

Adaptive Point Sampling for LiDAR-based Detection and Tracking of Fast-moving Vehicles using a Virtual Airport Environment

Hannes Braßel

Chair of Air Transport Technology
and Logistics
Technische Universität Dresden
Dresden, Germany
hannes.brassel@tu-dresden.de

Alexander Zouhar

Chair of Air Transport Technology
and Logistics
Technische Universität Dresden
Dresden, Germany
alexander.zouhar@tu-dresden.de

Hartmut Fricke

Chair of Air Transport Technology
and Logistics
Technische Universität Dresden
Dresden, Germany
hartmut.fricke@tu-dresden.de

Abstract— Safe and efficient airport operations are highly dependent on the availability of reliable visual scene information. Especially at airports without A-SMGCS support, the controller has to combine several input sources to interpret the overall situation on the ground. LiDAR sensor technology combined with computer vision algorithms for object detection and tracking was identified as a cost-effective method to support surveillance tasks of the controller, especially with regard to non-cooperative objects. Following previous research on semantic segmentation of airport scenes, this paper deals with a real-time detection and tracking method of fast-moving objects on the apron in labeled LiDAR scans. The method integrates a novel point sampling strategy into a basic Kalman filter by selecting a subset of representative points from the scanned points as a function of the velocity and distance of a moving object from the sensor. We show that the accuracy of the proposed detection and tracking method is in-line or close to ICAO A-SMGCS standards up to a distance of 350 m, covering a typical small apron.

Keywords- airport ground surveillance, apron control, safety, LiDAR, point cloud, 3D modeling, semantic segmentation, detection and tracking

I. INTRODUCTION

In this paper, we propose an adaptive point sampling strategy to detect and track non-cooperative, fast-moving vehicles with speeds up to 16 m/s in labeled LiDAR scans of the apron thereby emphasizing taxiing aircraft. To that end, we exploit the virtual airport environment introduced in [1] to handle the generation of labeled LiDAR scans of dynamic apron scenes. The task of tracking involves detecting objects of interest as well as their orientation or heading and their velocity. In this work, we follow the more general tracking-by-detection terminology and refer to detection and tracking as separate tasks that are solved jointly [2]. The motion parameters of the tracked objects provide a more precise picture of the traffic situation on the apron and are indispensable to predict and prevent risky object constellations either automatically or by means of a decision support system.

Detection and tracking of moving vehicles in LiDAR scans are particularly challenging due to the inherent sparsity of LiDAR and the sensor latency at lower scan rates giving rise to blurry, deteriorated point sets. The sparsity also increases with

increasing distance from the sensor due to the non-uniform resolution of LiDAR. Moreover, due to fast movements, it is also more likely that even a high-resolution LiDAR is not able to consistently cover the surface regions visible by the sensor in addition to self-occluded parts of the scanned surface.

Such sparse point sets may not be directly utilized to derive safety-critical decisions, for example, the compliance with distance rules between moving objects and obstacles. Instead, some form of prior information about the underlying object category is needed, e.g. derived from the category label such as the real object dimensions represented by a 3D bounding box [3]. This can be seen in the example of the virtual airport environment in Fig. 1 that depicts a labeled LiDAR point cloud of Dresden airport together with a blurry scan of a moving A320 collected over a time span of 10s.

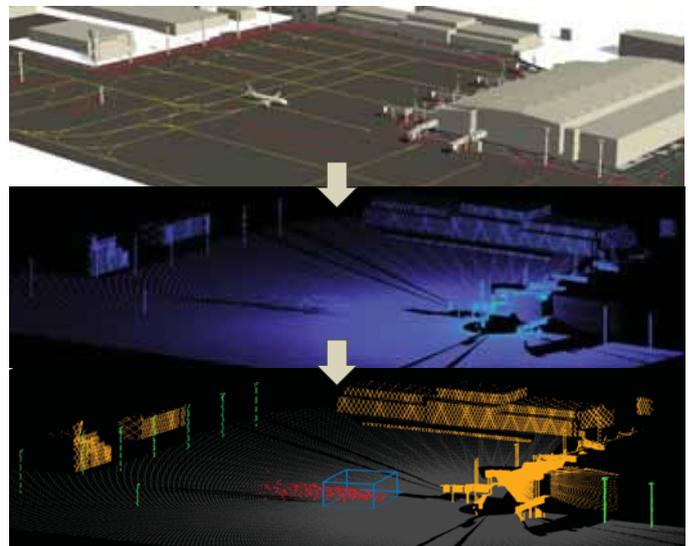


Figure 1. The proposed approach for detection and tracking of fast-moving vehicles (bottom) in simulated LiDAR scans (middle) of the apron using a virtual environment (top).

The detected aircraft is highlighted in the blurry point cloud by its 3D bounding box that indicates the current pose (location and heading). In fact, little work exists in the literature that



specifically deals with sparsity and motion blur for detection and tracking of objects in labeled LiDAR scans [4]–[6]. This is why in this work we specifically deal with the detection and tracking of taxiing aircraft where we exploit bounding box priors associated with the semantic category to estimate the pose and the velocity of aircraft in sparse, deteriorated LiDAR scans. On the other hand, some LiDAR sensors available on the market tend to be less affected by the impact of motion blur due to higher framerates and the way how points from previous measurement cycles are handled, however, often at the cost of lower scanning resolution and lower operating distance [1]. To overcome the challenge of motion blur, we first measure how sensitive the detection performance is towards varying velocities of an aircraft maneuvering along all permissible inbound tracks from the runway to the gate using Dresden airport as an example. Notice, that the insight gained from this analysis helps to understand how the motion and relative location of aircraft affect the scanning behavior of LiDAR and to derive a point sampling strategy for detection and tracking that takes the aircraft velocity and distance into account.

Next, we model a Kalman filter to smooth the detected aircraft positions in subsequent time frames. The filter integrates the proposed point sampling strategy that exploits the estimated dependency between the velocity and target distance to the sensor and the „optimal“ number of points sampled from the point set labeled as „aircraft“.

Moreover, in this prototypical implementation, we assume to have accurately labeled LiDAR scans of the apron at our disposal for which we exploit the virtual airport environment in [7] that integrates a single LiDAR sensor model also introduced in [7] resembling the scanning behavior of a first-generation Neptec OPAL-360HP sensor [8] installed at Dresden airport. In this way, the framework assigns each point a unique ground-truth label that specifies its semantics, e.g. A320, building, ground-plane, pole, etc. In contrast to [7], apron operations involving taxiing aircraft are simulated by imposing a motion model on the aircraft instances based on common airport regulations and aircraft kinematics. The quantitative analysis of our LiDAR-based detection and tracking model indicates that the accuracy of the generated results is in line or close to the limits set by the A-SMGCS licensing requirements [9], [10].

