Sensor scan timing compensation in environment models for automated road vehicles

Jens Rieken and Markus Maurer
Technische Universität Braunschweig
Institute of Control Engineering
38106 Braunschweig, Germany
Email: <rieken,maurer>@ifr.ing.tu-bs.de

***Abstract*—Environment perception is one of the most important tasks for Advanced Driver Assistance Systems and Automated Driving. Various types of sensors are used concurrently and fused into a consistent representation of the vehicle's surrounding. With more than one device involved, timing behavior becomes a relevant aspect of such systems. Delays due to processing and transferring data need to be considered when fusing sensor data. While some sensors, such as camera devices, perceive the environment as a snapshot, other sensors are based on a scanning technique. The iterative acquisition principle of scanning sensors lead to considerable time differences between single measurements within a scan. In addition to transfer and processing delays, these time differences have to be considered properly for a comprehensive environment representation.**

This paper focuses on time-related effects resulting from using a scanning sensor. The influence on two commonly utilized environment representations is discussed, grid-based and object-based approaches for stationary and movable elements, respectively. We present a comprehensive approach to incorporate the ego motion into grid-based models. We propose an adaptive prediction horizon for object tracking algorithms, based on sensor scan timing characteristics. Performance gains are evaluated with simulated data and verified within a real-world application. The presented algorithms are computationally feasible and real-time capable on a standard PC.

I. Introduction

Advanced Driver Assistance Systems (ADAS) and Automated driving have gained much attention during the last few decades. Many industrial and scientific projects have shown impressive demonstrations in this field of research. The perception, abstraction and understanding of the vehicle's environment are fundamental parts of these projects. Precise perception and representation of relevant elements are key prerequisites to offer a safe and comfortable driving experience for the passengers as well as other traffic.

Various types of sensors are combined for the perception task, either to provide redundancy or to overcome sensor-specific limitations. Information obtained from different sensors is fused in order to generate a comprehensive environment representation. Independent of the abstraction level of the fusion process, e.g. at sensor-level or track-level [1], and its domain (temporal, spatial), accurate timing information is crucial for this task. This demands precise knowledge about the timing behavior of the involved systems and the perceived sensor data. Constant as well as varying delays (i.e., *latencies* and *jitter*) resulting from data acquisition, pre-processing stages, and data transfer need to be considered. Apart from processing delays, the knowledge of the time-stamp at which a specific measurement was obtained by the sensor is an important factor. Vision-based sensors, such as mono and stereo camera devices, perceive the environment as a snapshot of the current scene. Rotating sensor devices such as laser scanners (Lidar sensors), however, require time for their sampling process. In general, this time span is referred to as *scan duration* or *sensor sweep time*. The measurements of a rotating system are usually not reported instantaneously, but are collected and transferred to the next stage at a given condition. In addition to other delays, the timing behavior of such a rotating sensor requires special consideration.

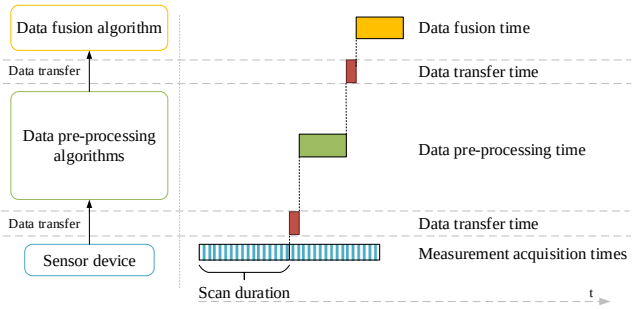


Fig. 1. Left: Schematic of a distributed environment perception system. Sensor device, data pre-processing and data fusion are located at different computation units. Each component of the processing chain requires processing time and thus delays data fusion; relevant time spans and timestamps are plotted on the right hand side. Scanning sensors do not perceive their environment at an instance, but at several consecutive measurement times. This results in a considerable duration of the scanning process, which needs to be regarded.

Problem statement

For automated road vehicles, the environment and the host vehicle itself move during the scan process of a rotating sensor device. In general, Lidar sensors retrieve multiple measurements from a physical target's contour, thus leading to multiple changes of the relative position between the sensor and a given target during the scan. In order to cluster measurements for a complete contour extraction, multiple

measurements are commonly collected and processed as a single scan. Although assumed in many cases, each cluster was not perceived at one single timestamp, but within a given scan duration. This influences the contour estimation of moving targets [2].

The fact that every element of the vehicle's environment is perceived at a different time within a sensor scan has considerable influence on environment perception tasks. For simplification, all measurements within a scan, and thus the extracted clusters, are usually assumed to be observed at the same timestamp. This assumption leads to systematic errors during the modeling and filtering process, which become even more critical at higher relative velocities between the objects or longer scan durations of the sensor devices.

