IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2023.0322000

Modeling of Motion Distortion Effect of Scanning LiDAR Sensors for Simulation-based Testing

ARSALAN HAIDER[1,2], LUKAS HAAS[1,2], SHOTARO KOYAMA[3], LUKAS ELSTER[4], MICHAEL H. KÖHLER [5], MICHAEL SCHARDT[5], THOMAS ZEH[1], HIDEO INOUE[3], MARTIN JAKOBI,[2] ALEXANDER W. KOCH[2]

[1]IFM–Institute for Driver Assistance Systems and Connected Mobility, Kempten University of Applied Sciences, Junkersstrasse 1A, 87734 Benningen, Germany
[2]Institute for Measurement Systems and Sensor Technology, Technical University of Munich, Arcisstrasse 21, 80333 Munich, Germany
[3]Kanagawa Institute of Technology, Shimoogino 1030, 243-0292 Atsugi Kanagawa, Japan
[4]Institute of Automotive Engineering, Technical University of Darmstadt, 64289 Darmstadt, Germany
[5]Blickfeld GmbH Barthstrasse 12, 80339 Munich, Germany

Corresponding author: Arsalan Haider (e-mail: arsalan.haider@hs-kempten.de).

This paper shows results from the project VIVID (German Japan Joint Virtual Validation Methodology for Intelligent Driving Systems). The authors acknowledge the financial support by the Federal Ministry of Education and Research of Germany in the framework of VIVID, grant number 16ME0170. The responsibility for the content remains entirely with the authors.

ABSTRACT Automated vehicles use light detection and ranging (LiDAR) sensors for environmental scanning. However, the relative motion between the scanning LiDAR sensor and objects leads to a distortion of the point cloud. This phenomenon is known as the motion distortion effect, significantly degrading the sensor's object detection capabilities and generating false negative or false positive errors. In this work, we have introduced ray tracing-based deterministic and analytical approaches to model the motion distortion effect on the scanning LiDAR sensor's performance for simulation-based testing. In addition, we have performed dynamic test drives at a proving ground to compare real LiDAR data with the motion distortion effect simulation data. The real-world scenarios, the environmental conditions, the digital twin of the scenery, and the object of interest (OOI) are replicated in the virtual environment of commercial software to obtain the synthetic LiDAR data. The real and the virtual test drives are compared frame by frame to validate the motion distortion effect modeling. The mean absolute percentage error (MAPE), the occupied cell ratio (OCR), and the Barons cross-correlation coefficient (BCC) are used to quantify the correlation between the virtual and the real LiDAR point cloud data. The results show that the deterministic approach matches the real measurements better than the analytical approach for the scenarios in which the yaw rate of the ego vehicle changes rapidly.

INDEX TERMS LiDAR sensor; motion distortion; point cloud distortion; false positive; false negative; open simulation interface; functional mock-up interface; functional mock-up unit; highly automated driving, simulation-based testing

I. INTRODUCTION

LIGHT detection and ranging (LiDAR) sensors have become a key technology for autonomous driving due to their outstanding angular resolution and higher ranging accuracy than automotive radio detection and ranging (RADAR) sensors [1]. Autonomous vehicles use LiDAR sensors to obtain the environment's 3D point cloud, which can be used for obstacle detection and avoidance, simultaneous localization and mapping (SLAM), object recognition and tracking [2]–[5]. Commercially available automotive LiDAR sensors can be classified into two categories, non-scanning and scanning, based on their 3D imaging strategies, as shown in Figure 1 [6]. Flash LiDAR sensors are non-scanning types of LiDARs. They illuminate their entire field of view (FoV) at once by the laser source and do not contain any mechanical moving parts to steer the beam [7]. Non-scanning LiDAR sensors can measure up to $50\,\mathrm{m}$ and are used for forward collision warning (FCW) and blind spot detection (BSD) [8]. The scanning LiDAR sensors steer the laser beam in the FoV by using the mechanical moving parts to obtain the complete 3D view of the vehicle's surroundings at a specific frame rate [11]. Moreover, scanning LiDAR focuses the laser beam in a particular area in one shot. Therefore, depending on the frame rate, they can measure the objects at a $200\,\mathrm{m}$ distance

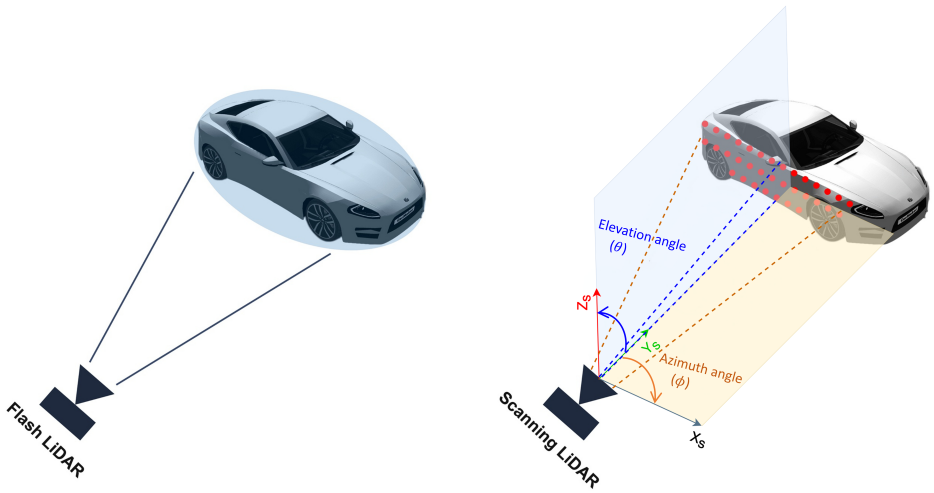

FIGURE 1. **Classification of LiDAR sensors based on their imaging strategies.**

with a typical horizontal and vertical resolution of $0.1°$ [12]. So far, the scanning LiDAR sensors are more suitable for automotive applications than flash LiDARs due to their higher angular resolution, broader FoV, and long-distance estimation capability [8]. That is why they are used for lane departure warnings (LDW), SLAM, FCW, and BSD [8]. Scanning LiDAR sensors require a specific duration to scan the complete environment. Therefore, fast relative motions between the LiDAR sensor and the objects in its FoV can lead to a distortion of the point cloud and a shift of the detected points of the objects, as shown in Figure 2. In addition, it is also possible that the LiDAR sensor may not be able to scan the intended object at all due to the fast relative motion. Once the 3D LiDAR data obtained during fast relative motion is provided to the LiDAR-based object recognition algorithm, such as PointPillars [17], it may detect the same object multiple times in the same frame, resulting in a false positive on object recognition level or it may fail to detect the intended object at all, resulting in a false negative. Figure 3 shows an exemplary false positive object recognition by the LiDAR-based object recognition algorithm from the motion-distorted LiDAR data. LiDAR systems use various approaches, including sensor fusion and data filtering using KALMAN filtering to detect these false positive and false negative detections. It should be noted that this work focuses only on modeling this effect. This motion-based distortion is known as the motion distortion effect, motion blur [9], and fast motion scan effect [10]. In the following, this effect is referred to as motion distortion. The magnitude of this effect increases with the increase in the relative translatory and rotatory motion between LiDAR and objects, as well as a decrease in the scanning frequency. For instance, if the relative velocity between the LiDAR sensor and object is $42.0\,\mathrm{ms}^{-1}$, and the LiDAR scanning duration is $0.1\,\mathrm{s}$, then the motion distortion from the first LiDAR scan

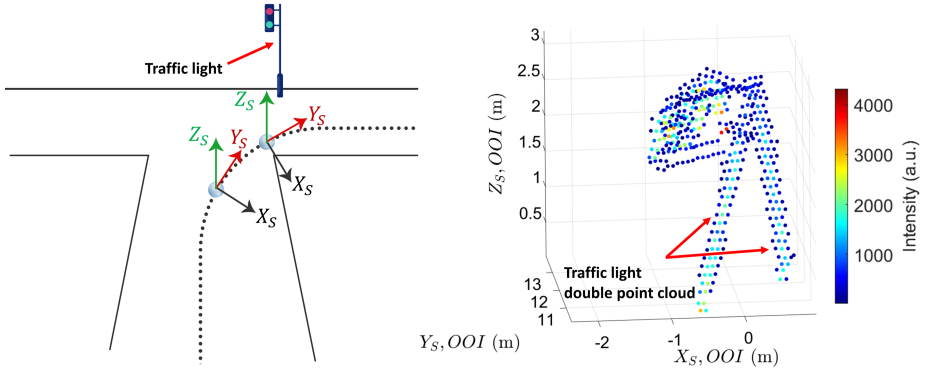

FIGURE 2. **Motion distortion effect due to the change in ego vehicle's yaw rate ψ. Specification of the scan pattern: horizontal FoV $\pm\,36°$ and vertical FoV $\pm\,15°$, frame rate $5.4\,\mathrm{Hz}$, horizontal angular resolution $0.4°$, vertical angular resolution $0.3°$, frame mode up and down scanning, and relative longitudinal velocity $-11.1\,\mathrm{ms}^{-1}$. The up-and-down scanning mode means the sensor will scan the environment twice in one frame to achieve higher vertical angular resolution [16]. For instance, if the frame mode is only up scanning, the vertical resolution will be $0.6°$. The change in the ego yaw rate ψ moves the LiDAR sensor's origin, resulting in a shift and dilation of the LiDAR points obtained from the traffic light in the horizontal direction. The traffic light appears twice in one frame due to the up-and-down scanning of the environment, leading to the shift of the point cloud of the same object. The sensor coordinates are represented by subscript *S*.**

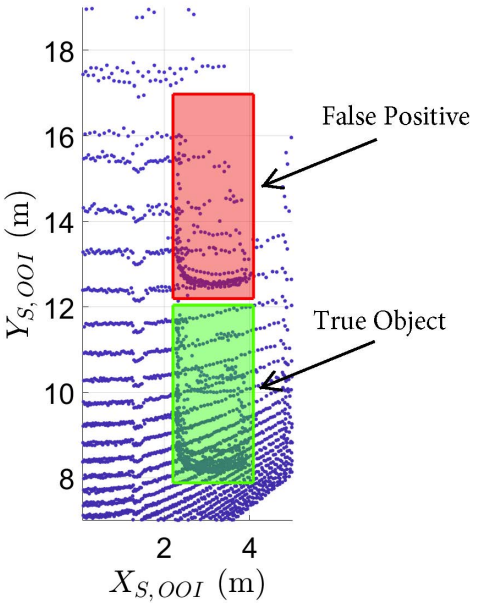

FIGURE 3. Exemplary visualization of a false positive object recognition by PointPillar using the simulated motion-distorted LiDAR data as shown in Figure 8 (c) with ground reflection. In this work, the closest position of the object relative to the ego vehicle predicted by PointPillar is considered an actual object, and the other position of the same object in the frame is regarded as a false negative.

point till the last will be $4.2\,\mathrm{m}$ in the worst-case scenario. The distorted point clouds lead to localization problems for the mapping [13] and a decrease in the point-to-point distance measurement accuracy that can negatively impact the object recognition algorithm's performance [14].