Finally, it is important to note that the proposed adaptive point sampling technique specifically exploits the scanning principle of the Neptec OPAL-360HP sensor. However, the method may readily be adapted to other LiDAR sensors available on the market.

A. Detection and Tracking

Object detection and tracking are essential tasks in automatic visual scene understanding. There exists a large body of literature dealing with 2D/3D image-based detection and tracking [11]–[13]. LiDAR sensing captures sparse 3D points from the surfaces in the scene and measures the depth information in contrast to e.g. stereo vision thereby resolving the scale problem. Much of the existing work on 3D detection and tracking in LiDAR-based point clouds is motivated by the autonomous driving domain. In this context, LiDAR data are often used to detect and track other vehicles relative to a reference car. The work in [14], for example, provides an extensive overview of the existing work in this field.

For the automotive domain, sensors with an overlapping scan pattern are commonly used giving rise to lower scan resolutions [1]. However, these sensors are not appropriate to identify fine differences, as is necessary at airports, for example, to distinguish between different types of aircraft or to detect small foreign objects on the airport surface.

The principle idea behind most traditional 3D detection and tracking techniques in raw point cloud scans is to locate and remove the ground plane and background points followed by clustering or region growing to identify and separate potential object proposals [15]. Once the object proposals are found an object-specific motion model may be applied in order to track the objects over time and to predict future trajectories to avoid collisions for example using Kalman filtering or particle filtering [16], [17]. More recent techniques for object detection and tracking make use of deep learning, however, often at the cost of computational efficiency [18].

Prior work on object detection in LiDAR scans of the apron focused on static or quasi-static scenes. The detection of small static objects (FOD), for example, was addressed in [19], [20]. The method in [21] jointly classifies and estimates the pose of aircraft parked at the gate. The work in [7] segments maneuvering aircraft on taxiways in quasi-static poses of aircraft traveling along predefined paths.

Little work exists that specifically deals with object detection and tracking in LiDAR scans of highly dynamic apron scenes. This is also due to the fact that state-of-the-art algorithms in detection and tracking often rely on large-scale data sets with point-wise annotations for learning of the underlying model parameters [22]. However, no such annotated large-scale data set currently exists and the generation of these data under real operating conditions is challenging. For this reason, we employ the virtual airport environment in [7] which potentially allows the simulation of arbitrary apron scenes.

Another challenge for 3D detection and tracking in LiDAR scans is the above-mentioned sparsity and latency of LiDAR giving rise to blurry deteriorated point clouds as a result of the relative motion between the sensor and the scanned objects of interest [23], [24]. However, ongoing developments in LiDAR sensing technology focus on increasing spatial resolutions at longer ranges and higher framerates [8]. This is particularly relevant in the context of apron surveillance where a real-time sensor coverage of the maneuvering area is essential.

B. The novelty in 3D LiDAR Tracking

This paper makes at least three contributions: (a) we extend the virtual airport environment in [7] by imposing motion models on the 3D aircraft models, e.g. an A320, (b) an adaptive point sampling strategy that directly operates in the point cloud domain and that takes the velocity of a moving vehicle (e.g., taxiing aircraft) and the sensor distance into account, and (c) baseline results for detection and tracking of aircraft in labeled LiDAR scans using Kalman filtering based on the sampling strategy in (b). Our work is also in line with the current trend of augmented automated airport operations, e.g. the implementation of higher-level A-SMGCS functions involving object interactions such as conflict detection and –alert [9], [25]. The paper is structured as follows. Starting with an overview of the method we continue to address each aspect of the framework in section II. Next, in section III we provide experimental results

for detection and tracking of taxiing aircraft on the apron in labeled LiDAR scans. Finally, section IV concludes with a summary of our findings and an outlook of our future work.

II. METHOD

This section describes each component of the proposed framework as illustrated by the orange arrows in Fig. 2. The input and output of each step of the simulation environment are indicated by the green and blue colors, respectively. See also [7] for more details about the data sources.

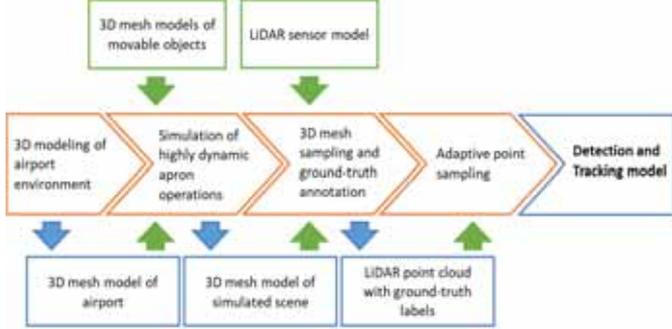


Figure 2. Overview of approach. See the text for details.

Section II.A addresses the modeling of fully dynamic apron scenes and the generation of labeled point scans using the virtual airport environment introduced in [7]. Our apron simulations constitute taxiing aircraft along all permissible tracks from the runway to the gate at Dresden airport covering a range of typical velocities on straight and curved sections of a taxiway (see Fig. 3).

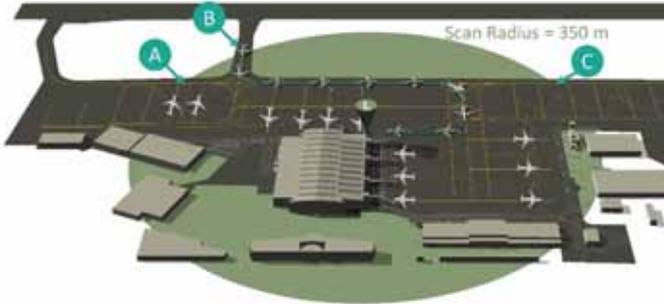


Figure 3. Close-Up View of the reconstructed CAD environment. “L”: Position Lidar. “A/B/C”: Origins of inbound tracks, Aircraft located at the gates represent final positions. Example tracks and motion patterns of aircraft moving along taxiways, e.g., A320 inbound.

Section II.B presents the details of the proposed point sampling strategy for 3D detection and tracking in labeled LiDAR scans generated in section II.A.

