Paper published at the 2020 IEEE International Instrumentation and Measurement Technology Conference (I2MTC) in May 2020

Increased Accuracy For Fast Moving LiDARS: Correction of Distorted Point Clouds

Tobias Renzler
Institute of Automation and Control
Graz University of Technology
A-8010 Graz, Austria
Email: tobias.renzler@tugraz.at

Michael Stolz
Institute of Automation and Control
Graz University of Technology
Virtual Vehicle Research Center
A-8010 Graz, Austria
Email: michael.stolz@tugraz.at

Markus Schratter
Virtual Vehicle Research Center
A-8010 Graz, Austria
Email: markus.schratter@v2c2.at

Daniel Watzenig
Institute of Automation and Control
Graz University of Technology
Virtual Vehicle Research Center
A-8010 Graz, Austria
Email: daniel.watzenig@tugraz.at

***Abstract*—For a long time LiDAR sensors have been used in special purpose applications in robotics to perceive the environment. By the evolution of automated driving to higher levels of automation, LiDAR sensors gain more and more importance also in the automotive domain. Currently, LiDAR is about to become one of the most important sensor technologies enabling automated driving. To limit the emitted power due to safety, state of the art LiDAR sensors scan the environment, which needs time. This scan produces a cloud of points. In this publication the distortion of the LiDAR measurement due to a moving sensor unit is analyzed and a compensation is proposed. That correction, also taking time delays into account, is applied within an autonomous racing application using the LiDAR sensor for localization. Finally, the advantage using the correction is discussed.**

***Index Terms*—Autonomous vehicles, Automated driving, Sensor systems, LiDAR, Distortion correction**

I. INTRODUCTION

LiDAR is an acronym for light detection and ranging. Sensors based on this measurement principle have been used in robotics for longer time [1]. LiDAR sensors are subject to rapid development, driven by automotive industry, due to their high potential as main sensor for environmental perception. Especially for localization and mapping applications, LiDARs are the preferred sensor type [2]. The ability to directly provide 3D information, the long detection range, as well as the independence from ambient light, are main advantages over other sensors used in automated vehicles. But still there is further improvement needed for LiDAR to enter volume market cars. The focus of ongoing development is increasing resolution and robustness. At the same time cost and form factor need to be reduced. Since emitted power is limited due to eye safety within the bandwidths of interest, state of the art is a scanning process that generates the LiDAR measurement. Thereby, the entire field of view, often 360°, is covered by repetitive measurements for increasing angles, stored in a so-called point cloud.

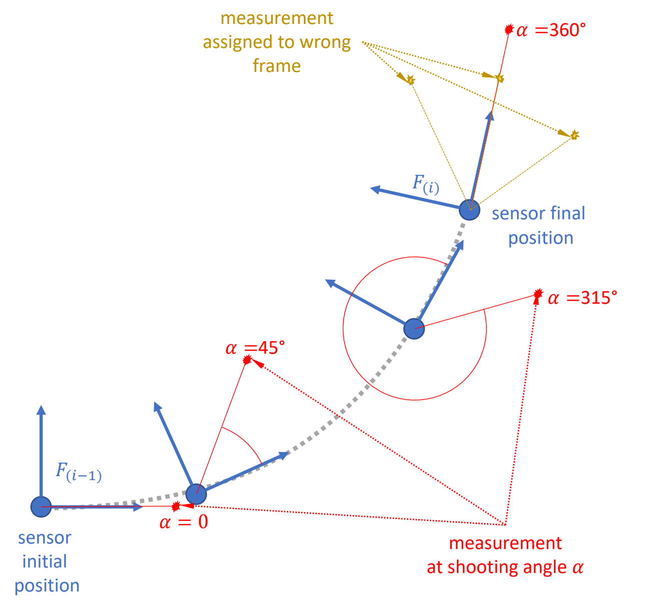

Fig. 1. Moving LiDAR sensor, while scanning. Measurement is referenced with last frame leading to displacement and orientation errors.