The scanning LiDAR sensors are part of the vehicle's safety-critical system; motion-related effects must be considered while designing the LiDAR system. In addition, their operational performance must be evaluated before using them for automotive applications. However, validating LiDAR sensors in the real world is costly and time-consuming [15]. Therefore, the automotive industry has started considering the type of approval based on simulation-based testing [14]. Consequently, it is required that motion-related effects should be modeled in the virtual LiDAR sensors with a high fidelity so they can exhibit the complexity and behavior of real LiDAR sensors. So, the obtained synthetic data can be used to evaluate the functional performance of the motion distortion correction algorithm and the LiDAR-based object detector.

In this work, we have modeled the motion distortion effect using ray tracing-based deterministic and analytical approaches in a virtual LiDAR sensor developed by the authors in their previous work [18]. The developed sensor model considers the modeling of the scan pattern, the complete signal processing toolchain, and sensor-specific imperfections to output the point cloud as realistic as possible. Such considerable imperfections include optical losses, inherent detector effects, effects generated by the electrical amplification, and noise produced by sunlight. The sensor model is developed using the standardized functional mock-up interface (FMI) and the open simulation interface (OSI). So, the sensor model can be tool-independent and easily integrated into any simulation tool that supports these interfaces without intellectual property infringement [18]. The presented approaches consider the sensor rotation, the rotation and the translation of the sensor mounting point, the sensor's scan pattern, and the rotation and the translation of objects in the sensor's FoV to model the motion distortion effect. In addition, we have conducted dynamic test drives at the Jtown proving ground in Japan [19] to obtain real data to validate the motion distortion effect modeling approaches. We have introduced a toolchain to bring real-world scenarios into the virtual environment and to compare the real and the virtual test drives frame by frame. Key performance indicators (KPIs) are defined to evaluate the virtual LiDAR sensor performance on the point cloud level. Moreover, we have used state-of-the-art metrics, including the mean absolute percentage error (MAPE), the occupied cell ratio (OCR), and the Barons cross-correlation coefficient (BCC), to quantify the difference between the simulated and the real data.

The paper is structured as follows: Section II describes the scanning LiDAR sensor working principle and the origin of the motion distortion effect. Then, an overview of the related work is given in Section III. The proposed approaches to model the motion distortion effects are given in Section IV. Section V describes the toolchain adopted to validate the motion distortion effect modeling and results. Finally, Sections VII and VIII provide the conclusion and the outlook.

II. WORKING PRINCIPLE OF SCANNING LIDAR SENSORS AND ORIGIN OF MOTION DISTORTION

Scanning LiDARs can be further classified into two categories, including 360° mechanical rotating and back-and-forth scanning LiDARs. The rotating scanning LiDAR sensors use a mechanical rotation system to spin the scanning part through the entire horizontal FoV. In contrast, the back-and-forth scanning LiDAR sensors do not have any mechanical rotating parts. Micro-electro-mechanical systems (MEMS), micro motion, scan/rotating mirror, and rotating prism-based LiDAR sensors have back-and-forth scanning patterns [20]. The motion distortion affects the rotating scanning LiDARs' operational performance less than back-and-forth scanning LiDARs [20]. Therefore, this work investigates the motion distortion effect for the MEMS LiDAR, a type of back-and-forth scanning LiDAR.

A. WORKING PRINCIPLE OF MEMS LIDAR SENSOR

A MEMS-based LiDAR consists of a laser and detector module (LDM) and a beam deflection unit (MEMS mirrors), as shown in Figure 4. The laser source emits laser pulses, so-called beams, and the beam deflection unit deflects the beam in different directions to obtain the point cloud of the environment. The photodetector receives the laser light partly reflected from the object's surface. The LiDAR sensor measures the time of flight (ToF) the laser light takes to hit an

object and to return to the detector to calculate the range. The measured range R can be written as [14]:

$$R = \frac{c \cdot \tau}{2}, \tag{1}$$

where c is the speed of light, and τ is the ToF.

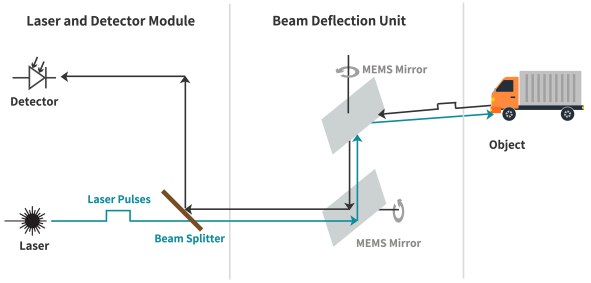

FIGURE 4. Block diagram of MEMS LiDAR sensor [14].

B. ORIGIN OF MOTION DISTORTION FOR SCANNING LIDAR

SCAN PATTERN

The automotive LiDAR sensors can adjust their scanning mode to the vehicle's driving situation [21]. Choosing the scan pattern is a trade-off between the frame rate, angular resolution, maximum detecting range, and 3D imaging. For example, if a use case is to detect a small object, then a LiDAR scan with high angular resolution will be required, which means more points and higher duration are needed to scan the environment, resulting in motion distortion. Figure 5 shows the exemplary scan pattern of the state-of-the-art MEMS-based LiDAR sensor Cube 1 from Blickfeld used in this research.

MOVING EGO VEHICLE

Figure 6 depicts the used vehicle and the sensor coordinate system with the different translatory and rotatory moving directions. This work uses the ISO 8855:2011 [22] conventions to describe the vehicle, sensor, and world coordinate system. The sensor coordinates are denoted by X_S, Y_S, Z_S, vehicle coordinates by X_V, Y_V, Z_V, and world coordinates as X_W, Y_W, Z_W. The magnitude of the motion distortion effect is more significant when the yaw rate ψ of the ego vehicle changes rapidly, for instance, while taking a left or right turn. The yaw rate ψ of the ego vehicle also rotates and translates the reference frame of the LiDAR sensor, which results in the motion distortion effect.

MOVING OBJECTS

The fast-moving object's lateral and longitudinal movement relative to the sensor is also a source of the motion distortion effect, for example, when an object crosses an intersection.

III. RELATED WORK

The degradation in the LiDAR sensor's performance due to the motion distortion effect is well described in the literature.

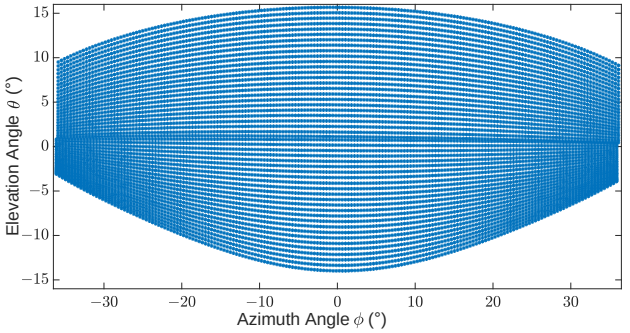

FIGURE 5. Exemplary scan pattern of Cube 1: $\pm 36°$ horizontal and $\pm 15°$ vertical FoV, 50 scan lines, $0.4°$ horizontal and $0.6°$ vertical angle spacing, frame rate $10.8\,\mathrm{Hz}$, maximum detection range $250\,\mathrm{m}$, and minimum detection range $1.5\,\mathrm{m}$. The scan pattern contains 9050 points.

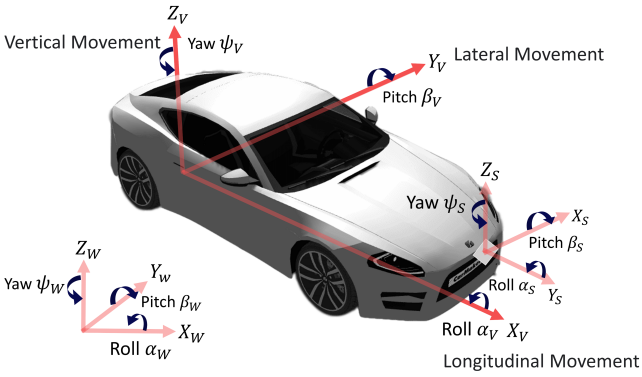

FIGURE 6. Vehicle, sensor, and world coordinate systems.

In addition, the correction of the motion distortion is also well explained in the literature. Ballard and Vacherand [23] have introduced a laser range finder scanning system simulator called LRF3S to study the impact of vehicle motion and scanning mechanism on the deformation of range images acquired by the LiDAR sensor. Their results show that the deformation of the range images acquired in fast motion is velocity-and trajectory-dependent. However, they have not compared the simulated and real measured range images acquired in fast motion to validate their modeling approach. Ono et al. [24] introduced an epipolar plane range image (EPRI) approach to generate 3D geometric models of urban spaces using LiDAR data. The proposed approach estimates the ego motion to remove the distortion from the LiDAR data because the motion-distorted LiDAR point cloud data can lead to inaccuracies in the resulting 3D models. The experimental data is used to demonstrate the working performance of the presented methodology for the city modeling. Zhang and Singh [25] present a real-time methodology for the odometry and mapping simultaneously by using the mechanical rotating LiDAR data. The proposed approach estimates the velocity of the LiDAR sensor to remove the distortion from the point clouds. They used the KITI data set to show the accuracy of the proposed approach. Gröll and Kapp [10] examine the

motion distortion effect on the distance map of automotive LiDAR sensors and quantify this effect by comparing it with statistical errors. They developed a simple motion model of the moving car's contour using the time-varying line segment to implement the motion distortion effect. The speed of the ego vehicle is assumed to be known by using odometry to accurately model and compensate for the motion distortion effect. Their paper focuses on the mathematical and theoretical aspects of motion distortion on the LiDAR sensor performance. Therefore, they did not show the visual representation of the motion-distorted simulated point clouds. Renzler et al. [26] have analyzed the motion distortion effect due to the fast-moving automotive scanning LiDAR sensor and proposed compensation. Their proposed methodology requires odometry information for motion distortion correction. Byun et al. [27] also presented a technique to correct the distorted LiDAR point clouds by incorporating the ego vehicle motion and posture information. Merriaux et al. [28] studied the impact of ego vehicle motion on LiDAR sensors. In addition, they also proposed the distorted LiDAR point cloud correction based on the CAN-bus data. Yang et al. [20] have fused LiDAR and camera data to estimate the velocity and to correct the motion distortion for the moving objects. McDermott and Rife [30] have introduced a velocity-corrected iterative compact ellipsoidal transformation (VICET), an extension of the normal distributions transform (NDT) algorithm to correct the motion distortion from the LiDAR data. The advantage of the proposed approach is that it does not require external sensor data to correct the motion distortion. Instead, it only requires a single LiDAR scan to remove motion distortion from the LiDAR data. The accuracy of the invented approach is shown by removing the motion distortion from the simulated and the real data of mechanical rotating LiDAR. They considered the simulated motion distortion due to the translation and rotational motion of the sensor. However, they have not explicitly explained the methodology used to obtain the simulated LiDAR data with motion distortion and how they validate the modeling of this effect. Gentil et al. [31] present a probabilistic framework to approximate the extrinsic calibration parameters of LiDAR and inertial measurement unit (IMU) to correct the motion distortion from the LiDAR data. They demonstrate the working performance of the proposed approach by correcting the motion distortion from the simulated and real data of mechanical rotating LiDAR. However, they have not explained the modeling approach adopted to obtain the motion-distorted simulated data. In [32], authors have presented inertial LiDAR localization autocalibration and mapping (IN2LAAMA) for extrinsic calibration of LiDAR and IMU to remove the motion distortion from the mechanical rotating LiDAR data. The presented approach is evaluated using the simulated and real data for the mechanical rotating LiDAR sensor. They used the Monte Carlo simulation to generate the simulated motion-distorted data. However, a detailed explanation of the motion distortion effect modeling and validation approaches are not discussed. Yoon et al. [33] introduced an approach to detect the dynamic and static objects from the 3D LiDAR data. The proposed approach does not require prior information about the object, such as a map or training data, and it also compensates for the motion distortion. The accuracy of the presented approach is validated by removing the motion distortion from the simulated and real mechanical rotating LiDAR data. They capture the motion distortion in the simulated data by activating each laser once in every simulation step. However, they have not explicitly mentioned how they validated their modeling approach. In addition, the simulated data does not consider the intensity of the LiDAR measurements, making it difficult to see the impact of material properties on the simulated point clouds. Shi et al. [34] have introduced an algorithm based on multioutput Gaussian process regression (MOGPR) to remove the motion distortion from the LiDAR point clouds. They have conducted the experiment on the simulated data, publicly available KITI data, and the real measured data obtained using their platform to show their proposed algorithm's effectiveness in eliminating motion distortion. They used the Monte Carlo simulation to get the motion-distorted simulated data. They consider motion distortion due to the circular and linear motion. However, they do not explain the modeling and validation methodology adopted to obtain the simulated motion distorted data.