A. 3D Modeling of the Airport Environment and Simulation of dynamic Apron Operations

In this section, we exploit the virtual airport environment introduced in [7] to handle the generation of labeled LiDAR scans of dynamic apron scenes. The virtual environment constitutes a true-to-scale 3D CAD model of Dresden airport including static components of the airport infrastructure (e.g., buildings, poles, ground-plane) as well as a large variety of movable objects (e.g., different types of aircraft and ground vehicles). The 3D CAD model is shown in Fig. 4 together with an aligned aerial view of the real airport environment. All static and movable objects composing the virtual scene are exported

individually in stereolithography (STL) format together with a unique reference that identifies the object category (semantic label). Without loss of generality, in this work, we restrict the set of movable objects to the most important category “aircraft” for which an A320-200 3D mesh model is also depicted in Fig. 4. As an example, the close-up view of the virtual CAD environment in Fig. 3 shows a taxiing aircraft A320 in subsequent positions to demonstrate typical motion patterns of arriving aircraft considered in this work. Eventually, all 3D models composing the scene are loaded into a simulation environment based on MATLAB which implements the scene dynamics. The motion of taxiing aircraft and other ground-vehicles is defined by rigid transformations (rotation, translation) on the apron ground-plane under the constraints of the operational rules of the airport and the kinematic properties of the aircraft. For this purpose, typical tracks of aircraft on the apron, i.e. taxiways, stop-lines, other road markings, and the permissible headings were integrated as additional constraints into the simulation environment. Fig. 6 depicts a bird’s eye view of the virtual airport environment with all permissible inbound tracks for A320 highlighted in green. Moreover, the permissible velocities of taxiing aircraft are based on the recommended speed according to the local airport regulations and specifications in accordance with [9]. For example, on straight sections of taxiways, aircraft move at speeds of up to approximately ~16 m/s. On curves and complex taxiway configurations, the speed does not exceed a value of ~6 m/s.

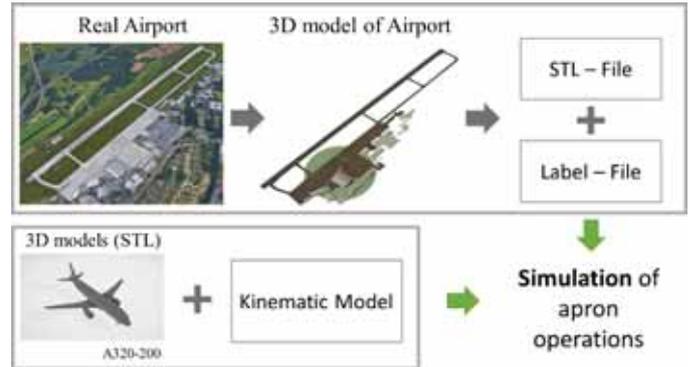


Figure 4. (Top-Left) Aerial view of DRS, (Top-Middle) 3D mesh model of Dresden Airport reconstructed in AutoCAD. 3D mesh models (STL) and labels of static objects (Top-Right) and movable objects¹ (e.g., aircraft A320-200) are passed as input to the simulation environment.

The motion of a taxiing aircraft along a predefined track may be described by the general equation of motion, i.e.,

$$\begin{bmatrix} x_{n+1} \\ y_{n+1} \\ v_{n+1} \\ a_{n+1} \\ \theta_{n+1} \end{bmatrix} = \begin{bmatrix} 1 & 0 & \cos(\theta_n)\Delta t & 0.5 \cos(\theta_n)\Delta t^2 \\ 0 & 1 & \sin(\theta_n)\Delta t & 0.5 \sin(\theta_n)\Delta t^2 \\ 0 & 0 & 1 & \Delta t \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_n \\ y_n \\ v_n \\ a_n \\ \theta_n \end{bmatrix} + \Delta t \begin{bmatrix} 0 \\ 0 \\ 0 \\ \dot{a}_n \\ \dot{\theta}_n \end{bmatrix} \quad (1)$$

x, y - Projected center (c) of aircraft on the ground

v - Aircraft velocity

a - Aircraft acceleration

θ - Aircraft orientation

n - Discrete timestep

under the constraints (2) and (3) where the quantities $v_{\text{segment}_{\max}}$, a_{\min} , a_{\max} denote the maximum permissible velocity in a particular segment (straight/curved) on the taxiway and the acceleration/deceleration limits of the aircraft.

$$\{v_n | 0 \leq v_n \leq v_{\text{Segment}_{\max}}\}, v_n, v_{\text{Segment}_{\max}} [m/s] \quad (2)$$

$$\{a_n | a_{\min} \leq a_n \leq a_{\max}\}, a_n, a_{\min}, a_{\max} [m/s^2] \quad (3)$$

Notice, that the simulated motion of an aircraft according to (1) is controlled by the temporal derivation of a and θ , i.e. \dot{a} , $\dot{\theta}$ where the orientation θ is defined by the configuration of the considered track. There are many possible ways to adjust the acceleration/deceleration value of a taxiing aircraft. In our approach, we sample the acceleration/deceleration a_n at time step n from a normal distribution with a mean of $2 m/s^2$ and $-0.5 m/s^2$ with standard deviation of $1 m/s^2$ and truncated at a_{\min} ($0 m/s^2/-1 m/s^2$), a_{\max} ($0 m/s^2/3 m/s^2$), respectively such that the velocities satisfy the speed constraint in (2) in each track segment. According to [9] an acceleration of 1 to $2 m/s^2$ should be considered for the taxiing (routing). In our simulation, the acceleration may also reach higher values in order to cover the entire range of velocities specified in the requirements in [9]. The virtual airport environment integrates a LiDAR sensor model to generate synthetic LiDAR scans of the apron and the movement area according to the hardware specifications of the sensor [8] installed at Dresden airport. The large green circle in Fig. 2 illustrates the considered range of the LiDAR sensor located on the north-east rooftop of the terminal building at Dresden airport as indicated by the green-colored cone. We also assume an idealized environment that abstracts from challenging atmospheric conditions (e.g., rain, fog, etc.) as well as from surface attributes beyond shape information, e.g., material properties and texture. On the other hand, in contrast to [7], the LiDAR scans obtained in this work are subject to motion dependent blur [23], [24] as a result of the relative motion between the scanned object and the sensor.