Contribution and outline of this paper

This paper focuses on the effects on environment perception algorithms resulting from the usage of scanning sensor devices. We will introduce a simple model of representing the scanning properties of such devices and use this model to improve the perception algorithms. For the modeling of stationary environments with grid-based models, we will consider precise ego motion information during the sensor scan to improve the estimation quality. We will present an adaptive prediction scheme for object tracking algorithms which accounts for the different timestamps of new objects hypotheses. The performance of this algorithm will be compared to a standard approach which assumes all measurements to be generated at the same time. Evaluation is done by simulation and the application to real-world sensor data in the context of the project *Stadtpilot* [3]. In this paper, we focus on the effects within these two modules. Additional algorithms required for perception tasks are out of scope and further described in previous publications [4], [5], [6], [7].

The remaining part of this paper is organized as follows. Sec. II gives an overview of approaches for timing and delay consideration within perception systems. Sec. III explains system prerequisites regarding clock synchronization and ego motion representation. Sec. IV and V deal with the proposed adaptions made for grid-based and object-based representations, along with an evaluation of accuracy improvements. Sec. VI concludes with a summary and outlook on further research.

II. Terminology and Related work

The need for modeling and accounting of timing-related parameters has been addressed in several publications. Many approaches consider timing-related effects within a specific application or bounded to a specific functional context. An overview of these approaches is given in the next paragraphs.

Various timestamps and delays appear during the processing and fusion of sensor data. Fig. 1 illustrates the basic composition of these timing parameters. Each measurement is extracted from the sensor's signal processing at the *measurement acquisition time*. In order to integrate measurements correctly, they must be processed taking this timestamp into account.

It is typically impossible to integrate new raw measurements into the environment model directly at their creation time. Signal conditioning and processing are required to extract relevant information from the raw signal. This extraction process takes time resulting in delays in the overall data processing and fusion process. Among these delays, the computation time of pre-processing algorithms (*data pre-processing time*) is usually the most dominant type. If the pre-processing and/or the fusion module are located outside the sensor device itself, additional delays occur from *data transfer times* between involved systems.

Measurement timestamps and different delays need to be known and considered for time-consistent data fusion. In general, the use of timestamps for measurements is required to keep track of the creation time of the data. Once a timestamp can be assigned to the measurements, the delays of pre-processing and data transfer can be accounted for in the fusion process.

A time-consistent generation of these timestamps relies on synchronized clocks among the involved devices. Basic principles on this topic are discussed by the authors in [8]. Different approaches to synchronize clock systems of different distributed entities in a multi-sensor network can be applied [9].

Estimating the occurring time spans can be tackled in different ways. Some approaches aim to determine specific parts of the overall processing delay (e.g. [10], [11]). Timing-relevant parameters might be estimated during runtime of a data fusion system [12] .

Other approaches focus on the consideration of specific effects related to timing parameters. In the case of different sensor devices with different non-constant latencies, the receive sequence of their data might not be in time-ascending order. This requires special consideration if typical recursive state estimators such as Kalman filters are applied. This is known as the *Out Of Sequence Measurements* (OOSM) problem in the field of object tracking and is topic of research (cf. [13], [14], [15]). In those tracking algorithms, multiple measurements within one sensor scan are typically considered as created at one single timestamp. In general, the internal timing structure of the measurements is not considered.

This becomes a relevant aspect when synchronizing raw data between different sensors. The authors in [16] consider the movement of the sensor during a scan sweep for the synchronization of a Lidar sensor with a camera system in order to create a proper assignment of Lidar range data to image color information.

Correct consideration of timing-related parameters is also mandatory for the estimation of stationary elements with moving platforms. Among others, grid-based approaches [17] are typical representation models for this purpose. In this concept, the pose of a sensor upon each scan has to be known [18], which can be derived from the ego motion between two consecutive scans. For higher accuracy, the ego motion during the creation of the scan might be considered additionally [19].

The ego motion in relation to its stationary environment

is also relevant for sensor-based localization approaches, like *Simultaneous Localization and Mapping* (SLAM). By considering the movement of the sensor throughout the scan process, an improvement of the localization quality can be achieved [20].

Regarding pre-processing algorithms on point cloud data, such as shape extraction and contour estimation, scanning a moving target leads to a distortion of its range image. This is known as *Motion-scan effect* and was investigated on simulated data [21] as well as for applications with sensors mounted on automated road vehicles [2].

Sensor scan timing characteristics are considered for stationary environment models already, but none of the aforementioned approaches addresses those effects for the detection and tracking of moving targets on the abstraction level of object hypotheses. To our best knowledge, an integrated concept for considering the sensor scan characteristics in an environment perception framework for automated road vehicles has not yet been published.

III. System prerequisites

The demand for time-correct processing of sensor data yields some requirements for the underlying system and its components, such as sensor devices and processing units. The algorithms presented in this contribution require a synchronized timebase among these components. Detailed descriptions of the techniques applied for synchronization are out of scope here and are further explained in e.g. [9] and [10]. It is assumed here, that the acquisition timestamp of each measurement is available and can be obtained either using a synchronized clock network or detailed internal timing information of the sensor.