The state-of-the-art research papers mentioned above focus on correcting the motion-distortion effect from the mechanical rotating LiDAR data. However, there is little focus on modeling and simulation of the motion distortion effect for simulation-based testing, especially for the MEMS-based automotive LiDAR sensor. The contribution of this paper is as follows:

- The introduction of the ray tracing-based deterministic and analytical approaches to model the motion distortion due to the ego and target vehicles translation and rotational motions for the simulation-based testing of different LiDAR scan pattern modes, motion distortion correction algorithms, and the LiDAR-based object recognition algorithms at the early stage of development.
- The generation of the digital twin of the real-world dynamic scenario and objects in the virtual world to validate the motion distortion effect modeling. The real measured and simulated point clouds are compared frame by frame to validate the proposed approaches, and the errors are quantified at the point cloud level using the MAPE, OCR, and BCC metrics.

IV. MODELING OF THE MOTION DISTORTION EFFECT

The modeling of the motion distortion effect is classified into two categories: deterministic and analytical approaches. The deterministic approach transmits the virtual LiDAR sensor scan pattern point by point like a real LiDAR sensor, and the motion distortion effect is modeled on the front end. In contrast, the analytical approach simultaneously shoots the complete scan pattern and directly models the impact of the motion distortion effect on the ideal ray tracing detection in

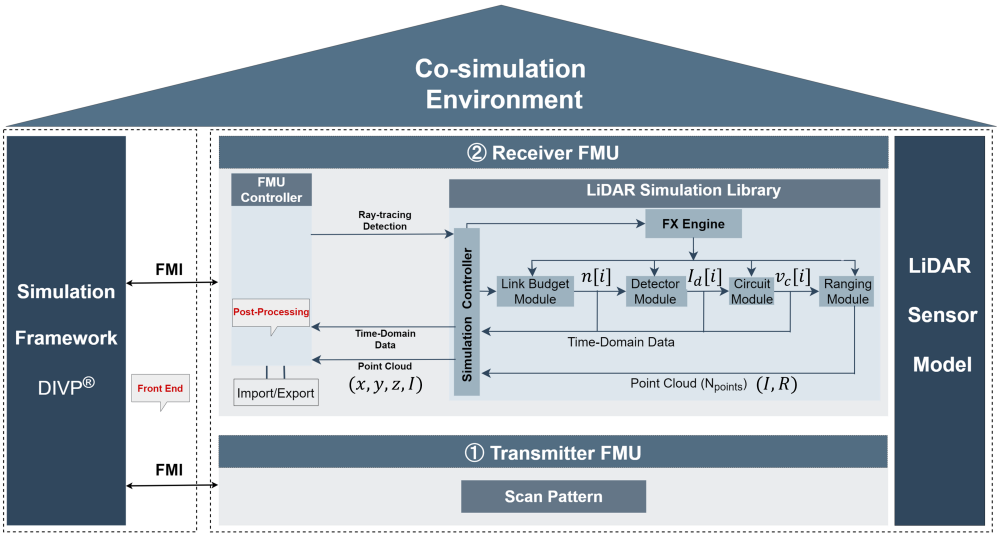

FIGURE 7. **Co-simulation framework of the proposed deterministic approach to model the motion distortion effect in a virtual LiDAR sensor.**

post-processing.

A. DETERMINISTIC MODELING APPROACH

Figure 7 depicts the toolchain and the signal processing steps to model the motion distortion effect due to the scan pattern and the ego and object motion in a virtual LiDAR sensor. The LiDAR sensor model consists of two submodules: transmitter and receiver. As mentioned earlier in Section I, the model is built using the standardized interface FMI and integrated into the virtual environment of the driving intelligence validation platform (DIVP®) by V-Drive Technologies [29]. It should be noted that the component that implements the FMI is known as functional mock-up unit (FMU) [35]. The virtual environment and LiDAR sensor model submodules, including the scan and receiver module, communicate via the FMI. DIVP® provides the ray tracing framework with a bidirectional reflectance distribution function (BRDF) that considers the direction of the incident ray θ, the material surface, and the color properties [36]. The LiDAR FMU model uses the ray tracing module of DIVP®. The material properties of the simulated objects, the angle-dependent spectral reflectance $R_\lambda(\theta)$, and the reflection types, including diffuse, specular, retroreflective, and transmissive, are specified in the material library of DIVP®.

1) Transmitter FMU

The virtual environment communicates with the transmitter FMU in the defined communication step size (call frequency) and passes the required input configuration via FMI. Then, the transmitter FMU verifies the input configuration and shoots the scan pattern point by point like a real MEMS-based LiDAR sensor in the virtual environment via FMI. For this use case, the transmitter FMU uses the scan pattern of the Blickfeld Cube 1, as shown in Figure 5. A detailed description of the Blickfeld Cube 1 scan pattern can be found in [16]. It should be noted that for this use case, each scan point gets shot after $0.0102\,\mathrm{ms}$ (t_{single}), and the complete scan pattern gets shot in $185.2\,\mathrm{ms}$.

2) Receiver FMU

The receiver FMU obtains the ray tracing data point by point, including the received power $P_r(t)$ and the 3D Cartesian coordinates of the detected echo of the ray. Then, the FMU controller calculates the virtual sensor spherical coordinates (d, θ, ϕ) from the Cartesian coordinates (X_S, Y_S, Z_S) and the time delay τ of the detected echo as given in the following equations

$$d_S = \sqrt{(X_S)^2 + (Y_S)^2 + (Z_S)^2}, \tag{2}$$

$$\theta = \tan^{-1}\left(\frac{Y_S}{X_S}\right), \tag{3}$$

$$\phi = \tan^{-1}\left(\frac{Z_S}{\sqrt{(X_S)^2 + (Y_S)^2}}\right), \tag{4}$$

$$\tau = \frac{2\,d}{c}, \tag{5}$$

where d is the distance, ϕ shows the azimuth angle, and the elevation angle is indicated by θ. Next, the FMU controller calls the LiDAR simulation library and passes the ideal ray tracing data, including time delay τ, azimuth ϕ, and elevation angles θ, and received power $P_r(t)$ to obtain the LiDAR point cloud as realistic as possible. The central component of the simulation library is the simulation controller. It is the primary interface component to provide interactions with the library, for instance, configuring the simulation pipeline, inserting ray tracing data, executing the simulations' steps, and retrieving the results. [18]

The simulation controller calls and passes the ray tracing data to the link budget module that implements the optics effect, sunlight, and calculates the photons' arrival over time

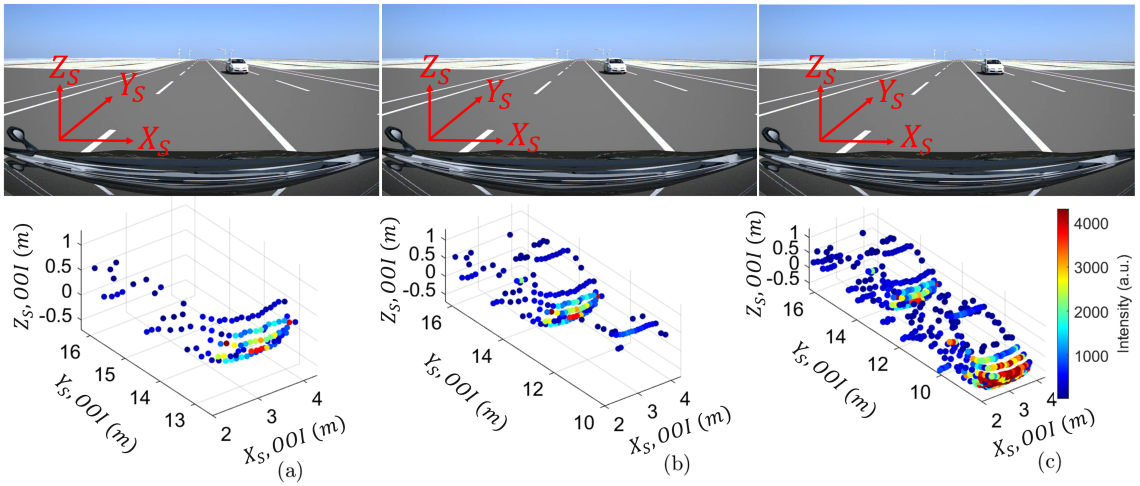

FIGURE 8. Exemplary multiple simulated point clouds obtained with a deterministic approach at different instances of a scan pattern as shown in Figure 5. One frame's scanning process is completed in $185.2\,\mathrm{ms}$. a) Point cloud obtained from the object of interest (OOI) after $20.5\,\mathrm{ms}$. b) Point cloud obtained from the OOI after $163.7\,\mathrm{ms}$. c) Point cloud obtained from the vehicle after $185.2\,\mathrm{ms}$. We have removed the points from the ground and surrounding to visualize the OOI better.

by sampling the received power $P_r(t)$ with the time interval of Δt. The detector module's task is to capture these discrete photons' arrivals $n[i]$ and convert them into a photocurrent $I_d[i]$ signal. This work implements the silicon photomultipliers (SiPM) as a detector. Next, the circuit module amplifies and converts the detector's photocurrent signal to a voltage signal $v_c[i]$ processed by the ranging module. The last part of the toolchain is the ranging module, which determines the range R and the intensity I of the target based on the voltage signal received from the circuit module for every reflected virtual LiDAR ray. The effect engine (FX engine) is a series of interfaces that interacts with environmental conditions or sensor-related effects and the blocks of the simulation pipeline. These interactions involve, for example, considering thermal noise in electrical components and increasing noise floor due to sunlight intensity. [18]

The detailed principle of the LiDAR simulation library can be found in the author's previous work [18]. Exemplary multiple point clouds obtained with a deterministic approach are shown in Figure 8. This approach can replicate the exact scanning behavior of the real LiDAR sensor, and the motion distortion effect will appear similar to the real world. On the other hand, this approach can be computationally expensive depending on the number of points N_{points} in the scan pattern and the computer's computational power. For instance, the simulation with a deterministic approach takes more than four hours to collect six minutes of virtual simulation data. The simulation is carried out on a computer with an Intel Core i7-9750H processor running at 2.60 GHz clock speed using 16 GByte RAM, running Windows 10 version Enterprise 21H2.