B. 3D Mesh Sampling and Ground-Truth Generation

This section summarizes the details of the LiDAR sensor² model highlighted in Fig. 2 and the sampling procedure to generate labeled LiDAR scans of dynamic apron scenes [26]. The sensor operates with a laser wavelength of 1540 nm and provides a field of view of 360° horizontally and 45° vertically with an azimuthal resolution of 0.0057° . The level of detail captured by the sensor greatly depends on the Pulse Repetition Rate (PRR) and the scanning duration. Assuming a scanning duration of 1 s and provided that the sensor operates with a PRR of 200 kHz the LiDAR sensor forms a point cloud of the scene containing up to 200.000 points. Each point is defined by its spatial x, y and z coordinates and its intensity. Fig. 5 illustrates how the scan pattern evolves. The red dot represents the sensor position. To take the sensor characteristics into account, we developed a LiDAR sensor model based on the hardware specification provided by Neptec. Please refer to [8] for more details. Scanning of simulated apron scenes involves computing triangle ray intersections for each light beam emitted by the sensor with a maximum scan rate of 200 kHz.

To this end, we employed a modified version of the Möller Trumbore ray triangle intersection algorithm [27]. For computational efficiency, the calculations were carried out on a GPU (GeForce RTX 2080 Ti).

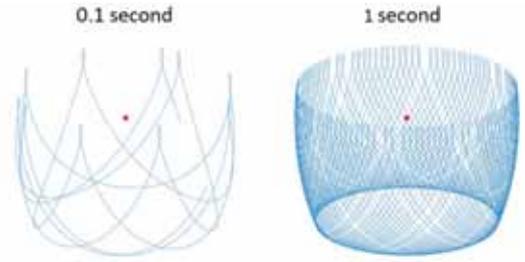


Figure 5. Evolution of the LiDAR scan pattern over time. The red dot represents the sensor position.

For each laser light beam intersecting a triangle of a 3D surface mesh the reflectance value is calculated and the semantic label of the corresponding triangle is assigned. If a ray intersects several triangles, only the first intersection is considered. TABLE I summarizes the semantic labels currently used in our simulations together with a unique color code that identifies the semantic category. The simulation of taxiing aircraft involves modeling of the motion parameters of an aircraft along the chosen track using (1). Based on the motion parameters it is then possible to compute the position and heading of an aircraft along the track assuming that the former is defined by the center of the bounding box of the aircraft. In our simulation, we choose a time discretization step of 0.005 s given that one azimuthal 360° rotation of the LiDAR sensor takes about 0.028 s. At each time step, the aircraft is positioned and oriented depending on the current taxiway configuration. From this, it follows that during a single azimuthal rotation of the sensor the aircraft adapts its pose approximately $0.028s/0.005s \sim 6$ times along the track. Fig. 1 depicts the labeled LiDAR scan of an arriving aircraft A320 after a time interval of 10 s. Notice, how the relative movement between the scanned aircraft and the sensor gives rise to motion blur.

The total number of inbound tracks at Dresden airport considered in our experiments is 42 (see also Fig. 6). Using different combinations of maximum velocities $v_{\text{Segment}_{\max}}$ and acceleration/deceleration values along the taxiways we obtain a total number of 500 labeled LiDAR scans of taxiing aircraft. Besides, each simulated track comprises a sequence of aircraft poses (positions, headings) along the track, velocities, and distances between the aircraft and the sensor which we use as ground-truth data to derive the adaptive point sampling strategy for detection and tracking. Notice, that the number of ground-truth positions per track may differ depending on the length of the track and the boundary conditions in (2) and (3).

TABLE I. SEMANTIC LABELS AND ASSOCIATED COLORS

	Semantic label			
	A320	Pole	Building	Ground-plane
Color code				

¹Moveable 3D models were kindly provided by TRANSOFT Solutions.

²The LiDAR sensor at our disposal is a first-generation OPAL-360 HP sensor developed by the company Neptec. The motion of a taxiing

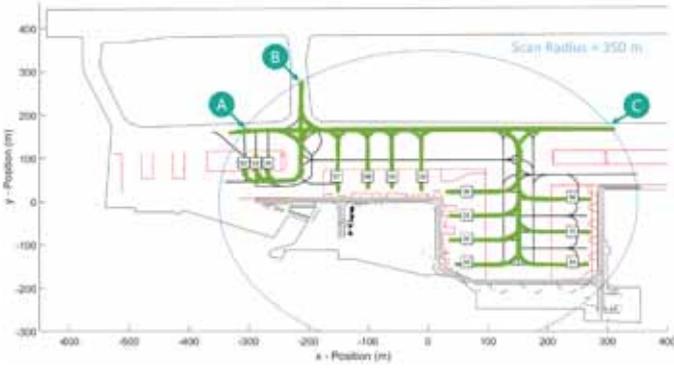


Figure 6. Bird's eye view of all inbound tracks for aircraft A320 at Dresden airport starting at position A/BC.

C. Adaptive Point Sampling for Detection and Tracking

Let P denote a set of labeled points of a taxiing aircraft captured by LiDAR during the scanning interval $\Delta t > 0$ starting at some point of time t as depicted in Fig. 1, where the red color encodes the class A320-Inbound (see Table I). Our objective is to estimate the pose of the aircraft represented by a bounding box (see Fig. 1) and the velocity at $t + \Delta t$ using P where P is affected by motion blur. Notice, that the dimensions of the bounding box are readily available from the aircraft type encoded by the semantic label of P . A naive approach would be to sample a constant cardinality subset $P' \subseteq P$ of P with $|P'| = \text{const}$ and to estimate the pose using P' irrespective of where the aircraft is located and oriented relative to the sensor and how fast it moves. A more reasonable strategy may be derived from the scanning principle of the OPAL-360HP sensor by noting that the points in P form an ordered sequence according to the acquisition timestamp t_k associated with each point $p_k \in P$ such that

$$P = (p_k, t_k)_{k=1, \dots, |P|} \quad (4)$$

forms a sequence of pairs (p_k, t_k) where $\forall k: t_k < t_{k+1}$, $1 \leq k < |P|$ defines the ordering of P . Notice, that in general, we have $t \leq t_1$ and $t_{|P|} \leq t + \Delta t$. A simple strategy to cope with the variable impact of motion blur in P is then to exploit the sequential ordering by sampling the last N^* points of P that best represent the true pose of the aircraft at $t + \Delta t$ given its velocity and its distance from the sensor. More formally, among all connected subsequences P' of the form