In addition, a proper representation of the ego motion is required to address its changing poses while perceiving the environment. Sensors related to the ego motion estimation are regarded as an additional component for the perception task. Hence, they have to be synchronized to the same timebase as the environment sensors as well. Ego motion information can be obtained by using dead-reckoning systems which utilize incremental sensors such as wheel speed encoders for position and movement estimation. Fusing additional inertial movement data (from accelerometers and gyroscope sensors) leads to increased performance [22]. Dead-reckoning systems suffer from a long-term drift of their position estimate due to wheel slip effects and sensor drifts. This effect is negligible for the representation of the vehicle's environment as only a short-term, yet continuous movement estimate is required.

In the application at hand, a new position is estimated upon each new dataset and stored with the corresponding timestamp. A ring buffer structure is utilized to store movement information within a given time range. This concept provides access to a unified ego motion model for the subsequent modules within a short history and is expected to lead to more precise estimation results especially during high-dynamic driving maneuvers.

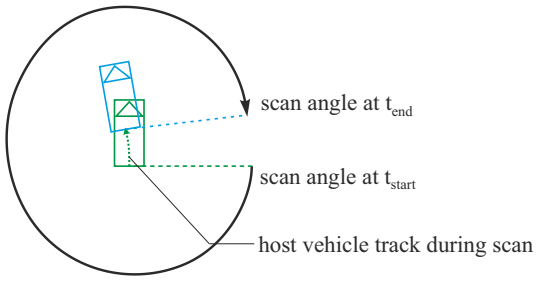


Fig. 2. Trace of the sensor scan point during one turn under moving conditions. Here, the scan angle trace for a 360° Velodyne HDL-64E S2 Lidar sensor is illustrated. The shown effect applies for other scanning systems as well.

IV. Scan time compensation for grid models

Grid-based approaches are a common representation form of the stationary environment. A grid map divides a given area into a finite number of cells. Each cell acts as a storage for information about the assigned area subset. A common piece of information is the presence or absence of obstacles (*occupancy grid*), but arbitrary types of information might be stored and considered for a comprehensive description of the stationary environment (e.g. [5], [6], [7]). For simplification, we will focus on a general grid model here; the presented algorithms are independent of the actual information type stored in the grid cells. The grid is typically defined in a local frame with the host vehicle moving through the covered area. By this, information about stationary environment can be inferred over several sensor scans.

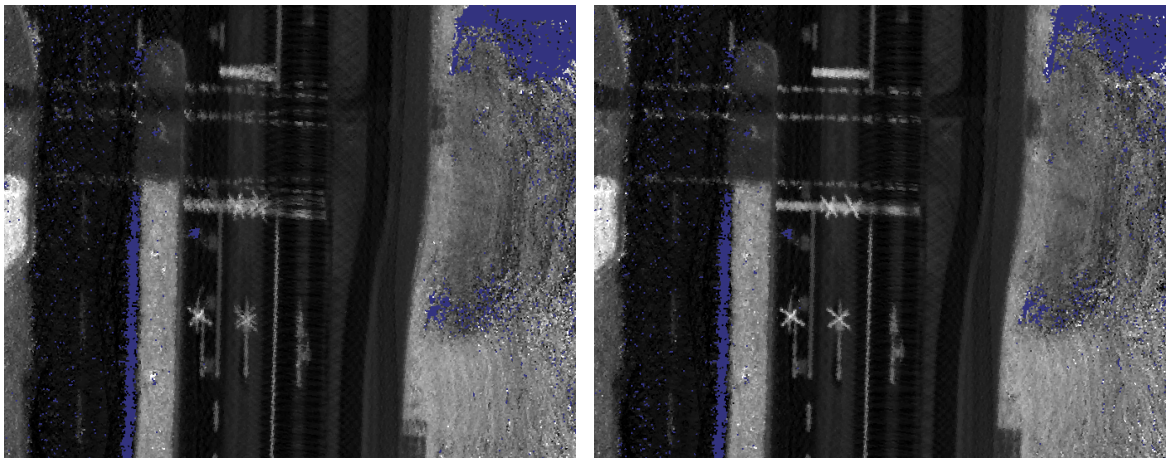
This requires knowledge about the sensor pose for each scan. Each new sensor scan s, received at time t, is coupled with a time-dependent pose of the host vehicle $\underline{p}(t) = (x(t), y(t), \varphi(t))^T$. The current pose $\underline{p}(t)$ of the host vehicle and the sensor mounting position are utilized to calculate the position of the scan origin within the grid frame.

When processing data from moving scanning sensors, as depicted in Fig. 2, accounting for the movement of the host vehicle during the scan process leads to increased accuracy. Each measurement is integrated into the grid cells by an inverse sensor model based on the pose $p(t_m)$ valid for the acquisition time of the measurement t_m. The relative pose is estimated from the ego motion data for timestamp t_m. The results of this extended update process are shown in Fig. 3. In particular, structures oriented perpendicular to the ego motion direction gain a sharper contour.

V. Adaption of object tracking algorithms

While the compensation of the ego motion is sufficient for modeling the stationary environment, additional effects need to be regarded when dealing with movable elements.