B. ANALYTICAL MODELING APPROACH

Primarily, the ray tracing modules provided by the simulation platforms shoot the complete scan pattern simultaneously like a flash LiDAR to offer real time sensor simulation. Therefore, it is challenging to model the motion distortion effect in the front end like a real-world LiDAR sensor. Consequently, an analytical approach is introduced to directly model the motion distortion effect on the ideal ray tracing detections in post-processing. The following steps are adopted to implement the motion distortion effect via the analytical approach to the ray tracing detection.

1) The simulation platforms should provide the detected ray identity number P_{id} to implement the motion distortion effect via an analytical approach. The identity number P_{id} is used to calculate the detected beam scanning duration t_{scan}.
2) The virtual target and ego vehicle's 3D position $(X_W,\ Y_W,\ Z_W)$, the orientation (yaw rate γ_W, pitch rate β_W, roll rate α_W), and the velocity $(v_{XW},\ v_{YW},\ v_{ZW})$ components are retrieved in world coordinates of the driving scenario. In addition, the 3D components of the relative radial velocity $\overrightarrow{v_{rV}}$ of the target in the ego vehicle coordinates are calculated by translation and rotation:

$$\overrightarrow{v_{rV}} = \left(\left(\overrightarrow{v_{\text{trg}W}} - \overrightarrow{v_{\text{ego}W}}\right) R_{rot}\right) \cdot \widehat{d_{\text{ego}V}}, \tag{6}$$

where $\overrightarrow{v_{\text{trg}W}}$ denotes the 3D components of the target vehicle velocity in the world coordinate system, $\overrightarrow{v_{\text{ego}W}}$ shows the 3D components of the ego vehicle velocity in the world coordinate system, R_{rot} is the rotation matrix to rotate the target vehicle coordinates to the ego vehicle, $\cdot$ is the inner product of the two vectors and $\widehat{d_{\text{ego}V}}$ is the unit vector pointing from the LiDAR sensor to the target. The rotation matrix R_{rot} is obtained by multiplying the ego vehicle rotation along the z-axis or yaw rate $\gamma_{\text{ego}W}$, y-axis or pitch rate $\beta_{\text{ego}W}$, x-axis or roll α_{ego_W} and can be written as

$$R_{\text{rot}} = R_{z,\text{rot}} R_{y,\text{rot}} R_{x,\text{rot}}, \tag{7}$$

$$R_{\text{rot}} = \begin{bmatrix} cos(\gamma_{\text{ego}_W}) & -sin(\gamma_{\text{ego}_W}) & 0 \\ sin(\gamma_{\text{ego}_W}) & cos(\gamma_{\text{ego}_W}) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} cos(\beta_{\text{ego}_W}) & 0 & sin(\beta_{\text{ego}_W}) \\ 0 & 1 & 0 \\ -sin(\beta_{\text{ego}_W}) & 0 & cos(\beta_{\text{ego}_W}) \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos(\alpha_{\text{ego}_W}) & -sin(\alpha_{\text{ego}_W}) \\ 0 & sin(\alpha_{\text{ego}_W}) & cos(\alpha_{\text{ego}_W}) \end{bmatrix}$$

[37]. The unit vector $\widehat{d_{\text{ego}_V}}$ pointing from the LiDAR sensor to the target can be written as

$$\widehat{d_{\text{ego}_V}} = \frac{\overrightarrow{d_{\text{ego}_V}}}{\left|\overrightarrow{d_{\text{ego}_V}}\right|}, \tag{8}$$

$$\overrightarrow{d_{\text{ego}_V}} = \left(\overrightarrow{d_{\text{trg}_W}} - \overrightarrow{d_{\text{ego}_W}}\right) R_{rot}, \tag{9}$$

where $\overrightarrow{d_{\text{ego}_V}}$ denotes the 3D components of the target position in the ego vehicle coordinate system, $\overrightarrow{d_{\text{trg}_W}}$ is the 3D position of the target in world coordinates, $\overrightarrow{d_{\text{ego}_W}}$ shows the ego vehicle 3D position in world coordinates and R_{rot} is a 3D rotation matrix to rotate the target coordinates into the ego vehicle. The relative radial velocity $\overrightarrow{v_{rV}}$ 3D components are used to calculate the value of the relative radial velocity v_{rV}, which can be written this:

$$v_{rV} = \sqrt{(v_{r,X_V})^2 + (v_{r,Y_V})^2 + (v_{r,Z_V})^2}. \tag{10}$$

3) In the next step, the analytical distance d_{analy_S} is calculated in the virtual sensor coordinates due to the motion distortion effect for all the ideal ray tracing detections received by simultaneously shooting the complete scan pattern.

$$d_{\text{analy}_S} = d_S + v_{rV}\, t_{\text{scan}}, \tag{11}$$

where d_S is the ideal ray tracing detection in the virtual sensor coordinates, v_{rV} is the relative radial velocity in the ego vehicle coordinates, and t_{scan} is the scanning duration of the detected virtual LiDAR ray which can be written as

$$t_{\text{scan}} = P_{\text{id}} \times t_{\text{single}}, \tag{12}$$

where P_{id} is the identity number of all the points received from the virtual environment and t_{single} is the time required for a single ray or scan point to shoot. Afterward, the cartesian coordinates are converted into spherical, and the signal processing toolchain and the sensor-specific imperfections, as given in Section IV-A, are applied to obtain the point cloud as realistic as possible. Figure 9 shows the exemplary simulated point cloud obtained by simultaneously shooting the complete scan pattern and analytically modeling the motion distortion effect.

V. VALIDATION OF THE MOTION DISTORTION EFFECT

A. VALIDATION TOOLCHAIN

Figure 10 shows the toolchain to validate the motion distortion effect modeling. The proposed toolchain consists of five steps: real-world scenario measurement, data analysis, scenario generation, integration of the sensor model into the virtual environment in a co-simulation framework, and validation of the simulated and real measured point clouds frame by frame.

STEP 1

The ego vehicle is equipped with LiDAR and camera sensors. The specification of the optical sensors used in this test cam-

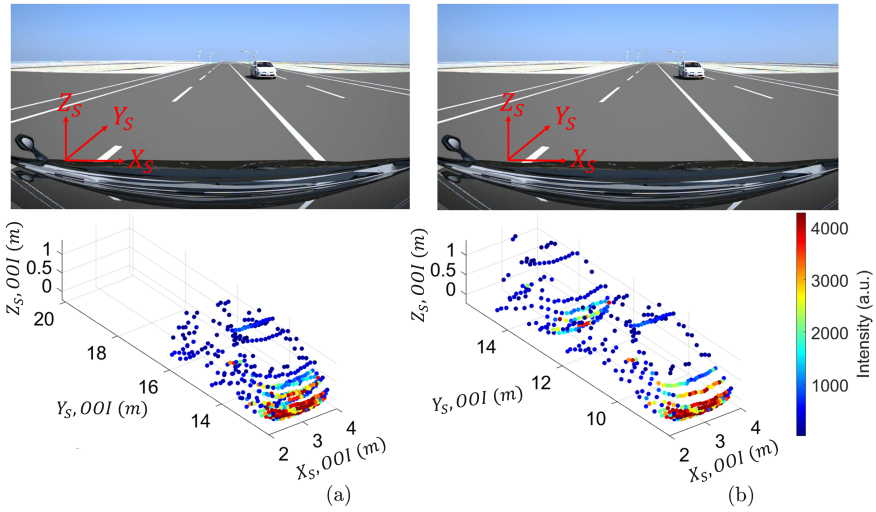

FIGURE 9. Exemplary simulated point clouds obtained by simultaneously shooting the complete scan pattern and analytically modeling the motion distortion effect. a) Simulated point cloud obtained by shooting the complete scan pattern simultaneously. b) Motion distortion effect modeled by using an analytical modeling approach on the point cloud. The constant relative velocity v_{rV} is $-27.8\,\text{m/s}$. The points from the ground and the surrounding are removed to visualize the OOI better.

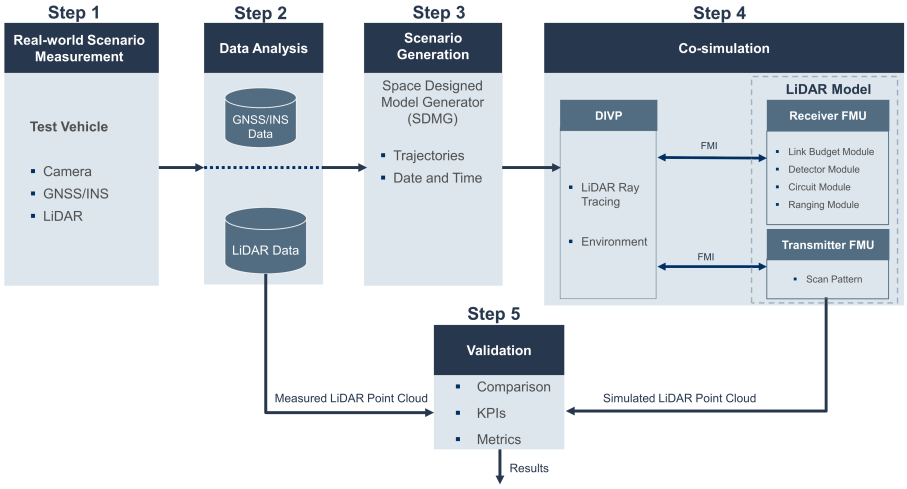

FIGURE 10. **Toolchain for the validation of the motion distortion effect.**

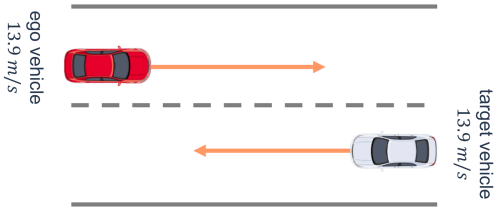

FIGURE 11. **Opposite-direction scenario performed at the proving ground to investigate the LiDAR motion distortion effect. The relative longitudinal velocity $v_{r,xV}$ between ego and target vehicles was $-27.8\,\mathrm{ms}^{-1}$. The cruise control keeps ego and target vehicle velocities constant. The LiDAR sensor frame rate f_{rate} is $5.4\,\mathrm{Hz}$.**

TABLE 1. **Specification of the sensors used in the test campaign.**

Sensors	Manufacturer	FoV	Range
IMX 490 (camera)	Sony Electronics	Azimuth ($\pm$ 15°) & Elevation ($\pm$ 8°)	300 m
Cube 1 [40]	Blickfeld	Azimuth ($\pm$ 36°) & Elevation ($\pm$ 15°)	250 m

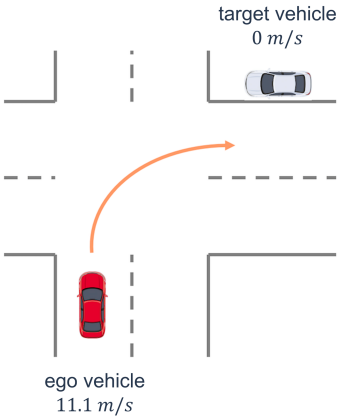

FIGURE 12. **Turning scenario to investigate the motion distortion effect due to the ego vehicle movement. The ego vehicle moves with a velocity $v_{ego,V}$ of $11.1\,\mathrm{ms}^{-1}$, and the target vehicle stands still. The LiDAR sensor frame rate f_{rate} is $5.4\,\mathrm{Hz}$.**

paign is given in Table 1. In addition, the ego and the target vehicles are equipped with a real-time kinematic (RTK)-based global navigation satellite/inertial navigation system (GNSS/INS) RT3000 v3 from OXTS [38] for reference measurements. The opposite and the turning scenarios, as shown in Figure 11 and Figure 12, are performed at the proving ground to obtain real data to validate the motion distortion effect modeling.