$$P' \in \{(p_k, t_k)_{k=k', \dots, |P|} \mid 1 \leq k' \leq |P|\} \quad (5)$$

we aim to find a connected subsequence P^* with

$$P^* = \arg \min_{P'} \text{Err}(\text{Pose}_{\text{GT}}, \text{Pose}_{P'}) \quad (6)$$

such that $N^* = |P^*|$ denotes an optimal number of representative points of P and $\text{Err} \geq 0$ computes the deviation between the ground-truth pose, say Pose_{GT} of the aircraft at $t + \Delta t$ and the estimated pose, say $\text{Pose}_{P'}$ given P' at $t + \Delta t$. We define the function Err to compute only the position error between the bounding box center $c_{P'} \in \mathbb{R}^3$ of P' and the ground-truth center $c_{\text{GT}} \in \mathbb{R}^3$ of the aircraft at $t + \Delta t$ using the L1-Norm which is in line with the A-SMGCS requirements covering the representation of detected objects [9], i.e.,

$$\text{Err}(\text{Pose}_{\text{GT}}, \text{Pose}_{P'}) = \|c_{P'} - c_{\text{GT}}\| \quad (7)$$

where we omit the argument $t + \Delta t$ for better readability. Fig. 7 plots the resulting number N^* of representative points using (5)-(7) as a function of the velocity and the distance of the aircraft from the sensor across all scans using the ground-truth data obtained in II.B where we set the sampling interval Δt across all scans to $\Delta t = 0.1$ s.

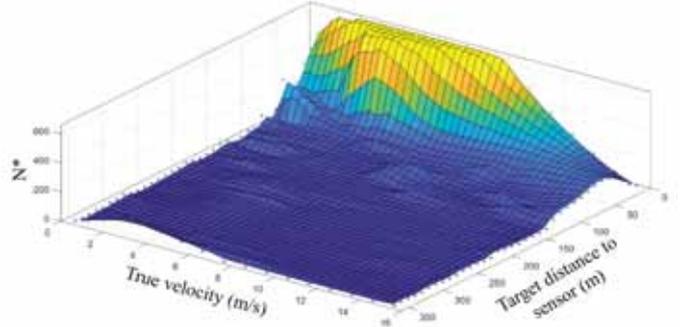


Figure 7. Number N^* of representative points as a function of the velocity and the distance between aircraft and sensor.

The resulting characteristic function in Fig. 7 interpolates linearly between the discrete measurements over the considered domain to approximate a continuous function. From the plot, one can see that the number N^* of representative points increases more rapidly with decreasing velocity and increasing distance from the sensor. On the other hand, higher velocities give rise to smaller values N^* which is expected since the impact of motion blur increases with increasing velocity so that points p_k in P with smaller time stamps t_k tend to distort the position estimates. Also notice, that N^* was limited to a maximum value of $N^* = 600$ points.

Finally, the characteristic function depicted in Fig. 7 is integrated into a Kalman filter [28] to smooth the detected aircraft positions along arbitrary tracks. The resulting procedure is summarized in Algorithm 1.

Algorithm 1: Kalman filter with Adaptive Point Sampling

Input: Segmented point cloud

Output: Taxiing sequence: (Projected center of aircraft on the ground c_n , Aircraft orientation θ_n , Aircraft velocity v_n) $_{n=0, \dots, n_{\max}}$

- 0: *Initialisation:* $\Delta t = 0.1$ s, $n = 0$, Initial estimate of c_0, v_0, θ_0 , distance to sensor $dist_0$ for t_0 , Kalman prediction of c_1
- 1: $n \rightarrow n + 1$ //Increment time step
- 2: Get point set P_n during next time frame Δt
Get last N^* points of P_n using $v_{n-1}, dist_{n-1}$
//Adaptive point sampling (Fig. 7); Assume constant velocity during Δt
- 3: Detect position c_n at t_n //Object detection: BBox center of P^*
- 4: Kalman correction of c_n at t_n using Kalman prediction of c_n at t_{n-1}
- 5: Kalman prediction of the position c_{n+1}
- 6: Estimate $v_n, dist_n, \theta_n$ using corrected position c_n and c_{n-1} , respectively
- 7: Go to 1.

III. RESULTS

In this section, we validate the performance of the proposed detection and tracking Algorithm 1 by comparing the method with a Kalman filter that replaces the adaptive point sampling method in Step 2 with a naive sampling method that draws a fixed number of points from the scanned point set P captured by the sensor during the scanning interval Δt . Moreover, we compare the proposed method with the licensing requirements for surveillance systems according to the A-SMGCS documentation [9].

The data set at our disposal consists of approximately 450.000 simulated ground-truth measurements of aircraft positions and orientations generated with the virtual airport environment described in II.A and II.B together with the associated velocities across all inbound tracks for A320 at Dresden airport. The simulation was carried out for all common velocities on straight and curved sections of the taxiways according to [9].

Fig. 8 and Fig. 9 depict two representative examples to illustrate our experimental setup and the capability of the proposed detection and tracking method in Algorithm 1. Specifically, Fig. 8 visualizes the estimated bounding box (blue) representing the pose estimate of a taxiing aircraft A320 (red) during some time interval Δt along the taxiway together with the ground-truth bounding box (green). The heading of the aircraft is encoded by the line segment that intersects the right face of the bounding box. The length of the line-segment is proportional to the estimated velocity shown at the top where the ground-truth speed in this example is 7 m/s . Notice, that such a visualization of the pose and the velocity may also assist the controller to quickly recognize the situation on the ground.

Fig. 9 shows the top view of a single track of a taxiing aircraft A320 moving from the runway at location A to the parking position at gate G30 (see Fig. 6). The green line indicates the ground-truth track of the taxiway and the blue points show the tracked sequence of aircraft positions. The labeled point cloud of the aircraft captured by the sensor is shown in red. Notice, that the algorithm starts to track the aircraft with some delay which is due to the fact that the tracker requires a minimum number of points to obtain a reasonable initialization (see Step 0 in Algorithm 1).