For the application at hand, a multi-object tracking approach based on an Extended Kalman Filter is utilized. Each object hypothesis establishes its own filter instance; the set of filter instances is referred to as *track database*. Each element of this database uses a dynamic model to describe the hypotheses' kinetic characteristics. New measurements are



(a) Grid map without ego motion compensation

(b) Grid map with ego motion compensation

Fig. 3. Effects of the ego motion compensation for grid models. The images show a reflectance intensity grid accumulated while driving on an inner-city road with the host vehicle driving in upward direction. Fig. 3a shows the resulting grid map without considering the ego motion during the scan. Lane markings perpendicular to the ego moving direction, in particular, are blurred. Fig. 3b shows the resulting grid map with proper ego motion compensation. The perpendicular lane marking features form sharper contour edges.

integrated by a prediction-correction scheme. The elements in the track database are predicted to the measurements' timestamp using the selected dynamic model. Association algorithms are utilized to solve the assignment problem between the existing elements of the track database and the measurements of the current sensor scan.

A. Problem statement

In case of a rotating sensor device, the extracted object hypotheses have different timestamps based on the sensor characteristics. Thus, a common prediction timestamp cannot be applied, but needs to be adjusted according to each measurement's creation timestamp.

Neglecting this leads to an incorrectly assumed time interval and thus a time error Δt toward the correct timestamp for the prediction process of each object. This shows as systematic error in the hypotheses' state estimates. Here, we focus on the estimate of the targets' distances. However, similar effects arise for other state estimates as well. The error in the distance estimation scales with the time error and the relative velocity between the target object and the host vehicle. If assuming a constant velocity during the relevant time difference, the error of the estimated target distance Δd_{error} is proportional to:

$$\Delta d_{\text{error}} \sim \Delta t \cdot v_{\text{rel}} \tag{1}$$

For scanning sensors, the time error is usually dependent on the angle φ under which the measurement is obtained:

$$\Delta d_{\text{error}} \sim \Delta t\,(\varphi) \cdot v_{\text{rel}} \tag{2}$$

As the predicted state estimates are utilized in the association step as well, erroneous state estimates influence the results of this assignment task. This leads to false assignments and in consequence to track losses and/or delayed object initializations.

This systematic error only has small impact on the estimated state variance in single-sensor systems. Assuming Δt stays approximately constant over two subsequent scans the error shows as an offset of the objects' positions as far as the distance estimation is concerned. In multi-sensor systems, however, the systematic error is likely to propagate into the state variance estimation because multiple sensors updating one hypothesis usually do not share the same timing error.

B. Basic approach

An exact way to account for the different generation timestamps of the hypotheses is to provide the predicted states of the current tracks for each timestamp occurring in the incoming measurement list. The individual prediction approach ensures that the association step is performed without the systematic errors arising from different timestamps between the predicted hypotheses and the incoming measurements.

However, high computational costs arise from large numbers of prediction iterations, especially when dealing with a large number of existing hypotheses in combination with a large number of incoming object hypotheses. For n existing hypotheses and m new measurements, a total number of $m \cdot n$ predictions is required. In urban environments, a mean of 100 object hypotheses (tentative as well as stable tracks) and about the same number of new measurements were found to be processed by the object tracking module in each update cycle, making this approach computationally demanding. In this basic approach, the timestamps of measurements might be distributed arbitrarily within the incoming measurement list. For a scanning system, however, this approach can be optimized by taking the sensor scan timing properties into account.

C. Runtime-efficient approximation

For the subsequent processing of sensor data, a model of the sensor timing generation is required. Based on typical

sensor designs, we use the following description of the rotation properties to approximate the sensor timing generation. We focus on *horizontally rotating* sensors within the following paragraphs. However, similar approaches can be applied to regard vertically rotating devices as well. Scan timing behavior of a sensor device is defined by three sensor-specific values, as also shown in Fig. 4:

- Scan start angle φ_{start}
- Scan end angle φ_{end}
- Rotation rate ω_{sensor}

The scan start and end angle describe the field of view of each rotating sensor. The total scan duration Δt is calculated based on this field of view and the rotation rate ω_{sensor}. In addition to these parameters, the scan start timestamp of each sweep, assumed to be located at φ_{start}, is required to approximate each measurement's timestamp:

$$t_{\text{meas,approx.}} = f\left(\varphi,\, t_{\text{start}},\, \omega_{\text{sensor}}\right) \tag{3}$$

Next to t_{start} and ω_{sensor}, which are assumed constant within one scan, the approximated measurement time $t_{\text{meas,approx.}}$ depends on the azimuth angle φ of a detected target. Given Eq. 3 and the azimuth angles of the tracked object hypotheses, individual prediction time ranges can be calculated. This *adaptive prediction horizon* is utilized to reduce the number of required prediction steps.

At the beginning of each update cycle, all object hypotheses in the track database are expected to have the same timestamp t_{base}. A sensor-specific mapping function implements Eq. 3 and maps the azimuth angle $\varphi_{\text{object}_n}$ of each object hypothesis to an individual prediction target time t_{predict_n}. The resulting timestamp is then utilized to calculate an individual prediction time range $\Delta t_{\text{n}} = t_{\text{predict}_n} - t_{\text{base}}$ for each object hypothesis.