STEP 2

The recorded data of the sensors are stored via the Robot operating system (ROS). The GNSS/INS and LiDAR sensor data are analyzed to ensure all the sensor's time stamps and reference points are aligned.

STEP 3

The LiDAR and GNSS/INS data are imported into the space designed model generator (SDMG) tool to extract the maneuver and description of each traffic object. The GNSS/INS data is used to reconstruct the trajectory and the velocity profile of the ego and the target vehicles. In addition, the simulated scenario is exported as a test run infofile for the DIVP® platform 1.0.0. It should be noted that the static traffic objects, including poles and traffic lights, are modeled for this use case, but the surrounding environment of the test track is not modeled, such as grass, trees, etc.

STEP 4

The test run is constructed from the measurement of the real-world environment and is imported into the DIVP® platform 1.0.0. In addition, the scenario modeling is validated by

comparing the real and the simulated ego and target vehicle velocities and trajectories before its integration with the co-simulation framework. Afterward, the transmitter FMU is parametrized according to the Cube 1 settings. Finally, the LiDAR model and the simulation scenario are integrated into the co-simulation framework, and the point clouds are exported into *.bag* files.

STEP 5

The LiDAR FMU model and the real measured point clouds are imported into MATLAB R2021b. The validation of the complete toolchain and the motion distortion effect is performed in two steps:

1) In the first step, the three KPIs are defined for the sanity check of the motion distortion effect modeling, which includes
   - The number of received points N_{points} from the surface of the simulated and the real OOI in each frame.
   - The distance error $d_{\text{error}S}$ of the point clouds obtained from the real and the virtual OOI due to the motion distortion effect. The distance error $d_{\text{error}S}$ is the difference between the reference distance $d_{\text{ref}S}$ and the mean distance of the reflections received from the surface of the simulated and the real OOI $d_{\text{mean, meas/sim}S}$ and can be written as

$$d_{\text{error}S} = d_{\text{ref}S} - d_{\text{mean, meas/sim}S}. \tag{13}$$

   - The comparison between the mean intensity values I_{mean} of the received reflections from the simulated and the real OOI surface in each frame is used to validate the modeling of the OOI material properties.

   The detailed reasoning for choosing these KPIs can be found in the author's previous work [18].

2) For the detailed validation of the motion distortion effect modeling, a 2D (YX_S, XZ_S) binary and probabilistic occupancy grid (OG) map [39] of synthetic and real data are constructed and state-of-the-art metrics, including OCR and BCC, are applied to compute the correlation. An occupancy grid map represents an environment or space in the FoV of a LiDAR sensor divided into a grid of cells, with each cell indicating whether it is occupied or free by LiDAR detection, and the cell represents a small portion of the space being considered.

B. RESULTS

OPPOSITE DIRECTION SCENARIO

The exemplary LiDAR points received from the simulated and the real OOIs are shown in Figure 13. Figure 14 compares the number of points N_{points} and the mean intensity values I_{mean} of the real and the simulated vehicles. The results show that the number of received points N_{points} and the mean intensity values I_{mean} obtained from the OOI in the simulation and the real measurements match for frame numbers one to six. Still, a slight mismatch between these quantities can be observed for frames 7-9. Although the simulation scenario is constructed from the measurements of the real-world scenarios without manual interpolation, it is very challenging to reconstruct the exact real dynamic ego and target vehicle's position, orientation, and velocity frame by frame in a virtual environment. Therefore, the frames for which the simulated and the real OOI orientation and position differ have a mismatch between the number of received points. It can also be seen from Figure 14 (b) that the standard deviation of the real measured intensity values is higher than the standard deviation of the simulated intensity values. The possible reason behind this mismatch is the difference between the modeled and the real object material properties, which significantly influences the LiDAR sensor performance. However,

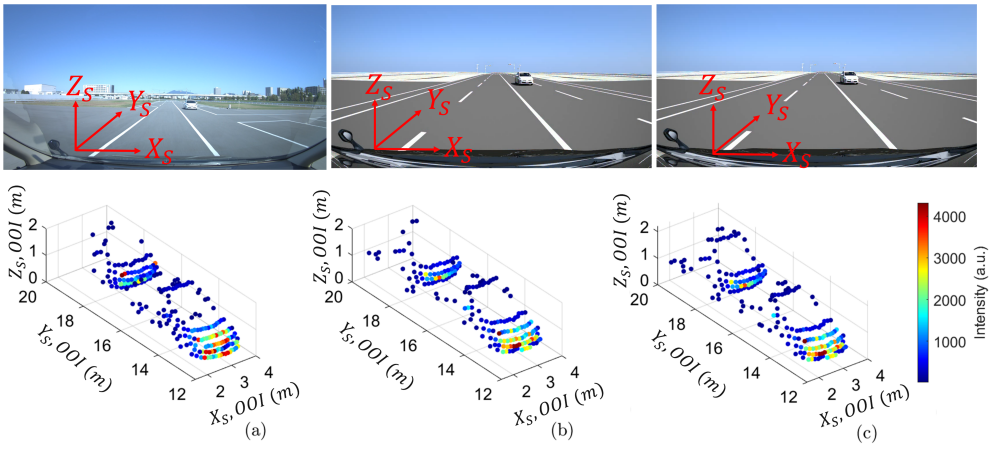

FIGURE 13. Comparison of real and simulated point clouds with the motion distortion effect. a) Exemplary real measured LiDAR point cloud. b) Exemplary simulated point cloud obtained with a deterministic approach. c) Exemplary simulated point cloud obtained with the analytical approach. The scanning frequency of the real and virtual LiDAR sensor is $5.4\,\text{Hz}$, and the relative velocity v_{rv} between the ego and target vehicles is $-27.8\,\text{m/s}$, which is why the LiDAR points obtained from OOI are shifted due to the motion distortion effect.

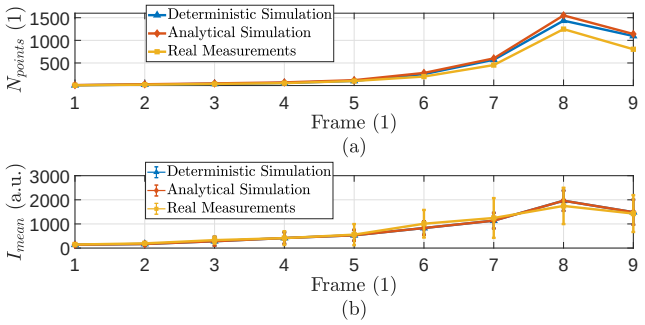

FIGURE 14. **Frame by frame comparison of the number of points N_{points} and mean intensity values I_{mean} received from simulated and real OOI. a) Number of received points N_{points} from OOI in simulation and real measurement matches for frame number one to six. However, a slight mismatch in the number of detections can be observed for frames 7-9. b) Mean intensity values I_{mean} of simulated and real target vehicle are approximately the same for all frames. Still, the standard deviation of real measured intensity values is higher than the standard deviation of the simulated intensity values.**

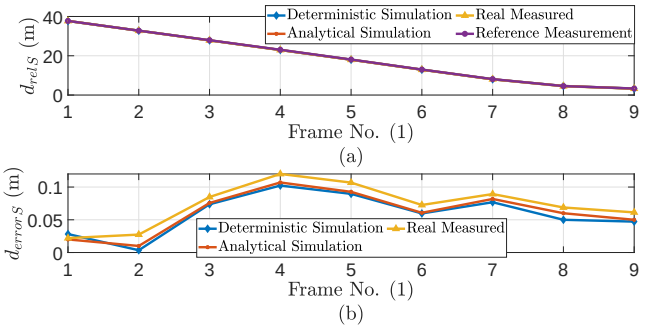

FIGURE 15. **Frame by frame comparison of reference, simulated, and real target relative distance d_{relS} and distance error d_{errorS}. a) Reference, simulated, and real OOIs relative distance d_{relS} in each frame. b) Distance error d_{errorS} of simulated and real vehicle frame by frame. The results show good agreement with each other, and a slight mismatch exists because it is tough to replicate the exact real-world targets' orientation and position in a virtual environment. The reference distance d_{refS} is calculated from the sensor reference point to the center of the target vehicle's front bumper.**

the simulation results with the deterministic approach match better with real measurements than those obtained with the analytical method because the virtual LiDAR sensor with the deterministic process shoots the complete scan pattern point by point and updates the virtual scene after every scan point shot.

Figure 15 a) shows the simulated and the real target vehicle relative distance d_{relS}, and Figure 15 b) gives the distance error d_{errorS} of each frame. The MAPE metric is used to quantify the difference between these quantities. Because it provides the magnitude of the errors relative to the real measured values in percentage, enabling the interpretation of the results to be more accessible compared to other metrics like mean squared error (MSE) or root mean squared error (RMSE) [41]. Mathematically, it can be written as

$$M = \frac{1}{n}\sum_{i=1}^{n} \left| \frac{y_i - x_i}{y_i} \right|, \tag{14}$$

TABLE 2. **Opposite scenario: MAPE of N_{points}, I_{mean}, and distance error d_{errorS} for deterministic and analytical approaches.**

KPI	Deterministic approach MAPE	Analytical approach MAPE
N_{points}	8.90%	14.30%
I_{mean}	8.83%	9.20%
d_{errorS}	0.10%	0.08%

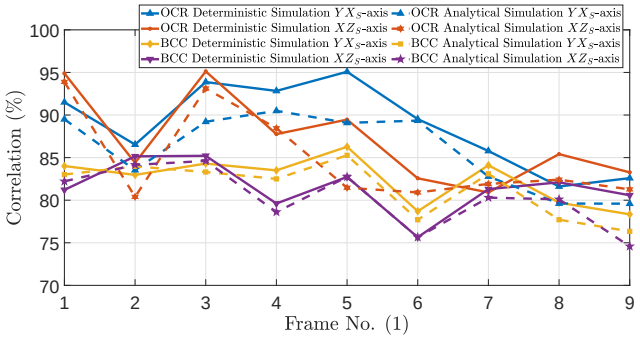

FIGURE 16. **Correlation comparison of simulated and real measured point clouds 2D (YX_S, XZ_S) binary and probabilistic OG maps. The frames for which the simulated and the real OOI's position and orientation differ have low correlation. The deterministic approach results match better with the real measurements than the analytical approach due to its virtual environment scanning methodology.**

where y_i is the measured value, the simulated value is denoted by x_i, and n shows the total number of values in a data set [42]. Table 2 provides the MAPE of these KPIs for the deterministic and the analytical approaches. Furthermore, the MAPE of the relative distance d_{relS} is 0.1% for the real measurement.

