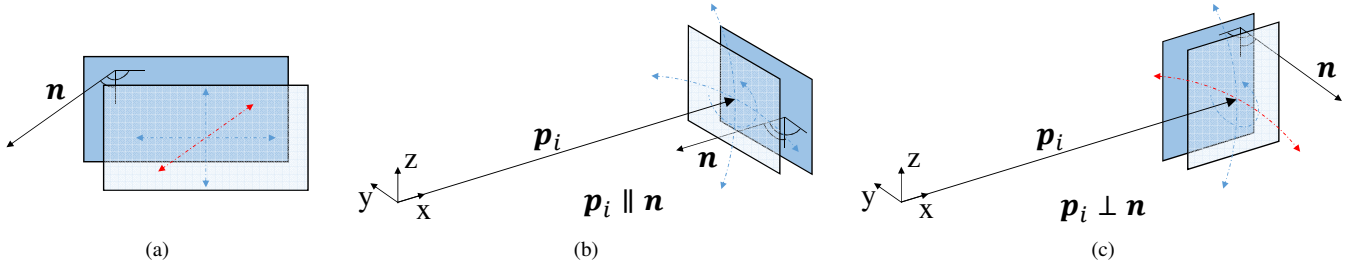


Fig. 4. Visualization of sensitivity for registering the light on the dark surface: (a) Sensitivity only to translational transformation perpendicular to local surface, (b) no sensitivity to rotational transformation causing a shift parallel to the local surface, (c) sensitivity to rotational transformation around z-axis.

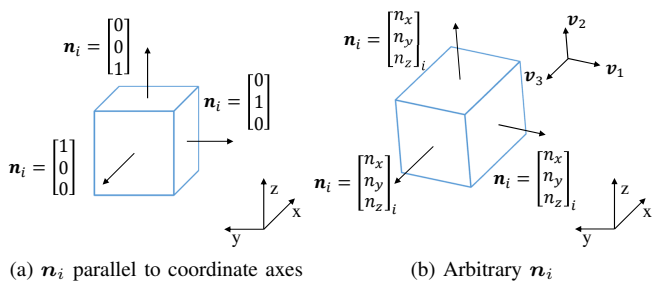


Fig. 5. Sensitivity to translations: (a) Eigenvalues of Ψ_t corresponding to diagonal elements, (b) eigenspace transformation of Ψ_t required using the eigenvectors \mathbf{v}_1 , \mathbf{v}_2 and \mathbf{v}_3 .

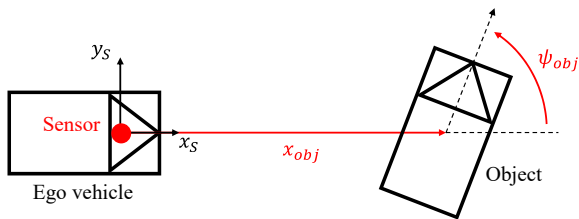


Fig. 6. Scheme of parameters for synthetic sensor data generation.

B. Static Pose Estimation Accuracies Using Real-World Data

To validate the theoretic accuracies from section V-A, we recorded real-world data from a vehicle (also BMW 740Li) and estimated its poses with our approach. Fig. 8 shows the corresponding empirical accuracies for various poses. Overall, the accuracies computed from real sensor data provide less optimistic estimations compared to the theoretic accuracies. The deviation mainly arise from the proneness of the registration algorithm to convert into local minima and the lever arm effect of ψ -noise on the mapping to the reference point (mid of the vehicle box). The results also demonstrate that within near range, the accuracy of point cloud registration outperforms DGPS data. However, real data experiments with higher distances than 20m have shown an even stronger deviation from theoretic accuracies due to the decrease of detected laser scanner points.

C. Tracking Accuracies Using Real-World Data

We show the performance of our tracking concept using real-world data with alternating highway take-over maneuvers. To account for distortions of the sensor point cloud \mathcal{S}

(due to ego and object vehicle motion) and the source of errors as presented in section V-B, we set $\sigma_S = 15\text{cm}$.