Being predicted by Δt_{n}, each hypothesis will approximate potentially detected objects, which a sensor might have perceived at the given azimuth angle, more accurately. Hence, object state prediction and ego motion compensation need to be performed only once for each hypothesis in the track database.

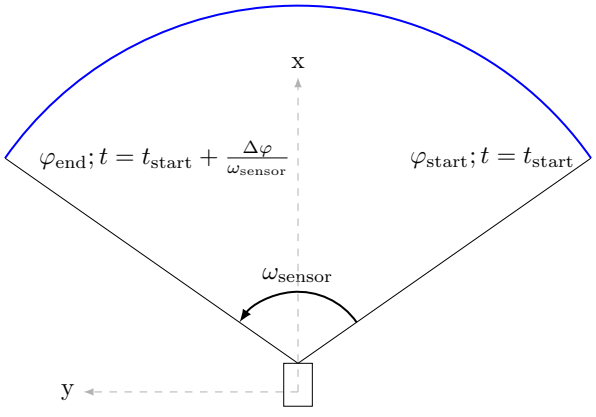


Fig. 4. Definition of sensor scan timing parameters for a rotating sensor

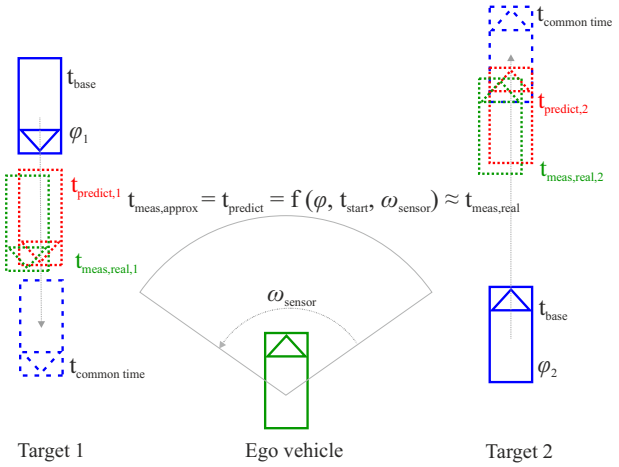


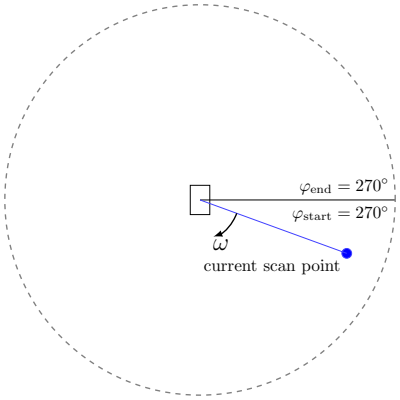
Fig. 5. Principle of the adaptive prediction horizon algorithm. Target 1 and 2 (solid blue boxes) are valid at timestamp t_{base} with their corresponding azimuth angles φ_{n}. If one single measurement timestamp $t_{\text{common time}}$ for all incoming measurements is wrongly assumed, all object hypotheses are predicted by the same time range (dashed blue boxes). The adaptive prediction horizon algorithm, in contrast, accounts for the sensor's timing characteristics and applies a customized prediction time $t_{\text{predict,n}}$ for each target. Hence, the predicted hypotheses (dotted red boxes) fit the position of the incoming measurements (dotted green boxes, perceived at $t_{\text{meas,real,n}}$) more accurately. (For this example, ego motion is neglected and all targets are assumed to move with constant velocity; positions and dimensions not drawn to scale.)

The mapping function can have arbitrary form. A typical mapping function for a rotating sensor describes a linear relation between the azimuth angle and the prediction time. As a result of this approximation, each object is predicted along with the movement of the sensor scanning point, like depicted in Fig. 5. An example plot of the proposed mapping function is shown in Fig. 6. For the presented example, the scan break point for completing a scan was set to 270° in the vehicle local frame. This provides consistent segments for hypothesis generation in the front, left and rear area of the host vehicle. Particular consideration of the segments near the scan break point is required to prevent possible segmentation issues, but is out of scope here.

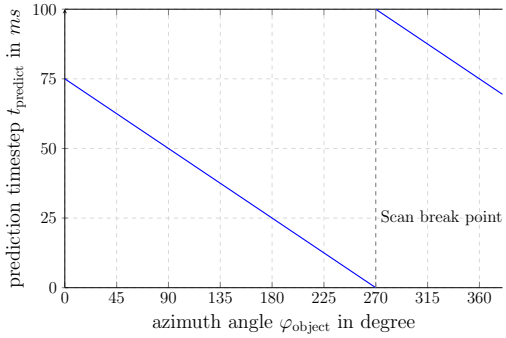
D. Results

The presented algorithm is evaluated in two steps. In the first step, the benefit of the algorithm is evaluated in a simulation framework. A single object was simulated approaching and passing the host vehicle on its left. This simulates a large relative velocity as well as a change of the object's relative azimuth angle. For the simulation, a virtual sensor device was extended to imitate a scanning sensor system based on a ray-tracing approach. This virtual sensor is part of an existing simulation framework [23]. The generated object list contains each object's position and the timestamp of the first hit by a virtual laser beam. The timestamp of the object is set to the time the corresponding beam was fired; this complies with the timing characteristics of a scanning system.