The 2D (YX_S, XZ_S) binary and probabilistic OG maps [39] are constructed, and the OCR and the BCC metrics are applied to validate the motion distortion effect modeling further. The OCR is applied to the binary OG map due to its simplicity and intuitive interpretation of the occupied cells (OCC). It is the ratio between the true cells classified as occupied (cells that are occupied in the simulated map and the real map) in the simulated map and the total number of OCC in the real map. Mathematically, it can be written as [43]:

$$OCR = \frac{\sum cell_{sim,\,map,\,occ\,true}}{\sum cell_{real,\,map\,occ}}. \tag{15}$$

BCC is used to compare the 2D probabilistic OG of the simulated and the measured point clouds because of its simplicity and intuition in interpreting the results and its ability to handle noise and variation compared to Pearson correlation [44]. It can be written as [45]:

$$BCC = \frac{\langle OG_s \cdot OG_r \rangle - \langle OG_s \rangle \langle OG_r \rangle}{\sigma(OG_s)\sigma(OG_r)}, \tag{16}$$

where the average value of the occupied and the free space cells of the simulated point clouds OG map is denoted by $\langle OG_s \rangle$ and the real OG map with $\langle OG_r \rangle$, the standard deviation of the simulated OG maps is shown by $\sigma(OG_s)$ and the real with $\sigma(OG_r)$.

Figure 16 shows the OCR and the BCC metrics results for the 2D (YX_S, XZ_S) binary and probabilistic OG map of the deterministic and the analytical approaches. It should be noted

TABLE 3. Opposite scenario: OCR and BCC metrics average correlation for all frames of deterministic and analytical approaches.

Metric	Deterministic appr. YX_S-axis OG map avg. corr.	Analytical appr. YX_S-axis OG map avg. corr.	Deterministic appr. XZ_S-axis OG map avg. corr.	Analytical appr. XZ_S-axis OG map avg. corr.
OCR	88.8%	85.9%	87.1%	84.8%
BCC	82.4%	81.4%	81.5%	80.3%

that the OG map cell size is adjusted according to the beam divergence of the real LiDAR sensor, which is $0.4°$ in the horizontal and the vertical direction in this case. For example, the OG map cell size width and height is $1.74\,\mathrm{m} \times 1.74\,\mathrm{m}$ at $250\,\mathrm{m}$ and $0.01\,\mathrm{m} \times 0.01\,\mathrm{m}$ at $1.5\,\mathrm{m}$. Table 3 provides the OCR and the BCC metrics average correlation for all frames of the deterministic and the analytical approaches.

The deterministic approach shows a higher correlation with the real measurements than the analytical approach because the virtual LiDAR sensor with a deterministic method shoots the scan pattern point by point like a real MEMS LiDAR sensor. In addition, the mismatch between the results exists because it is challenging to replicate the 100% real-world dynamic ego and target vehicle orientation, position, and velocity frame by frame in a virtual environment using the GNSS/INS data. The results of the deterministic approach can be further optimized by modeling the virtual environment more realistically. Still, this work focuses on generating the virtual environment using the sensor data without manual interpolation. Table 4 provides the exemplary calculation to show the real and the simulated vehicle orientation difference and the distance error $d_{\mathrm{error}S}$ impact on the OCR and BCC metric correlation. The BCC metric shows more sensitivity to the environmental and the motion distortion effect modeling

TABLE 4. Exemplary real and simulated vehicle orientation difference and distance error $d_{\mathrm{error}S}$ impact on OCR and BCC metric correlation. The simulated and real target vehicle is placed at $13\,\mathrm{m}$, and the OG map cell size is $0.09\,\mathrm{m} \times 0.09\,\mathrm{m}$.

Yaw diff. ψ_{diff}	Pitch diff. β_{diff}	Roll diff. α_{diff}	Distance error $d_{\mathrm{error}S}$	OCR corr. YX_S-axis	BCC corr. YX_S-axis	OCR corr. XZ_S-axis	BCC corr. XZ_S-axis
0°	0°	0°	0 m	86.4%	83.2%	95.1%	84.1%
2°	0°	0°	0 m	84.3%	70.5%	85.4%	75.0%
0°	2°	0°	0 m	82.5%	71.5%	85.2%	80.0%
0°	0°	0°	0.1 m	81.6%	70.9%	94.6%	82.3%
2°	2°	0°	0.1 m	79.2%	68.6%	89.6%	75.9%

than the OCR metric because it considers the average value and the standard deviation of the simulated and the real OG maps as given in Equation 16.

TURNING SCENARIO

The turning scenario shown in Figure 12 is replicated in the virtual environment using the GNSS/INS data to verify further the motion distortion effect modeling and the validation toolchain. The exemplary simulated and real point clouds obtained from the OOI for the turning scenario are shown in Figure 17. The simulated and the real point clouds obtained from the OOI are dilated due to the motion distortion effect. Figure 18 compares the number of received points N_{points} and the mean intensity values I_{mean} obtained from the simulated and the real OOI frame by frame.

For the number of points N_{points}, the simulation and the real measurement match for frames 1-9. However, a high mismatch in the number of points N_{points} can be observed, especially for frames 10-14, because when the ego vehicle turns right, it changes the origin of the sensor multiple times in one frame. Measuring the exact orientation of the sensor's origin while turning is challenging, so it is difficult to model

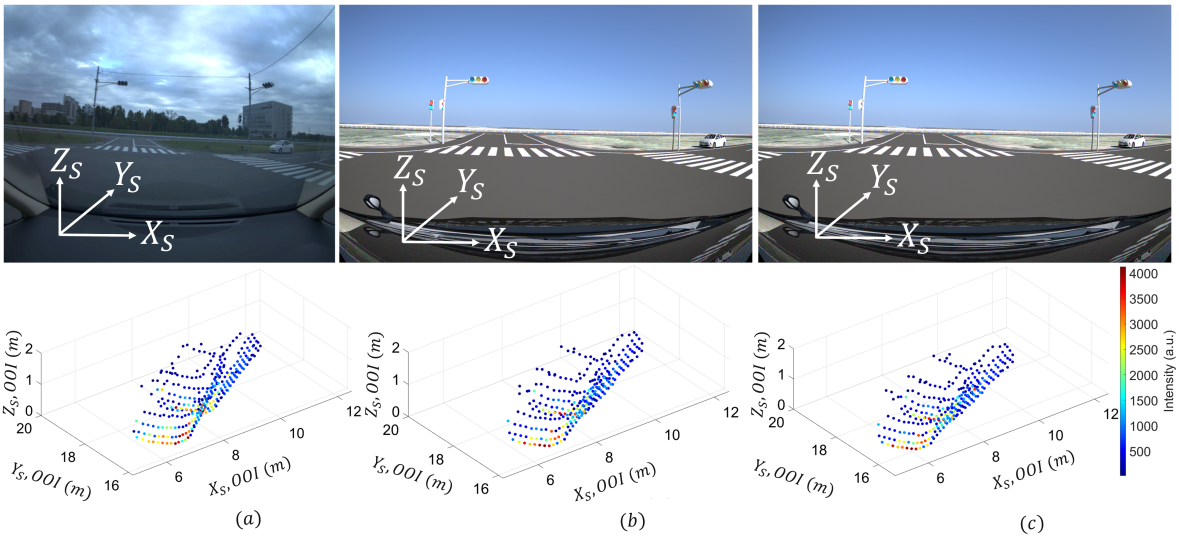

FIGURE 17. Comparison of real and simulated OOI point clouds. a) Exemplary real measured LiDAR point cloud for a turning scenario. b) Simulated point cloud obtained with a deterministic motion distortion approach. c) The simulated point cloud obtained with an analytical motion distortion approach. The asphalt and the static traffic object's material properties are not modeled, so the road and poles' reflections are removed to visualize the point clouds better. The scanning frequency of the real and virtual LiDAR sensor is $5.4\,\mathrm{Hz}$ and the relative velocity v_{rV} between the ego and target vehicles is $-11.1\,\mathrm{m/s}$, so the simulated and real point clouds get dilated due to the motion distortion effect.

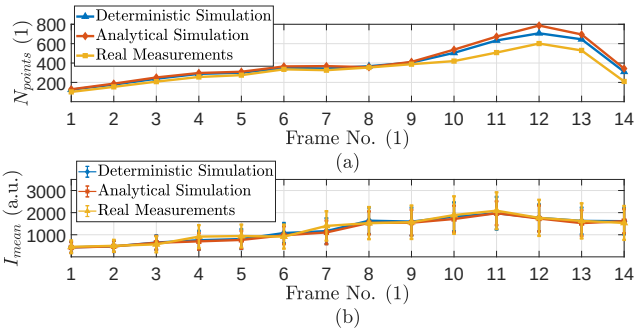

FIGURE 18. **Frame by frame comparison of the number of points N_{points} and mean intensity values I_{mean} received from simulated and real OOI. a) Number of received points N_{points} from the OOI in simulation and real measurement for each frame. (b) Real measured and simulated mean intensity values I_{mean}. The deterministic and analytical approach mean intensity values I_{mean} match the real measured intensity values. It can also be observed that the standard deviation of the real measured intensity values is higher than that of the simulated intensity values.**

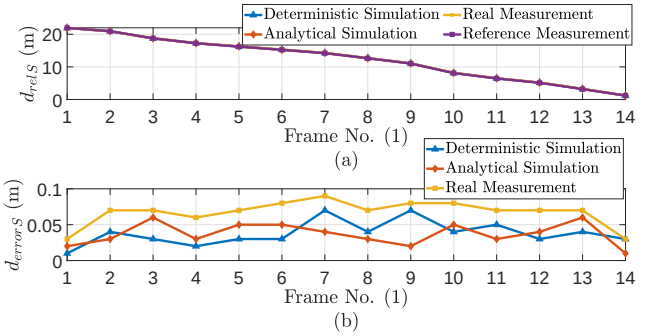

FIGURE 19. **Frame by frame comparison of reference, simulated, and real target relative distance d_{relS} and distance error d_{errorS}. a) Simulated and real target vehicle relative distance d_{relS} in each frame. b) Distance error d_{errorS} of simulated and real vehicle frame by frame. The results show good agreement with each other because the MAPE is less than 0.1% for the real measured and simulated distance error d_{errorS}. The reference distance d_{refS} is calculated from the sensor reference point to the center of the target vehicle's front bumper.**

it in a virtual environment using GNSS/INS data, as only the ego vehicle orientation is recorded. It can be seen from Figure 18 (b) that the real measured and the simulated OOI intensity values match well. However, the standard deviation of the real measured intensity values is higher than that of the simulated ones. The possible reason behind this deviation is the difference in the simulated and the real OOI material properties. However, a high-fidelity digital twin of the real OOI is used. It is still difficult to model the exact real OOI material roughness in the virtual environment. Figure 19 a) shows the reference, the simulated, and the real target vehicle relative distance d_{relS} for each frame. Figure 19 b) gives the distance error d_{errorS} of each frame. The MAPE metric given in Equation 14 is used to quantify these quantities' differences. Table 5 provides the MAPE of these quantities for the deterministic and the analytical approaches. Moreover, the MAPE of the relative distance d_{relS} is 0.09% for the real measurement.