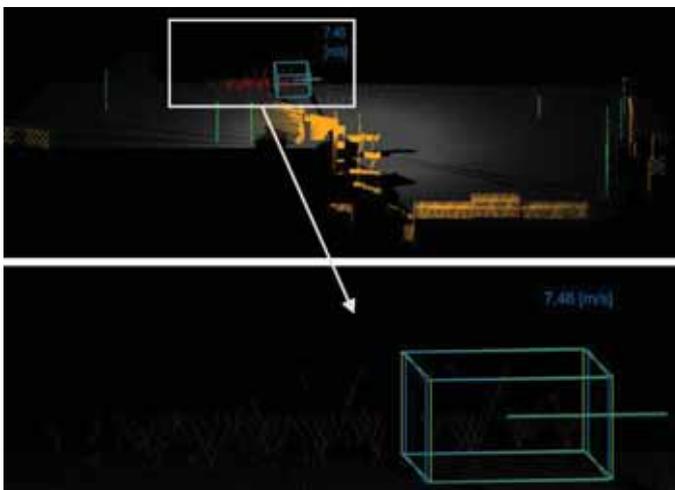


Figure 8. Snapshot of a tracked aircraft A320 using Algorithm 1 (blue) compared with the ground-truth (green). See the text for details.

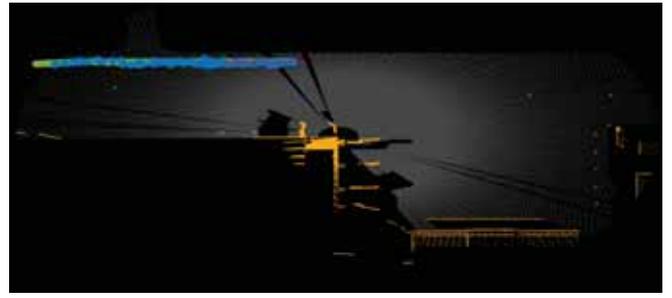


Figure 9. Bird's eye view of the tracked positions (blue) of a taxiing aircraft A320 (red points) along the ground-truth path (green) using Algorithm 1. See the text for details.

We conduct an extensive quantitative analysis of Algorithm 1 by validating the localization error of the detected aircraft positions across all tracks in the data set, the orientation (heading) deviation of the estimated bounding boxes, and the error of the estimated velocities. As a baseline, we use a Kalman filter with a simplified sampling strategy in Step 2 that selects a constant number of points from a point set P captured during a time interval Δt . We set the number of sampled points to the mean value $\bar{N}^* = 62$ computed over the entire data set which we found to be the best naive sampling strategy during our experiments. Notice, that Algorithm 1 also satisfies the run-time requirements outlined in [9] due to the update-rate of $\Delta t = 0.1 \text{ s}$ of the tracked positions.

In Fig. 10 we plot the detected aircraft positions using Algorithm 1 over all inbound tracks in the data set together with the localization error in (7) ranging from 0 m (blue) to $> 30 \text{ m}$ (yellow). From the plot one can see that the detected aircraft positions are overall consistently arranged along the tracks. Larger deviations from the ground-truth are mainly caused by cross-track errors which primarily occur as a result of self-occlusions of the scanned surfaces so that the detected positions tend to be biased towards surface regions visible by the sensor. TABLE II. also indicates that the adaptive point sampling strategy reduces the localization error of the detected positions compared to the naïve sampling method. Moreover, our results are mostly in-line or close to the A-SMGCS requirements referenced in the last row in TABLE II. Further analysis shows that after removing 1% of the outliers we achieve an average localization error of $\sim 7.5 \text{ m}$ as required.

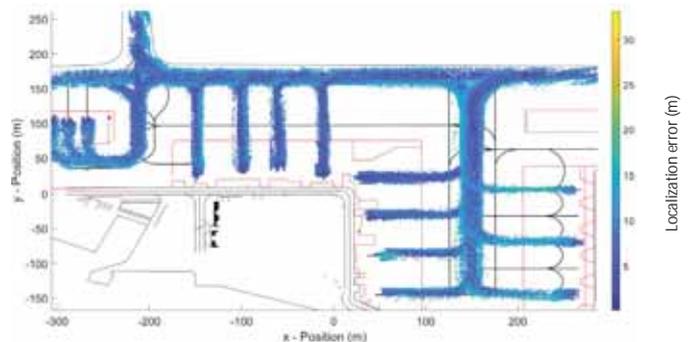


Figure 10. Bird's eye view of tracked measured aircraft positions across the entire data set using Algorithm 1. The color code indicates the localization error (7) ranging from 0 m (blue) to $> 30 \text{ m}$ (yellow). The Sensor is located at (0/0/0). Notice, that the majority of the measured positions are in line with the A-SMGCS requirements.

In Fig. 11 we plot the localization error as a function of the ground-truth velocity and the distance between aircraft and sensor to better interpret challenging constellations indicated by the yellow regions. Based on Fig. 11 we found that the largest localization errors occur when a taxiing aircraft reaches the parking positions 32, 33 (bottom, left yellow regions in Fig. 11) and the parking positions 37, 38 (top, left yellow regions in Fig. 11) located further away from the sensor. In fact, in the former case, the aircraft is partially occluded by a passenger bridge and in the second case, larger regions of the aircraft were not visible by the sensor due to self-occlusions. Moreover, the localization error tends to grow with increasing velocities and distances from the sensor which is also caused by the non-uniform resolution of LiDAR and initialization errors.

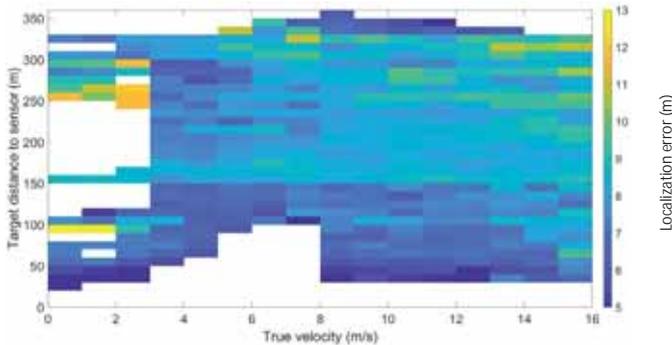


Figure 11. Localization error as a function of the velocity and the target distance between the sensor and scanned aircraft using (7). Notice, that the localization error is averaged over velocity intervals of 1 m/s and distance intervals of 10 m. For the white sections, no measurements were available.