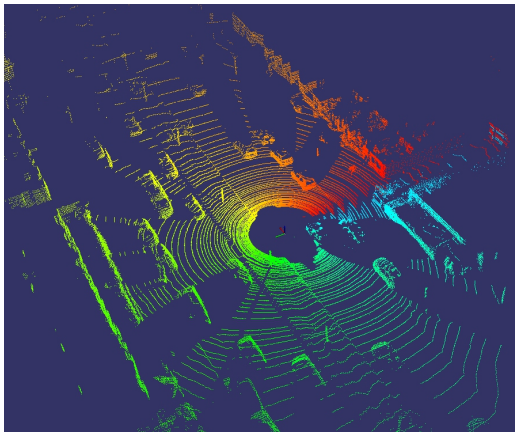
The results of two different configurations of the object tracking algorithm are compared to ground truth data obtained from the simulation framework. The first configuration



(a) Sensor scan timing diagram



(b) Sensor timing mapping function



(c) Velodyne point cloud data

Fig. 6. Sensor scan timing description (Fig. 6a) and the resulting mapping function (Fig. 6b) between the azimuth angle and the prediction time t_{predict} for a Velodyne HDL-64E S2 sensor with a scan duration of 100 ms and a scan break angle at 270°. Fig. 6c shows exemplary point cloud data of this sensor when mounted on the vehicle's roof, generated during a 360° sweep. Sensor points are colored by their generation age relative to the scan end (red: same age, green: 50 ms old, cyan-blue: 100 ms old).

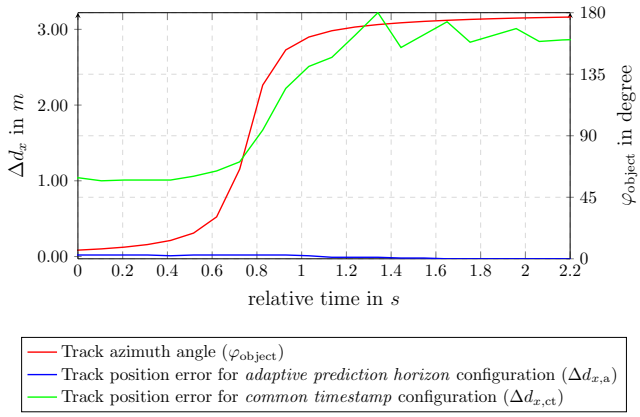


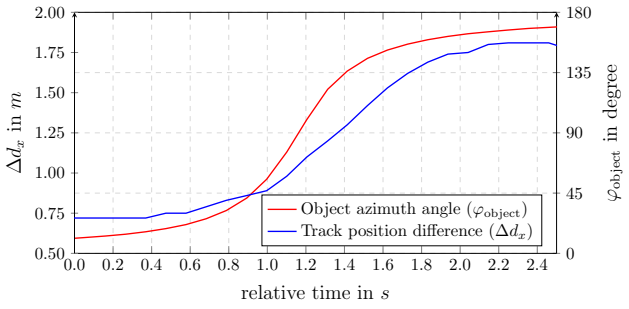
Fig. 7. Comparison of object tracking results for two different timing configurations (common timestamp assumption and the adaptive prediction horizon) with simulated data of an approaching object. The plots show the track's x-distance compared to ground truth ($\Delta d_x = d_{\text{x,[ct, a]}} - d_{\text{x,ref}}$). The position difference of the common timestamp assumption-configuration is dependent on the target's azimuth angle φ_{object} and increases with larger angles (φ_{object} plotted for comparison). The adaptive prediction horizon algorithm is independent of the object's azimuth angle and is able to estimate the object's position with a maximum difference of $\Delta d_{x,\text{max}} < 4\,\text{cm}$ compared to ground truth.

assumes all objects as generated at the same timestamp (e.g. the latest timestamp of the scan). The second configuration uses the object's individual timestamp and implements the adaptive prediction horizon.