TABLE 5. **MAPE of N_{points}, I_{mean}, and distance error d_{errorS} for the deterministic and analytical approach.**

KPI	Deterministic approach MAPE	Analytical approach MAPE
N_{points}	13.00%	22.30%
d_{errorS}	0.07%	0.06 %
I_{mean}	8.10%	8.80%

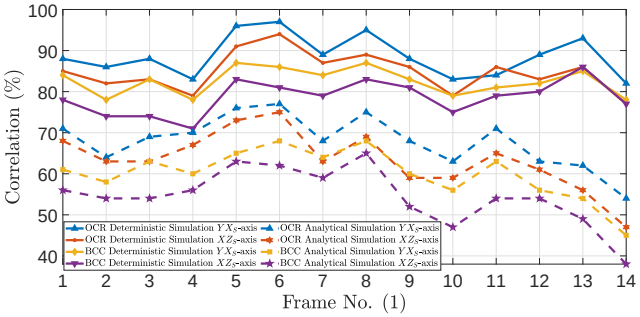

FIGURE 20. **Correlation comparison of simulated and real measured point clouds 2D (YX_S, XZ_S) binary and probabilistic OG maps for the turning scenario. The deterministic approach correlation with real measurements is better than the analytical approach because, in the deterministic approach, the ego vehicle orientation and position are updated every $0.0102\,\mathrm{ms}$, while in the analytical method, the orientation of the ego vehicle gets updated every $185.2\,\mathrm{ms}$.**

Figure 20 shows the OCR and the BCC metrics results for the turning scenario 2D (YX_S, XZ_S) binary and probabilistic OG maps of the deterministic and the analytical approach. The deterministic approach results match better with the real measurements than the analytical approach because once the ego vehicle turns right, the ego vehicle's yaw angle ψ changes multiple times in one frame and also changes the sensor's origin. Therefore, the changes in the real sensor origins in one frame need to be modeled to obtain the simulated results close to the real measurements. The deterministic approach updates the virtual scene, including sensor origin and orientation, after every $0.0102\,\mathrm{ms}$ in one frame, while on the other hand the analytical approach updates the virtual scene and the sensor origin and orientation every $185.2\,\mathrm{ms}$. That is why the deterministic approach results correlate better with real measurements. So, the analytical approach can only provide realistic results in cases where the ego vehicle orientation changes slowly. Table 6 provides the OCR and the BCC metrics average correlation for all deterministic and analytical approaches frames of the turning scenario.

TABLE 6. **OCR and BCC metrics average correlation for deterministic and analytical approaches frames of the turning scenario.**

Metric	Deterministic appr. YX_S-axis OG map avg. corr.	Analytical appr. YX_S-axis OG map avg. corr.	Deterministic appr. XZ_S-axis OG map avg. corr.	Analytical appr. XZ_S-axis OG map avg. corr.
OCR	88.6%	67.9%	84.8%	63.4%
BCC	82.5%	60.0%	78.6%	54.5%

VI. DISCUSSION

This work introduces a deterministic and an analytical approach to model the motion distortion effect on the LiDAR

sensor performance. The deterministic approach shoots the virtual LiDAR sensor's scan pattern point by point like a real scanning LiDAR sensor, and the virtual scene is updated after each emitted ray. In the analytical approach, the complete scan pattern of the LiDAR model is shot like a flash LiDAR sensor, and the motion distortion effect is modeled on the ideal ray tracing detections directly. In addition, the different reasons that cause the motion distortion effect for the scanning LiDAR sensor are discussed. For instance, the impact of the motion distortion effect increases with the decrease in LiDAR sensor scanning frequency and an increase in the relative velocity between the LiDAR sensor and the target vehicle.

Opposite and turning scenarios were performed at the proving ground in Jtown to validate the motion distortion effect modeling. In addition, a toolchain is introduced to replicate the real-world scenarios in the virtual environment using the GNSS/INS data to validate the motion distortion effect. The motion distortion effect modeling is validated on the point cloud level in two steps. In the first step, three KPIs are defined, including the number of points N_{points}, the distance error $d_{\text{error}S}$, and the mean intensity values I_{mean} obtained from an OOI. Moreover, the 2D (YX_S, XZ_S) binary and probabilistic OG maps are constructed from the synthetic and the real point clouds, and the OCR and the BCC metrics are applied to check the correlation between them frame by frame.

The MAPE of the deterministic approach for the number of points N_{points} received from the OOI in the opposite scenario is 8.9% and 13.0% in the turning scenario. On the other hand, the MAPE of the analytical approach for the number of points N_{points} is 14.3% for the opposite scenario and 22.3% for the turning scenario. The MAPE for the mean intensity values I_{mean} received from the OOI is less than 10% for the deterministic and the analytical approaches for both scenarios. In addition, the MAPE of the distance error $d_{\text{error}S}$ for both simulated approaches and scenarios is less than 0.1%. The mismatch between the simulation and the real measurements exists, although the real-world scenarios are generated using the GNSS/INS data. However, it is still very challenging to replicate the exact orientation, the position, and the velocity of the ego and the target vehicles frame by frame in the simulation. For instance, a difference of $2°$ in the simulated and the real ego vehicle orientation can lead to a mismatch of 10.7% for the XZ_S-axis correlation of the BCC metric and 9.1% for the OCR metric. In addition, it is also quite tough to replicate the exact real OOI material properties in the virtual environment including material roughness, which is why the mean intensity mismatch exists.

The OCR metric average correlation for the deterministic and the analytical approaches is above 84% for the opposite scenario's YX_S-axis and XZ_S-axis OG maps. The BCC metric average correlation for all deterministic and analytical approach frames is above 80% for the opposite scenario's YX_S-axis and XZ_S-axis OG maps. The difference between the results of both approaches and metrics is less than 3% for the YX_S-axis and the XZ_S-axis OG maps of the opposite scenario.

On the other hand, for the deterministic approach, the OCR and the BCC metrics average similarity is above 78% for the YX_S-axis and the XZ_S-axis OG maps of the turning scenario. But for the analytical approach, the OCR metric average similarity is above 63% for the YX_S-axis and the XZ_S-axis OG maps of the turning scenario. However, the BCC metric average correlation is more than 54% for the turning scenario's YX_S-axis and XZ_S-axis OG maps. But, the OCR metric shows a mismatch of 20.7% for the YX_S-axis OG map and 21.4% for the XZ_S-axis OG map between both approaches results. The BCC metric shows a 22.5% mismatch between both approaches' results for the YX_S-axis OG map and 24.1% for the XZ_S-axis OG map.

VII. CONCLUSION

The findings show that the simulation results with a deterministic approach match better with the real measurements than the analytical approach, especially for scenarios in which the yaw rate ψ of the ego vehicle changes rapidly, because the yaw angle changes the sensor's origin and the LiDAR sensor point cloud gets distorted and dilated. Therefore, the changes in the real sensor origins and orientation in one frame need to be modeled to obtain the simulated results close to the real measurements. The deterministic approach updates the virtual scene, including sensor origin and orientation, after every $0.0102\,\text{ms}$ in one frame and is computationally expensive. On the other hand, the analytical approach updates the virtual scene and the sensor origin and the orientation every $185.2\,\text{ms}$ and can compute in real time. If high-fidelity virtual LiDAR data is required, then the user should use the deterministic approach rather than an analytical one.

It can also be concluded from the initial results that the proposed deterministic and analytical approaches can be used to model the motion distortion effect on the performance of automotive LiDAR sensors. Firstly, the demonstrated approach helps to study the effects of LiDAR's scan pattern in different real-world traffic conditions. Secondly, the obtained synthetic data can also be used to test the motion distortion correction algorithms and the LiDAR-based object detection algorithm performance in the early stage of development.

VIII. OUTLOOK

In the next step, we will train the LiDAR-based detector with synthetic data to evaluate the impact of the motion distortion effect on the object list of virtual and real LiDAR sensors. In this context, we also aim for a method to estimate the relative velocity from the motion distortion effect.

ACKNOWLEDGMENT

The authors thank the DIVP consortium for providing technical assistance and equipment for measurements at Jtown proving ground. The authors thank the anonymous reviewers for providing feedback to improve the manuscript.

REFERENCES

[1] I. Bilik, "Comparative Analysis of Radar and Lidar Technologies for Automotive Applications," *IEEE Intelligent Transportation Systems Magazine*, vol. 15, no. 1, 2023, pp. 244-269, doi: 10.1109/MITS.2022.3162886.

[2] A. Mateen et al., "Smart Roads for Autonomous Accident Detection and Warnings", *Sensors*, vol. 22, 2022, pp. 2077, doi: 10.3390/s22062077.

[3] S. Hening, C.A. Ippolito, K.S. Krishnakumar, V. Stepanyan and M. Teodorescu, "3D LiDAR SLAM integration with GPS/INS for UAVs in urban GPS-degraded environments", *AIAA Inf. Syst. AIAA Infotech Aerosp*, 2017, pp. 448-457.

[4] D. Maturana and S. Scherer, "VoxNet: A 3D Convolutional Neural Network for real-time object recognition," *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Hamburg, Germany*, 2015, pp. 922-928, doi: 10.1109/IROS.2015.7353481.

[5] T. Miyasaka, Y. Ohama and Y. Ninomiya, "Ego-motion estimation and moving object tracking using multi-layer LIDAR," *IEEE Intelligent Vehicles Symposium, Xi'an, China,* 2019, pp. 151-156, doi: 10.1109/IVS.2009.5164269.

[6] D. Wang, C. Watkins and H. Xie, "Mems mirrors for LiDAR: A review", *Micromachines*, vol. 11, no. 5, 2020, doi: 10.3390/mi11050456.

[7] R. Roriz, J. Cabral and T. Gomes, "Automotive LiDAR Technology: A Survey," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 7, 2022, pp. 6282-6297 doi: 10.1109/TITS.2021.3086804.

[8] R. Thakur, "Scanning LIDAR in Advanced Driver Assistance Systems and Beyond: Building a road map for next-generation LIDAR technology," *IEEE Consumer Electronics Magazine, vol. 5, no. 3*, 2016, pp. 48-54, doi: 10.1109/MCE.2016.2556878.

[9] B. Rodriguez, X. Zhang, D. Rajan, "Probabilistic Modeling of Motion Blur for Time-of-Flight Sensors," *Sensors*, vol. 22, 2022, pp. 1182, doi: 10.3390/s22031182.

[10] L. Gröll and A. Kapp, "Effect of Fast Motion on Range Images Acquired by Lidar Scanners for Automotive Applications," *IEEE Transactions on Signal Processing*, vol. 55, no. 6, 2007, pp. 2945-2953, doi: 10.1109/TSP.2007.893945.

[11] S. Royo and M. Ballesta-Garcia, "An overview of LiDAR imaging systems for autonomous vehicles", *Appl. Sci.*, vol. 9, no. 19, 2019, pp. 4093, doi: 10.3390/app9194093

[12] InnovizOne, "InnovizOne LiDAR Technology Automotive-Grade Solid-State LiDAR". Available online: https://innoviz.tech/innovizone#top (Accessed on 8 May 2023).

[13] R. W. Wolcott and R. M. Eustice, "Robust LIDAR localization using multiresolution Gaussian mixture maps for autonomous driving", *Int. J. Robot.* Res., vol. 36, no. 3, 2017, pp. 292-319, doi: 10.1177/02783649176965.

[14] A. Haider et al., "Performance Evaluation of MEMS-Based Automotive LiDAR Sensor and Its Simulation Model as per ASTM E3125-17 Standard", *Sensors*, vol. 23, no. 6, 2023, pp.3113, doi: 10.3390/s23063113.