If the sensor is moving during the scanning process the point cloud is distorted. This leads to severe disadvantages in performance as is depicted in Fig. 1. First, distortion has a significant impact on the accuracy of maps, when LiDAR data is used for mapping [3]. Second, it increases the difficulty of localization [4] once performed on basis of LiDAR maps, independent of the amount of used LiDARs [5]. Finally, the accuracy of distance measurements used for object detection and further for collision avoidance may be diminished [6]. The correction of this distortion mainly receives attention in the automotive sector: Byun *et al.* [7] perform a correction based on GPS/INS sensor. To do so, before and after every scan the exact position and orientation of the vehicle has to be known. As these measurements rely on infrastructure (satellites), dependent on the environment, accurate enough measurements may be challenging to obtain. A different approach is chosen by the authors in [8]: there, only CAN bus data is used. Although the results look very promising, the approach in [8] contains inconsistencies in (4, 5, 6): they seem to have different angle counting directions and intervals assumed, although using the same symbol α.

Particularly when high performance is required, accurate, reliable, and timely data is needed: for example, in [9] the authors calculate a minimum curvature trajectory to reduce the lap time of a race vehicle. The calculation relies on LiDAR scans that precisely detect the track borders, allowing to estimate the track width and in a further step the racing line. LiDAR localization relying on a particle filter is used in [10] to localize a vehicle at high speed. Caporale *et al*. [11] and Betz *et al*. [12] introduce software architectures to map, localize, and control a fully autonomous racing vehicle, clearly showing that it is valuable to test approaches in racing conditions: there, inaccuracies and small time delays may cause major deviations.

The contribution of this publication is a detailed derivation of motion distortion correction for scanning LiDAR measurements using odometry information in an autonomous driving application. For the above reasons, the same correction procedure can be applied for extrapolating the point cloud to future reference frames. This additionally accounts for known time delays between point cloud measurement and computations based on it. The corrected data allows accurate localization and object detection. Furthermore, it is not bound to infrastructure and thus, environment and scenario independent.

The article is structured as follows: After introducing the LiDAR sensor measurement principle and estimating the lost accuracy due to motion for typical sensors in section II, a correction is derived and proposed in section III. A real-world example is presented in section IV, showing the impact of the proposed approach for an automated driving application. Finally, section V discusses the benefits and summarizes findings.

II. LIDAR DISTORTION

LiDARs that are able to cover a field of view of $360°$, often use a fan of lasers that are positioned vertically with different angles. Within one single measurement, each laser generates one single point containing distance and intensity information. As shown in Fig. 2, all rays are vertically aligned. Therefore, the field of view can be seen as a single vertical scanning line.

To further increase the field of view, the rays are deflected by using a movable (usually rotating) mirror. Due to simplicity and velocity, most LiDARs scan just in one direction (1-D), meaning the mirror can be turned in one dimension only.

After rotating the mirror to a new angle α, again, each laser generates one single point measurement. Often, LiDARs can be configured according to the desired resolution. This influences how many different scanning positions need to be covered during one revolution. Sometimes, also the spinning frequency is adjustable, allowing to target more time critical applications. Measured points from each scanning position, i.e., a vertical set of points belonging to a defined mirror rotation angle, are stored in a single point cloud. After a full revolution the sensor provides the entire point cloud that therefore contains measurements from different time instances. Its origin is the sensor itself. Once the LiDAR sensor is mounted on a moving object (a vehicle), the origin moves during the measurement. This movement during one $360°$ swipe, causes the single measurement points to have different reference locations in space. As this is not taken into account during the creation of the full point cloud at the end of scanning, the data is inconsistent. Due to the different reference locations, measurements from recent scanning positions exhibit less errors than the ones from the start of the scan. Assuming straight line movement only, points in direction of travel seem to be more distant - while points in the opposite direction of travel seem to be closer than they actually are. Once the vehicle's orientation additionally changes (vehicle is turning) points also appear at wrong directions. This effects in sum we name as *LiDAR distortion*. The distortion error is increased with faster movement, meaning higher linear or angular velocity. Assuming the relative velocity between a vehicle and an object is $30\,\mathrm{m\,s^{-1}}$ and a single revolution of a LiDAR takes $100\,\mathrm{ms}$, then the distortion from the first measured angle position to the last is $30\,\mathrm{m\,s^{-1}} \cdot 100\,\mathrm{ms} = 3\,\mathrm{m}$. Objects may change their shape significantly, once parts are detected from rays at the start of the scanning process and other parts are detected at its end. To this day, in the automotive field LiDARS are mostly used in low speed applications, e.g., in inner-city traffic for detection of static and dynamic obstacles. As the effect of distortion increases with movement speed, vehicle ego-motion dependent distortion is usually neglected.