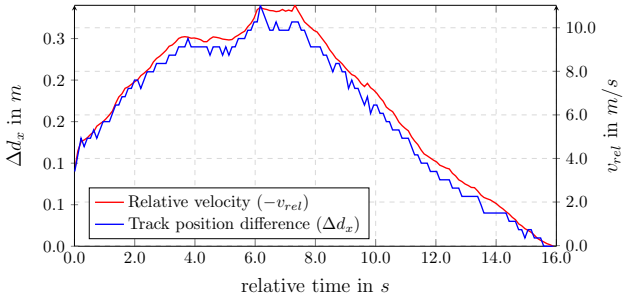
Fig. 7 illustrates the performance of the different configurations by showing the position estimate error obtained with ground truth data. The first configuration leads to an increasing error proportional to the azimuth angle φ_{object} of the tracked object. For larger azimuth angles near the end of the sequence, association of the incoming measurement to the existing object hypothesis fails in some update iterations resulting in an unstable position estimation. In contrast, the adaptive prediction horizon configuration is able to reduce the error to almost zero. Minor differences to ground truth ($\Delta d_x < 4\,\text{cm}$) occur due to the assumption of a constant azimuth angle for the target timestamp calculation. The considered object might change its relative position as a result of the prediction step; this change in position in turn might lead to an azimuth rate $\dot{\varphi}(t)$ of the object. This is relevant for objects performing a tangential movement in relation to the sensor, such as approaching traffic, but the resulting time error is negligibly small for the application at hand. Hence, this effect is not considered for now.

In the second step, the algorithm is evaluated with real-world sensor data using the aforementioned Velodyne Lidar sensor. The results are shown in Fig. 8. Both diagrams plot the position difference between the tracked object hypotheses resulting from a common timestamp assumption and those hypotheses estimated using the presented adaptive prediction horizon algorithm.

Fig. 8a illustrates the influence of neglected sensor timing behavior when tracking approaching and passing traffic left of the host vehicle. In this scenario, the relative velocity is



(a) Oncoming traffic



(b) Approaching a stationary target

Fig. 8. Results based on real-world data: The plots show the object's position estimate difference between the results of two timing configurations (common timestamp ($d_{\text{x,ct}}$) and adaptive prediction horizon ($d_{\text{x,a}}$)) in exemplary traffic situations.
Fig. 8a illustrates the position difference between the state estimates for an oncoming vehicle ($\Delta d_x = d_{\text{x,ct}} - d_{\text{x,a}}$). The changing azimuth angle of the tracked object results in a larger time error, which shows in a false position estimate when assuming all measurements at the same timestamp.
Fig. 8b shows the position difference when approaching a stationary target. The timing error is constant, but the relative velocity towards the tracked object leads to a false position estimate and lets the target appear at a further distance.

nearly constant, but the relative angle between the sensor and the tracked object changes rapidly. Due to the scanner's rotation characteristics, the timing error increases with the target's relative angle. This leads to a larger position error for larger relative angles up to half of the vehicle's length. The tracked object appears in a closer distance to the host vehicle when neglecting timing characteristics. These results are consistent with those from the simulation stage.

Fig. 8b illustrates the results when approaching an ongoing vehicle waiting at a traffic light. The tracked object keeps its relative angle towards the host vehicle. Hence, the timing error is constant in this sequence; during the approach phase a relative velocity is estimated. The position error is proportional to the negated relative velocity. Thus, the target object appears in a greater distance when neglecting the timing characteristics.

VI. Conclusion and Outlook

In this paper, we have focused on timing characteristics of rotating sensor devices and their proper consideration in environment perception tasks of a moving platform within moving environments. We have outlined the influence of sensor rotation characteristics on different perception tasks, namely grid-based approaches for stationary elements and object-based approaches for movable elements.

We have shown, that the correct consideration of sensor timing characteristics leads to more accurate results within both representations. Ego motion compensation based on the exact measurement timing leads to sharper features in grid-based models and thus increases the performance of subsequent extraction modules. We have identified systematic errors arising from neglecting the proper sensor timing within object tracking algorithms. By deriving an accurate hypotheses creation time based on the sensor's timing characteristics and considering them during the filtering process, a consistent representation of the surrounding moving traffic has been achieved.

Based on the presented algorithms, we have been able to create a time-consistent view for stationary as well as movable elements in the host vehicle's surrounding. Future research will address the incorporation of multiple sensor systems into the grid and tracking modules; for this step, the proposed algorithms define the basis for a time-consistent data fusion and environment representation.

Acknowledgment

The authors wish to thank the members of the Stadtpilot team for their support, especially Fabian Schuldt for his valuable support regarding simulation frameworks.