[15] N. Kalra and S. M. Paddock, "Driving to safety: How many miles of driving would it take to demonstrate autonomous vehicle reliability?", *Transp. Res. A Policy Pract.*, vol. 94, 2016, pp. 182-193, doi.org/10.1016/j.tra.2016.09.010

[16] Blickfeld, "Blickfeld Scan Pattern". Available online: https://docs.blickfeld.com/cube/latest/scan_pattern.html (Accessed on 10 Jan. 2023).

[17] A. H. Lang, S. Vora, H. Caesar, L. Zhou, J. Yang and O. Beijbom,"PointPillars: Fast Encoders for Object Detection From Point Clouds," *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA*, 2019, pp. 12689-12697, doi: 10.1109/CVPR.2019.01298.

[18] A. Haider et al., "Development of High-Fidelity Automotive LiDAR Sensor Model with Standardized Interfaces", *Sensors*, vol. 22 no. 19, 2022, pp. 7556, doi: 10.3390/s22197556

[19] JARI, "Automated Driving Test Center (Jtown)", 2022. Available online: https://www.jari.or.jp/en/contract_testing_equipment/facilities_equipment/jtown/ (accessed on January 2023).

[20] W. Yang, Z. Gong, B. Huang and X. Hong, "Lidar With Velocity: Correcting Moving Objects Point Cloud Distortion From Oscillating Scanning Lidars by Fusion With Camera," *IEEE Robotics and Automation Letters*, vol. 7, no. 3, 2022, pp. 8241-8248, doi: 10.1109/LRA.2022.3187506.

[21] F. Petit, "Laser scanner for measuring distances from vehicles", *United States Patent 11,428,818,* 2022.

[22] ISO8855:2011 Road vehicles — Vehicle dynamics and road-holding ability — Vocabulary. *International Organization for Standardization, Geneva, Switzerland*, 2011-12.

[23] P. Ballard and F. Vacherand, "Simulation and understanding of range images acquired in fast motion," *Proceedings of the 1994 IEEE International Conference on Robotics and Automation, San Diego, CA, USA*, 1994, pp. 2242-2247, doi: 10.1109/ROBOT.1994.350952.

[24] S. Ono, H. Kawasaki, K. Hirahara, M. Kagesawa, K. Ikeuchi, "Ego-motion estimation for efficient city modeling by using epipolar plane range image analysis", *Proc. 10th World Congr. Intelligent Transport Systems and Services (ITSWC 2003)*, 2003.

[25] J. Zhang and S. Singh, "LOAM: Lidar odometry and mapping in real-time", *Proc. Robot. Sci. Syst.*, vol. 2, no. 9, 2014.

[26] T. Renzler, M. Stolz, M. Schratter and D. Watzenig, "Increased Accuracy For Fast Moving LiDARS: Correction of Distorted Point Clouds," *2020 IEEE International Instrumentation and Measurement Technology Conference (I2MTC), Dubrovnik, Croatia*, 2020, pp. 1-6, doi: 10.1109/I2MTC43012.2020.9128372.

[27] J. Byun, K.-I. Na, B.-S. Seo and M. Roh, "Drivable Road Detection With 3D Point Clouds Based on the MRF for Intelligent Vehicle", *Cham, Switzerland:Springer*, 2015, pp. 49-60.

[28] P. Merriaux, Y. Dupuis, R. Boutteau, P. Vasseur and X. Savatier, "LiDAR point clouds correction acquired from a moving car based on CAN-bus data", *arXiv:1706.05886*, 2017.

[29] DIVP, "Driving Intelligence Validation Platform". Available online: https://divp.net/ (Accessed on 11 May 2023).

[30] M. McDermott, & J. Rife,"Correcting Motion Distortion for LIDAR HD-Map Localization", *arXiv preprint arXiv:2308.13694*, 2023.

[31] C. Le Gentil, T. Vidal-Calleja and S. Huang, "3D Lidar-IMU Calibration Based on Upsampled Preintegrated Measurements for Motion Distortion Correction," *IEEE International Conference on Robotics and Automation (ICRA), Brisbane, QLD, Australia*, 2018, pp. 2149-2155, doi: 10.1109/ICRA.2018.8460179.

[32] C. L. Gentil, T. Vidal-Calleja and S. Huang,"IN2LAAMA: Inertial Lidar Localization Autocalibration and Mapping," *IEEE Transactions on Robotics, vol. 37, no. 1*, 2021, pp. 275-290, doi: 10.1109/TRO.2020.3018641.

[33] D. Yoon, T. Tang and T. Barfoot,"Mapless Online Detection of Dynamic Objects in 3D Lidar," *16th Conference on Computer and Robot Vision (CRV), Kingston, QC, Canada*, 2019, pp. 113-120, doi: 10.1109/CRV.2019.00023.

[34] J. Shi, W. Wang, X. Li, Y. Yan and E. Yin, "Motion Distortion Elimination for LiDAR-Inertial Odometry Under Rapid Motion Conditions," *IEEE Transactions on Instrumentation and Measurement, vol. 72*, pp. 1-16, 2023, Art no. 9514516, doi: 10.1109/TIM.2023.3328695.

[35] T. Blochwitz, M. Otter, M. Arnold, C. Bausch, H. Elmqvist, A. Junghanns, et al., "The functional mockup interface for tool independent exchange of simulation models", *Proc. 8th Int. Modelica Conf.*, no. 63, 2011, pp. 105-114.

[36] DIVP, Driving Intelligence Validation Platform. Available online: https://divp.net/overview/ (Accessed on 11 May 2023).

[37] Goldstein, H., C. Poole and J. Safko, " Classical Mechanics, 3rd Edition,"*San Francisco: Addison Wesley*, 2022, pp. 142–144.

[38] OXTS, "RT3000 v3". Available online: https://www.oxts.com/products/rt3000-v3/ (Accessed on 18 Sep. 2023).

[39] S. Thrun, W. Burgard and D. Fox,"Probabilistic Robotics", *Cambridge, MA, USA:MIT Press*, 2005.

[40] Blickfeld "Cube 1 Outdoor v1.1", Datasheet, 2022. Available online: https://www.blickfeld.com/wp-content/uploads/2022/10/blickfeld_Datasheet_Cube1-Outdoor_v1.1.pdf (accessed on 10 January 2023).

[41] C. M. Bishop,"Pattern Recognition and Machine Learning", *New York, NY, USA:Springer*, 2006.

[42] P. M. Swamidass, "Mean Absolute Percentage Error (MAPE)", *Boston, MA, USA:Springer*, 2004, pp. 462.

[43] R. Grewe, M. Komar, A. Hohm, S. Lueke and H. Winner, "Evaluation method and results for the accuracy of an automotive occupancy grid," *2012 IEEE International Conference on Vehicular Electronics and Safety (ICVES 2012), Istanbul, Turkey*, 2012, pp. 19-24, doi: 10.1109/ICVES.2012.6294297.

[44] D. S. Moore and G. P. McCabe,"Introduction to the Practice of Statistics", *WH Freeman/Times Books/Henry Holt & Co*, 1989.

[45] A. Schaermann, A. Rauch, N. Hirsenkorn, T. Hanke, R. Rasshofer and E. Biebl, "Validation of vehicle environment sensor models," *2017 IEEE Intelligent Vehicles Symposium (IV), Los Angeles, CA, USA*, 2017, pp. 405-411, doi: 10.1109/IVS.2017.7995752.

ARSALAN HAIDER received his bachelor's degree from the Superior University, Pakistan, in 2016 and a Master's degree from Kempten University of Applied Sciences, Germany, in 2019. He is pursuing his Ph.D. from the Institute for Measurement Systems and Sensor Technology (MST) of the Technical University of Munich in cooperation with the Institute for Driver Assistance and Connected Mobility (IFM) of Kempten University of Applied Sciences. His research interests include physical modeling and simulation of the automotive LiDAR and RADAR sensors for software in-loop testing (SiL).

LUKAS HAAS Lukas Haas received in 2020 his B. Eng. in mechatronics from the Baden-Wuerttemberg Cooperative State University Stuttgart, Germany, and 2022 his M. Eng. in automation and robotics from the University of Applied Sciences Kempten, Germany. He is working on his Ph.D. thesis at the Technical University of Munich, Germany. His present research fields are information extraction from sensor-specific effects and sensor data fusion.

SHOTARO KOYAMA received his master's degree in environmental engineering from the University of Tokoyo in 2016. He is a software engineer of the ADAS system at Hitachi Solutions, Ltd. He is assigned to the Kanagawa Institute of Technology (KAIT) as a project researcher for the DIVP (Driving Intelligence Validation Platform) consortium, is the 2nd phase of the SIP-adus.

LUKAS ELSTER was born in Fulda, Germany, in 1993. He received his B.Sc. and M.Sc. degrees in mechanical engineering from the TU Darmstadt, Germany, in 2016 and 2020, respectively. Since 2020, he has been a research associate at the Institute of Automotive Engineering, TU Darmstadt. His research interests include the effects of radar and lidar sensors in complex surroundings.

MICHAEL H. KÖHLER received the M.Sc. degree in electrical engineering and information technology from the Technical University of Munich (TUM) in 2017 and obtained his Ph.D. on the development of broadband static Fourier transform spectrometers for time-resolved fluid and gas analysis and hyperspectral imaging systems. He is currently a senior optics engineer for LiDAR systems at Blickfeld GmbH.

MICHAEL SCHARDT studied electrical engineering at the Technical University of Munich and obtained his Ph.D. on spectrometers for the mid-infrared region. During his doctorate, he developed spectroscopy solutions suitable for industrial applications. At the moment, Michael Schardt is leading the optical development of lidar sensors at Blickfeld GmbH.

THOMAS ZEH was born in Frankfurt a.M., Germany, in 1965. He studied at UAS Frankfurt and FernUniversität Hagen. He received his Ph.D. from the Technical University of Munich (TUM) in 2004. He has over 25 years of R&D experience in industrial and aerospace engineering. Before he became a full professor at UAS Kempten in 2013, we worked as Head of the Department of Space Science & Exploration at Kayser-Threde GmbH, Munich (today: OHB System AG). His research focuses on automotive sensors for environmental perception, space technology, and electronic circuits and their reliability. He is a founding member of the IFM – Research Institute for Driver Assistance Systems and Connected Mobility, Benningen, and since 2023, speaker of the research group Mobility of UAS Kempten.

HIDEO INOUE received the bachelor's degree in mechanical engineering from Waseda University in 1978. He is currently a Professor with the Kanagawa Institute of Technology and a Visiting Professor with the Tokyo University of Agriculture and Technology, Japan.

MARTIN JAKOBI has been a Research Scientist with the Department of Electrical and Computer Engineering, Institute for Measurement Systems and Sensor Technology, Technical University of Munich since 2003. His research interest includes optomechatronic measuring systems.

ALEXANDER W. KOCH studied at the Technical University of Munich (TUM), Germany. He received the Ph.D. degree from the Universität der Bundeswehr München in 1988. After working as a Research Fellow at the Max Planck Society, he completed his lecturer qualification in electrophysics in 1992. After working as a Professor of Measurement Science at Saarland University, he became a Full Professor of Measurement Systems and Sensor Technology at TUM in 1998. His research interests include optomechatronic measurement technology, laser measurement systems, optical fiber sensors, holographic speckle interferometry, image and sensor data processing, FTIR spectroscopy, environmental monitoring, multisensory systems, and medical measuring systems.