Fig. 9 shows our accuracy estimation (modeled noise) and the empirical registered poses for a sample cut-in maneuver. While the empirical x -noise stays at a low level, the empirical y - and ψ -noise increase when the object vehicle cuts into the lane of the ego vehicle. The corresponding modeled noise components demonstrate that such a decrease in accuracy, which is caused by the change of the observation perspective from side and rear to only rear, is well modeled by our approach. These observations confirm the perspective dependency of the accuracies as shown in the simulation results in Section V-A.

For our recorded data, Fig. 10 presents the overall state errors with respect to DGPS and the corresponding root mean square error (RMSE). Since ego vehicle heading errors from the DGPS system augment the Cartesian position errors by a lever arm, the position errors are specified in polar coordinates $r = \sqrt{x^2 + y^2}$ and $\alpha = \text{atan2}(y, x)$. The errors of the EKF are computed based on the estimations of the forward pass of the ERTS smoother. The application of ERTS slightly improves the position estimates x and y due to the already highly accurate measurements. Regarding $\delta\psi$, EKF and ERTS both provide worse estimates since our test drives were recorded on a highway test circuit with an almost constant lateral tilt to the left. However, this leads to a slip angle which is not considered in the coordinated turn model. In contrast, the performance gain of the ERTS compared to the EKF is clearly visible for the dynamic states v and ω since the influence of the smoothable process noise (w_a and w_ω in (1)) are higher on these states. Overall, Fig. 10 shows that the state estimation errors with respect to DGPS are tightly bounded.

VI. CONCLUSIONS

This paper addresses the issue of generating highly accurate ground truth data for object tracking in laser scanner data using CAD models of objects. To this end, we have exploited the benefits of an offline approach by applying several tracking iterations and by the consideration of future measurements with smoothing algorithms. We further demonstrated the benefits of a novel accuracy model providing the measurement noise for the tracking process using simulation and real-world data. In the presented scenarios,

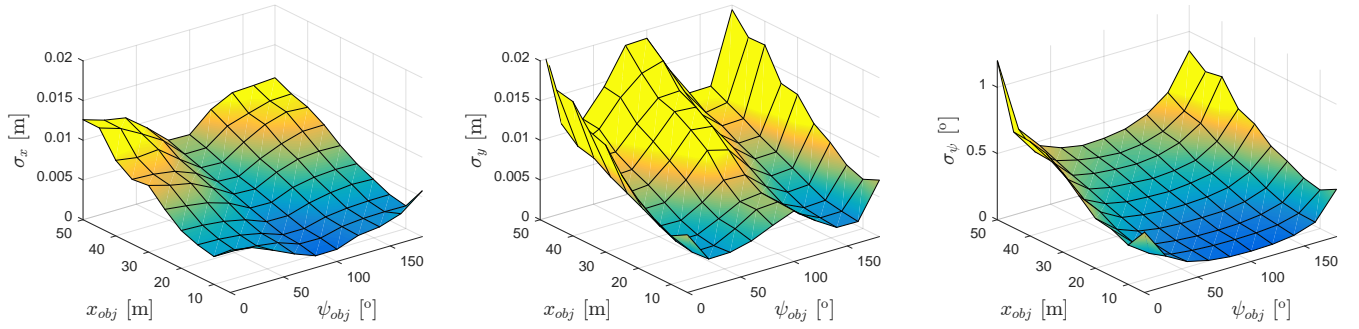


Fig. 7. Theoretic accuracies of static pose estimation: The accuracy values are computed using simulated sensor data sampled from the CAD model and the expressions in (12) and (13).