References

[1] M. Darms, "Data Fusion of Environment-Perception Sensors for ADAS," in *Handbook of Driver Assistance Systems*, 3rd ed., H. Winner, S. Hakuli, F. Lotz, and C. Singer, Eds. Springer, 2016.
[2] L. Gröll and A. Kapp, "Effect of Fast Motion on Range Images Acquired by Lidar Scanners for Automotive Applications," in *IEEE Transactions on Signal Processing*, vol. 55, no. 6. IEEE, 2007, pp. 2945–2953.
[3] J. M. Wille, F. Saust, and M. Maurer, "Stadtpilot: Driving autonomously on Braunschweig's inner ring road," in *IEEE Intelligent Vehicles Symposium (IV)*. San Diego, USA: IEEE, 2010, pp. 506–511.
[4] J. Choi, S. Ulbrich, B. Lichte, and M. Maurer, "Multi-Target Tracking using a 3D-Lidar sensor for autonomous vehicles," in *IEEE Intelligent Transportation Systems (ITSC)*. The Hague, Netherlands: IEEE, 2013, pp. 881–886.
[5] R. Matthaei, G. Bagschik, J. Rieken, and M. Maurer, "Stationary Urban Environment Modeling using Multi-Layer-Grids," in *International Conference on Information Fusion (FUSION)*. Salamanca, Spain: IEEE, 2014.
[6] J. Rieken, R. Matthaei, and M. Maurer, "Benefits of Using Explicit Ground-Plane Information for Grid-based Urban Environment Modeling," in *International Conference on Information Fusion (FUSION)*. Washington D.C., USA: IEEE, 2015.
[7] J. Rieken, R. Matthaei, and M. Maurer, "Toward perception-driven urban environment modeling for automated road vehicles," in *IEEE Intelligent Transportation Systems (ITSC)*. Gran Canaria, Spain: IEEE, 2015, pp. 731–738.
[8] F. Sivrikaya and B. Yener, "Time synchronization in sensor networks: A survey," *IEEE Network*, vol. 18, no. 4, pp. 45–50, 2004.
[9] N. Kaempchen and K. Dietmayer, "Data Synchronization Strategies For Multi-Sensor Fusion," in *Proceedings of 10th World Congress on Intelligent Transport Systems (ITS)*. Madrid, Spain: IEEE, 2003.
[10] M. Brahmi, K. Schueler, S. Bouzouraa, M. Maurer, K. H. Siedersberger, and U. Hofmann, "Timestamping and latency analysis for multi-sensor perception systems," in *Proceedings of IEEE SENSORS*. Baltimore, USA: IEEE, 2013, pp. 1–4.
[11] S. Park, B. Kim, K. Kim, Y. Son, and K. Yi, "Time delay compensation for environmental sensors of high-level automated driving systems," in *IEEE Intelligent Vehicles Symposium (IV)*. Seoul, South Korea: IEEE, 2015, pp. 555–560.

[12] A. Westenberger, T. Huck, M. Fritzsche, T. Schwarz, and K. Dietmayer, "Temporal synchronization in multi-sensor fusion for future driver assistance systems," in *IEEE International Symposium on Precision Clock Synchronization for Measurement, Control, and Communication (ISPCS)*. Munich, Germany: IEEE, 2011, pp. 93–98.

[13] Y. Bar-Shalom, "Update with out-of-sequence measurements in tracking: Exact solution," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 38, no. 3, pp. 769–778, 2002.

[14] M. Muntzinger, M. Aeberhard, S. Zuther, M. Mählisch, M. Schmid, J. Dickmann, and K. Dietmayer, "Reliable automotive pre-crash system with out-of-sequence measurement processing," in *IEEE Intelligent Vehicles Symposium (IV)*. San Diego, USA: IEEE, 2010, pp. 1022–1027.

[15] A. Westenberger, M. Muntzinger, M. Gabb, M. Fritzsche, and K. Dietmayer, "Time-to-Collision Estimation in Automotive Multi-Sensor Fusion with Delayed Measurements," in *Advanced Microsystems for Automotive Applications 2013*, J. Fischer-Wolfarth and G. Meyer, Eds. Springer International Publishing, Switzerland, 2013, pp. 13–20.

[16] S. Schneider, M. Himmelsbach, T. Luettel, and H. J. Wuensche, "Fusing vision and LIDAR - Synchronization, correction and occlusion reasoning," in *IEEE Intelligent Vehicles Symposium (IV)*. San Diego, USA: IEEE, 2010, pp. 388–393.

[17] A. Elfes, "Using occupancy grids for mobile robot perception and navigation," *Computer*, vol. 22, no. 6, pp. 46–57, 1989.

[18] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*. The MIT Press, 2005.

[19] M. Himmelsbach, F. V. Hundelshausen, and H.-J. Wuensche, "Fast segmentation of 3D point clouds for ground vehicles," in *IEEE Intelligent Vehicles Symposium (IV)*. San Diego, USA: IEEE, 2010, pp. 560–565.

[20] J. Choi and M. Maurer, "Hybrid map-based SLAM with Rao-Blackwellized particle filters," in *International Conference on Information Fusion (FUSION)*. Salamanca, Spain: IEEE, 2014.

[21] P. Ballard and F. Vacherand, "Simulation and Understanding of Range Images acquired in Fast Motion," in *IEEE International Conference on Robotics and Automation (ICRA)*, vol. 1. San Diego, USA: IEEE, 1994, pp. 2242–2247.

[22] A. Rudolph, "Quantification and Estimation of Differential Odometry Errors in Mobile Robotics with Redundant Sensor Information," *The International Journal of Robotic Research*, vol. 22, no. 2, pp. 117–128, 2003.

[23] F. Schuldt, B. Lichte, M. Maurer, and S. Scholz, "Systematische Auswertung von Testfällen für Fahrfunktionen im modularen virtuellen Testbaukasten (Systematic evaluation of test cases for driving functions with the modular virtual testing toolbox)," in *9. Workshop Fahrerassistenzsysteme (FAS 2014)*. Walting, Germany: Uni-DAS e.V., 2014.