TABLE II. LOCALIZATION-ERROR

Localization Error	Mean	Standard Deviation
Adaptive sampling	7.91m	3.79m
Naïve sampling	8.95m	4.26m
A-SMGCS requirements	„4.2.3 The actual position of an aircraft, vehicle or obstacle on the surface should be determined within a radius of 7.5 m.“ [9]	

Next, in Fig. 12 we plot the velocity estimation error as a function of the true velocity and the ground-truth distance from the sensor. The velocity estimation error was defined by the absolute value of the difference between the ground-truth velocity and the estimated velocity. The figure shows that for very low ground-truth velocities the error tends to be higher (yellow) which corresponds to situations where the aircraft are closer to the final parking positions and the localization accuracy is lower, partly due to self-occlusion (Fig. 11)

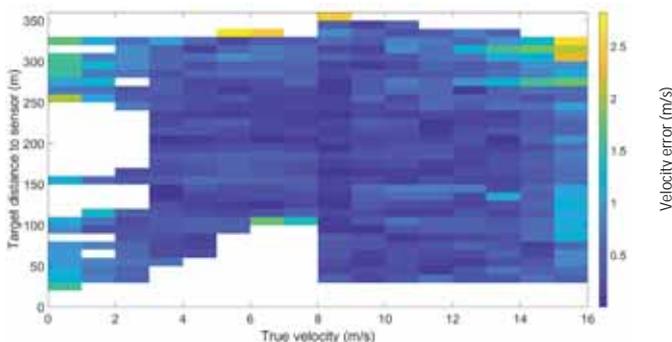


Figure 12. Velocity estimation error as a function of the true velocity and the target distance between the sensor and scanned aircraft. Notice, that the error

values are averaged over velocity intervals of 1 m/s and distance intervals of 10 m. For the white sections, no measurements were available.

Moreover, in some cases, the measurements also tend to under-estimate the velocities especially during the initialization phase of the tracker where the ground-truth velocities are actually higher as indicated by the yellow regions in the top right part in Fig. 12. Due to the constant velocity assumption, the Kalman filter may propagate larger velocities to subsequent time frames where the true velocity may be lower and vice versa. This can lead to erroneous Kalman predictions of subsequent aircraft positions. On the other hand, the statistics in Table III confirm that our approach satisfies the A-SMGCS requirements for velocity estimates

TABLE III. VELOCITY-ERROR

Velocity Error	Mean	Standard Deviation
Adaptive sampling	-0.33 m/s	1.86 m/s
Naïve sampling	-0.34 m/s	3.17 m/s
A-SMGCS requirements	„The A-SMGCS should be able to accommodate the following speeds determined to within ± 2 km/h (1 kt)“ [9]	

Finally, Fig. 13 plots the orientation estimation error as a function of the ground-truth velocity and the distance of the scanned object from the sensor. The orientation estimation error was defined by the absolute value of the difference between the ground-truth orientation and the estimated orientation. Compared to the A-SMGCS requirements in TABLE IV, the orientation estimation error appears to be relatively large. As indicated by the yellow and light blue regions in Fig. 13 larger errors are typically associated with larger distances from the sensor. To a great extent, these cases correspond to inconsistent location estimates during the initialization phase of the Kalman tracker. Moreover, for ground-truth velocities close to zero the orientation estimates also tend to exhibit larger deviations from the ground-truth which coincides with the localization errors near the parking positions shown in Fig. 10 and Fig. 11.

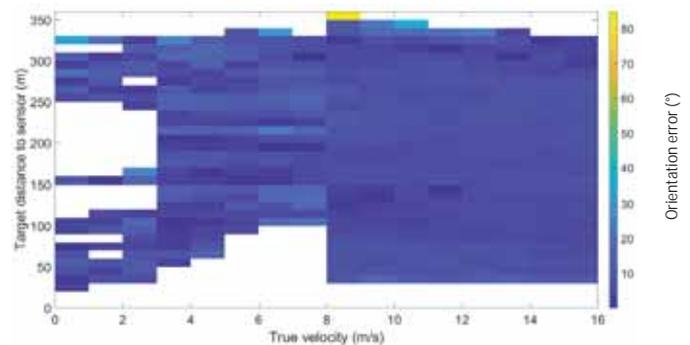


Figure 13. Orientation estimation error as a function of the true velocity and the target distance between the sensor and scanned aircraft. Notice, that the error values are averaged over velocity intervals of 1 m/s and distance intervals of 10 m. For the white sections, no measurements were available.

TABLE IV. ORIENTATION-ERROR

Orientation Error	Mean	Standard Deviation
Adaptive sampling	4.86°	34.83°
Naïve sampling	4.76°	40.78°
A-SMGCS requirements	„The A-SMGCS should determine the direction of movement in terms of the magnetic heading of each participating aircraft and vehicle to within $\pm 1^\circ$.“ [9]	

IV. CONCLUSIONS AND OUTLOOK

We have presented a fast detection and tracking method for non-cooperative objects in semantic LiDAR scans using a virtual airport environment. Our method integrates a novel sampling strategy into a basic Kalman filter that exploits the scanning principle of the LiDAR sensor to select points from sparse, deteriorated point sets as a function of the velocity of the scanned object and its distance from the sensor. Experimental results on simulated data demonstrate that our proposed strategy performs within or close to the limits set by the A-SMGCS requirements even by using just one sensor. Limitations of the approach mainly result from self-occlusions of the scanned surfaces which we plan to address in our future work, e.g., by integrating multiple LiDAR sensors into the airport environment. Further research directions include advanced tracking models with a constant steering angle assumption [29], multi-object detection and tracking, and long-term motion prediction to assess the risk of potential object positions ahead of time, e.g. to avoid collisions. A preliminary result is shown in Fig. 14. Furthermore, the suitability for different traffic levels and visibility ranges according to *Table C-1. Equipment evolution for A-SMGCS* [8] will be investigated.

Also notice, that in our scenarios the variability of the orientation of the aircraft along the inbound tracks was rather limited. In order to study the impact of the orientation on the optimal number N^* of sampled points, we will enrich the data set with additional poses of the aircraft to cover a larger range of orientations.

Finally, we believe that our proposed LiDAR-based detection and tracking framework can provide additional situational awareness for the controller and is well suited to be integrated into established surveillance systems such as surface movement radar (SMR) to improve the combined performance. To that end, the proposed framework will also be extensively validated on real LiDAR scans of the apron under different visibility conditions.

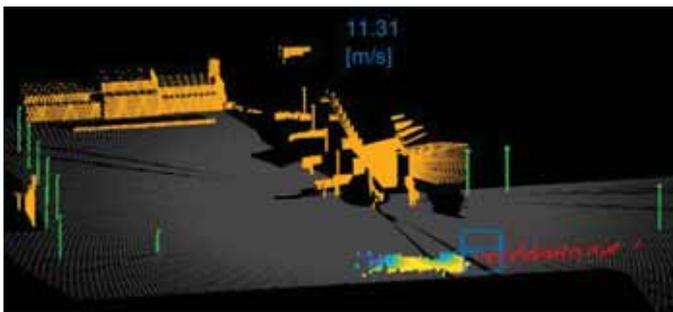


Figure 14. Quantitative motion prediction. The color indicates the probability of the future aircraft position from high (yellow) to low (blue).