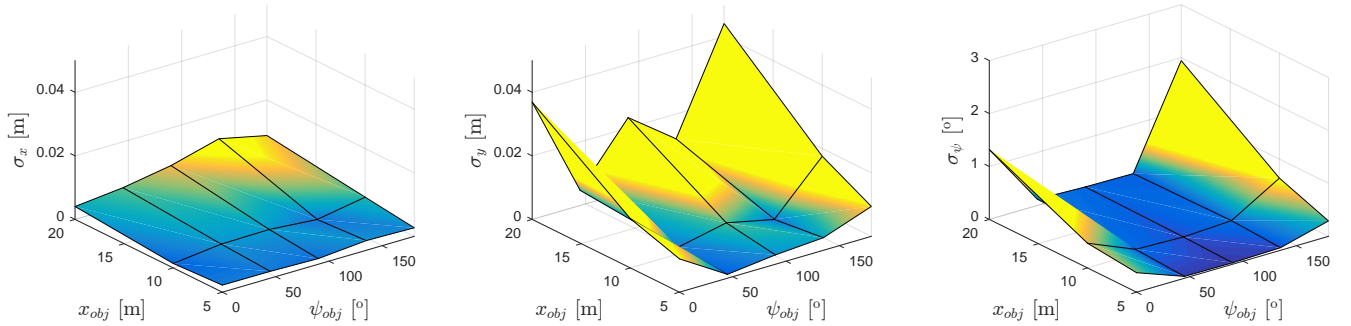


Fig. 8. Empirical accuracies of static pose estimation: The accuracy values are the standard deviations computed from the pose estimations of about 50 - 100 real sensor data frames per configuration. Thereby, the initial pose for point cloud registration is defined by DGPS data.

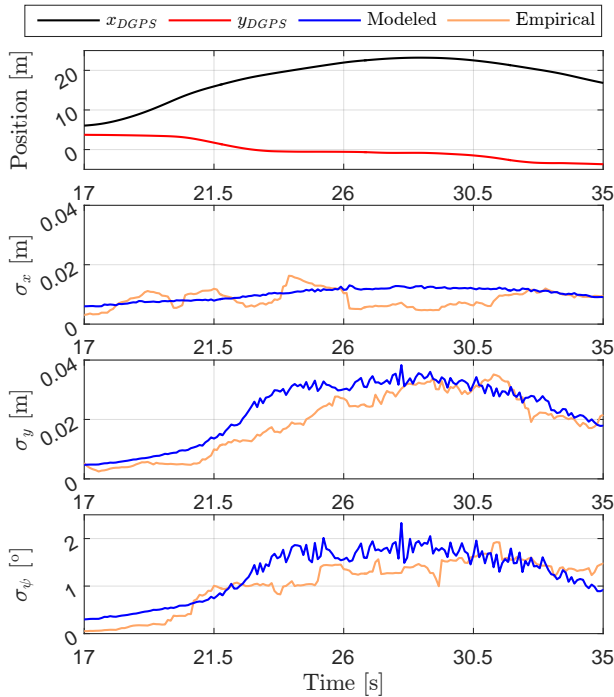


Fig. 9. Noise progress during cut-in maneuver: Modeled noise refers to the expressions in section IV, while the empirical noise is the standard deviation computed from the state errors between the registered pose and the recorded DGPS states x_{DGPS} , y_{DGPS} , and ψ_{DGPS} (error values are taken from a sliding window with length of 2s around every time step).

our tracking approach achieves highly precise state estimates, which enables developers of autonomous vehicles to generate ground truth data more easily than in state-of-the-art approaches. In the future, we want to increase the robustness in case of occlusions for the application in industrial development processes.