REFERENCES

- [1] 'LiDAR Comparison Chart', *AutonomouStuff*. <https://autonomoustuff.com/lidar-chart/> (accessed Sep. 24, 2020).
 - [2] M. D. Breitenstein, F. Reichlin, B. Leibe, E. Koller-Meier, and L. Van Gool, 'Robust tracking-by-detection using a detector confidence particle filter', in *2009 IEEE 12th International Conference on Computer Vision*, Kyoto, Sep. 2009, pp. 1515–1522
 - [3] M. Berger *et al.*, 'State of the Art in Surface Reconstruction from Point Clouds', *Eurographics 2014 - State Art Rep.*, p. 25 pages, 2014
 - [4] Z. Koppányi and C. K. Toth, 'Object Tracking with LiDAR: Monitoring Taxiing and Landing Aircraft', *Appl. Sci.*, vol. 8, no. 2, Art. no. 2, 2018
 - [5] M. Sualeh and G.-W. Kim, 'Dynamic Multi-LiDAR Based Multiple Object Detection and Tracking', *Sensors*, vol. 19, no. 6, Art. no. 6, Jan. 2019
 - [6] M. Sualeh and G.-W. Kim, 'Visual-LiDAR Based 3D Object Detection and Tracking for Embedded Systems', *IEEE Access*, vol. 8, pp. 156285–156298, 2020
 - [7] H. Braßel, A. Zouhar, and H. Fricke, '3D Modeling of the airport environment for fast and accurate LiDAR semantic segmentation of apron operations', presented at the 39th Digital Avionics Systems Conference, San Antonio, Texas, USA, Oct. 2020.
 - [8] Neptec, 'Neptec Technologies Corp: OPAL-360-series Performance-Sheet', neptectechnologies.com
 - [9] ICAO, Doc 9830, AN/452. 2004
 - [10] 'EUROCAE ED-87 Minimum Aviation System Performance Specification (MASPS) for A-SMGCS'. 2019.
 - [11] L. Liu *et al.*, 'Deep Learning for Generic Object Detection: A Survey', *ArXiv180902165 Cs*, Aug. 2019
 - [12] X. Cheng, P. Wang, and R. Yang, 'Learning Depth with Convolutional Spatial Propagation Network', *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 10, pp. 2361–2379, Oct. 2020
 - [13] W. Luo *et al.*, 'Multiple Object Tracking: A Literature Review', *ArXiv14097618 Cs*, May 2017
 - [14] E. Yurtsever, J. Lambert, A. Carballo, and K. Takeda, 'A Survey of Autonomous Driving: Common Practices and Emerging Technologies', *IEEE Access*, vol. 8, pp. 58443–58469, 2020
 - [15] W. Wang, K. Sakurada, and N. Kawaguchi, 'Incremental and Enhanced Scanline-Based Segmentation Method for Surface Reconstruction of Sparse LiDAR Data', *Remote Sens.*, vol. 8, no. 11, Art. no. 11, 2016
 - [16] A. Ess, K. Schindler, B. Leibe, and L. Van Gool, 'Object Detection and Tracking for Autonomous Navigation in Dynamic Environments', *Int. J. Robot. Res.*, vol. 29, no. 14, pp. 1707–1725, Dec. 2010
 - [17] M. He, E. Takeuchi, Y. Ninomiya, and S. Kato, 'Precise and efficient model-based vehicle tracking method using Rao-Blackwellized and scaling series particle filters', in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct. 2016
 - [18] S. Chowdhuri, T. Pankaj, and K. Zipser, 'MultiNet: Multi-Modal Multi-Task Learning for Autonomous Driving', in *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, Jan. 2019
 - [19] J. Mund, A. Zouhar, L. Meyer, H. Fricke, and C. Rother, 'Performance Evaluation of LiDAR Point Clouds towards Automated FOD Detection on Airport Aprons', in *Proceedings of the 5th International Conference on Application and Theory of Automation in Command and Control Systems - ATACCS '15*, Toulouse, France, 2015, pp. 85–94
 - [20] J. Mund, A. Zouhar, and H. Fricke, 'Performance Analysis of a LiDAR System for Comprehensive Airport Ground Surveillance under Varying Weather and Lighting Conditions', 2018, p. 7.
 - [21] J. Mund, F. Michel, F. Dieke-Meier, H. Fricke, L. Meyer, and C. Rother, 'Introducing LiDAR Point Cloud-based Object Classification for Safer Apron Operations', p. 10.
 - [22] A. Geiger, P. Lenz, and R. Urtasun, 'Are we ready for autonomous driving? The KITTI vision benchmark suite', in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, Jun. 2012, pp. 3354–3361
 - [23] A. W. Steinforth and J. G. Eden, 'Speckle-Free Imaging Lidar without Motion Blur using Moderate-Coherence, Nanosecond-Pulsed Lasers', in *Imaging and Applied Optics 2019 (COSI, IS, MATH, pcAOP) (2019)*, paper IW3B.2, Jun. 2019, p. IW3B.2
 - [24] L. Groll and A. Kapp, 'Effect of Fast Motion on Range Images Acquired by Lidar Scanners for Automotive Applications', *IEEE Trans. Signal Process.*, vol. 55, no. 6, pp. 2945–2953, Jun. 2007
 - [25] 'Sesar Automated Airport: problem definition', Edition 02.01.00, 2017
 - [26] J. Mund, L. Meyer, and H. Fricke, 'LiDAR Performance Requirements and Optimized Sensor Positioning for Point Cloud-based Risk Mitigation at Airport Aprons', p. 8.
 - [27] T. Möller and B. Trumbore, 'Fast, Minimum Storage Ray-Triangle Intersection', *J. Graph. Tools*, vol. 2, no. 1, pp. 21–28, Jan. 1997
 - [28] R. E. Kalman, 'A New Approach to Linear Filtering and Prediction Problems', presented at the Transaction of the ASME, Journal of Basic Engineering, 1960
- R. Schubert, E. Richter, and G. Wanielik, 'Comparison and Evaluation of Advanced Motion Models for Vehicle Tracking', p. 6