REFERENCES

- [1] J. E. Stellet, M. R. Zofka, J. Schumacher, T. Schamm, F. Niewels, and J. M. Zöllner, "Testing of advanced driver assistance towards automated driving: A survey and taxonomy on existing approaches and open questions," in *Proc. of the 18th IEEE International Conference on Intelligent Transportation Systems*, 2015, pp. 1455–1462.
- [2] Y. Takeda, Tsuchiya Chikao, and Khiat Abdelaziz, "Ground truth generation for quantitative performance evaluation of localization methods in urban areas," in *Proc. of the IEEE Intelligent Vehicles Symposium*, 2019, pp. 1024–1030.
- [3] J. E. Stellet, L. Walkling, and J. M. Zöllner, "Post processing of laser scanner measurements for testing advanced driver assistance systems," in *Proc. of the 19th International Conference on Information Fusion*, 2016, pp. 1999–2006.
- [4] H. E. Rauch, C. T. Striebel, and F. Tung, "Maximum likelihood estimates of linear dynamic systems," *AIAA Journal*, vol. 3, no. 8, pp. 1445–1450, 1965.
- [5] U. Lages, M. Spencer, and R. Katz, "Automatic scenario generation based on laserscanner reference data and advanced offline processing," in *Proc. of the IEEE Intelligent Vehicles Symposium*, 2013, pp. 153–155.
- [6] A. Dewan, T. Caselitz, G. D. Tipaldi, and W. Burgard, "Motion-based detection and tracking in 3D LiDAR scans," in *Proc. of the IEEE International Conference on Robotics and Automation*, 2016, pp. 4508–4513.

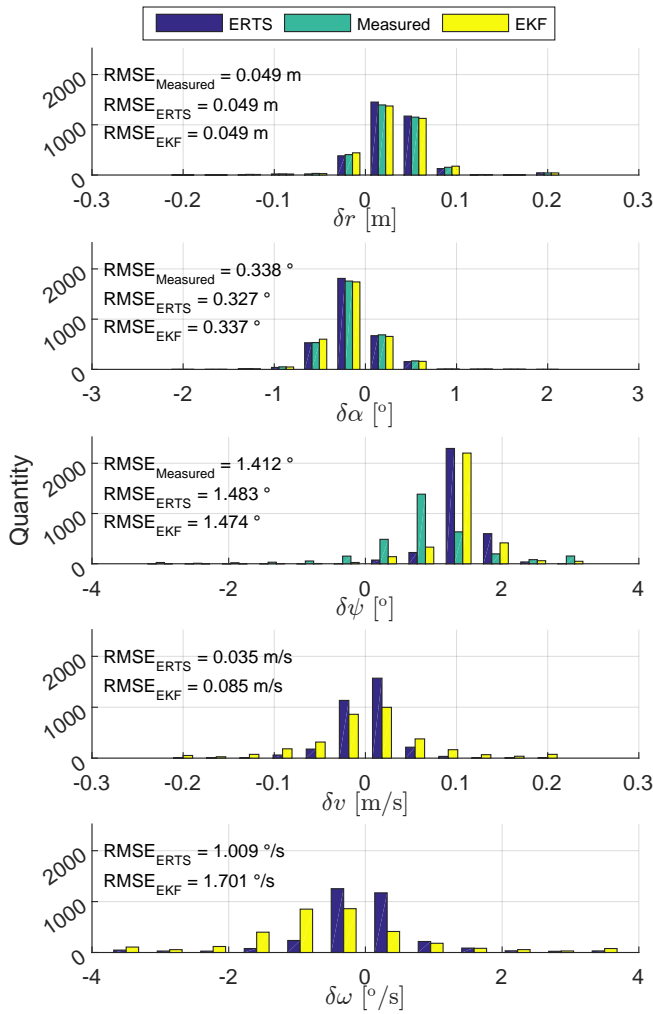


Fig. 10. Errors of measured, extended Kalman Filter (EKF), and extended Rauch-Tung-Striebel smoother (ERTS) estimates with respect to DGPS.

[7] F. Moosmann and T. Fraichard, "Motion estimation from range images in dynamic outdoor scenes," in *Proc. of the IEEE International Conference on Robotics and Automation*, 2010, pp. 142–147.

[8] 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.

[9] D. Z. Wang, I. Posner, and P. Newman, "Model-free detection and tracking of dynamic objects with 2D lidar," *The International Journal of Robotics Research*, vol. 34, no. 7, pp. 1039–1063, 2015.

[10] A. Petrovskaya and S. Thrun, "Model based vehicle detection and tracking for autonomous urban driving," *Autonomous Robots*, vol. 26, no. 2-3, pp. 123–139, 2009.

[11] N. Kämpchen, "Feature-level fusion of laser scanner and video data for advanced driver assistance systems," Ph.D. dissertation, Universität Ulm, 2007.

[12] L. Zhu, J. Barhak, V. Srivatsan, and R. Katz, "Efficient registration for precision inspection of free-form surfaces," *The International Journal of Advanced Manufacturing Technology*, vol. 32, no. 5-6, pp. 505–515, 2007.

[13] M.-Y. Liu, O. Tuzel, A. Veeraraghavan, Y. Taguchi, T. K. Marks, and R. Chellappa, "Fast object localization and pose estimation in heavy clutter for robotic bin picking," *The International Journal of Robotics Research*, vol. 31, no. 8, pp. 951–973, 2012.

[14] B. Kehoe, A. Matsukawa, S. Candido, J. Kuffner, and K. Goldberg, "Cloud-based robot grasping with the google object recognition engine," in *Proc. of the IEEE International Conference on Robotics and Automation*, 2013, pp. 4263–4270.

[15] 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.

[16] 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.

[17] 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.

[18] T. Möller and B. Trumbore, "Fast, minimum storage ray-triangle intersection," *Journal of Graphics Tools*, vol. 2, no. 1, pp. 21–28, 1997.

[19] K. Klasing, D. Wollherr, and M. Buss, "A clustering method for efficient segmentation of 3D laser data," in *Proc. of the IEEE International Conference on Robotics and Automation*, 2008, pp. 4043–4048.

[20] M. Himmelsbach, F. v. Hundelshausen, and H.-J. Wuensche, "Fast segmentation of 3D point clouds for ground vehicles," in *Proc. of the IEEE Intelligent Vehicles Symposium*, 2010, pp. 560–565.

[21] F. Moosmann, O. Pink, and C. Stiller, "Segmentation of 3D lidar data in non-flat urban environments using a local convexity criterion," in *Proc. of the IEEE Intelligent Vehicles Symposium*, 2009, pp. 215–220.

[22] Y. Bar-Shalom and T. E. Fortmann, *Tracking and data association*, ser. Mathematics in Science and Engineering. Boston: Academic Press, 1988, vol. 179.

[23] S. Pietzsch, "Model-based sensor data fusion of laserscanner and radar for the perception of complex vehicle environments," Ph.D. dissertation, Technische Universität München, 2014.

[24] M. Althoff, M. Koschi, and S. Manziinger, "Commonroad: Composable benchmarks for motion planning on roads," in *Proc. of the IEEE Intelligent Vehicles Symposium*, 2017, pp. 719–726.

[25] R. Schubert, E. Richter, and G. Wanielik, "Comparison and evaluation of advanced motion models for vehicle tracking," in *Proc. of the 11th International Conference on Information Fusion*, 2008, pp. 1–6.

[26] A. Beck, *Introduction to nonlinear optimization: Theory, algorithms, and applications with MATLAB*, ser. MOS-SIAM series on optimization. Philadelphia, Pa.: SIAM Society for Industrial and Applied Mathematics, 2014, vol. 19.

[27] A. W. Fitzgibbon, "Robust registration of 2D and 3D point sets," *Image and Vision Computing*, vol. 21, no. 13-14, pp. 1145–1153, 2003.

[28] S. J. Russell and P. Norvig, *Artificial intelligence: A modern approach*, 3rd ed., ser. Prentice Hall series in artificial intelligence. Boston: Pearson, 2010.

[29] K.-L. Low, "View planning for range acquisition of indoor environments," Ph.D. dissertation, University of North Carolina, 2006.

[30] D. A. Simon, "Fast and accurate shape-based registration," Ph.D. dissertation, Carnegie Mellon University, 1996.

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