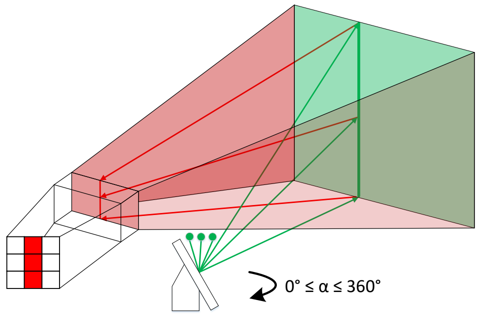

Fig. 2. Working principle of a $360°$ 1-D horizontal scanning LiDAR: laser rays are deflected by a movable mirror (green), reflected by objects (red), and captured with photodetectors. Each ray delivers one point that is added to the point cloud.

III. DISTORTION CORRECTION

As discussed, LiDAR distortion is highly undesirable once point clouds are used for high accuracy applications and/or in connection with safety, e.g., collision avoidance and localization, or for high speed applications, such as autonomous racing.

In the following we state the necessary relations between motion and measurement to later derive a correction of the point cloud for a moving sensor. Constant velocity v and turn rate ω of the sensor during point cloud recording is assumed. For example, these two values can be calculated as the mean from values at time instance before t_{i-1} and after t_i the point cloud recording:

$$\bar{v}_i = \tfrac{v_i + v_{i-1}}{2}, \quad \bar{\omega}_i = \tfrac{\omega_i + \omega_{i-1}}{2}, \quad \Delta t = t_i - t_{i-1} \tag{1}$$

To state a relation between the final global position X_i, Y_i, the orientation Θ_i, and the corresponding values at a shooting angle α, a dimensionless correction-factor

$$c = 1 - \left| \tfrac{\alpha - \alpha_{\text{start}}}{\alpha_{\text{end}} - \alpha_{\text{start}}} \right|, \tag{2}$$

is introduced. Note that the shooting angle is (only) used for defining the *time instance* of shooting using dimensionless parameter c. Therefore, there is no geometric interpretation of the shooting angle done in the correction algorithm. As a result the scanning direction of the LiDAR (clockwise or counterclockwise) does not affect the correction. We assume that $\alpha_{\text{end}} = 2\pi$ and $\alpha_{\text{start}} = 0$ in order to scale the interval of the entire scan to the interval between a specific shooting angle α and the final measurement at t_i. Therefore, the correction factor is zero for the final position ($\alpha = 2\pi$) and one for the initial position of the recording ($\alpha = 0$). For calculating the movement we introduce an angle δ. As shown in Fig. 3, angle δ is *half* of the turning increment of the orientation between frame F_α and frame F_i. It is positive for clockwise and negative for counterclockwise turning. Given that $\Theta_i - \delta = \Theta_\alpha + \delta$ one gets

$$\delta = \tfrac{\Theta_i - \Theta_\alpha}{2} = \tfrac{c\, \bar{\omega}_i\, \Delta t}{2}, \quad \text{where } \delta \in [-\pi, \pi]. \tag{3}$$

The miss-placement of a measurement M due to motion and the required correction is depicted in Fig. 4.

From now on the following notation is used: ${}_A r_B$ denotes a vector r starting in A, pointing to B. The corresponding coordinates of this vector r with respect to a coordinate frame F_C (euclidean right hand) are collected in a column matrix of the same size and denoted with ${}^C_A r_B$.

The corrected column matrix of coordinates of a single measurement point ${}^i_i r_M$ with respect to frame F_i can be calculated as the sum of the displacement of the sensor ${}^i_i r_\alpha$ and a rotation of the measurement ${}^\alpha_\alpha r_M$ to anticipate the change in orientation

$${}^i_i r_M = {}^i_i r_\alpha + {}^i R_\alpha \cdot {}^\alpha_\alpha r_M. \tag{4}$$

Rotation matrix ${}^i R_\alpha$ translates the measurement from frame F_α to frame F_i. This is usually called passive rotation, since the same vector is represented in different coordinate frames. Since a positive δ is defined in direction of positive angles (counterclockwise) from frame F_α to frame F_i, the rotation from frame F_i to frame F_α using ${}^i R_\alpha$ involves rotation about -2δ resulting in:

$${}^i R_\alpha = \begin{bmatrix} \cos(2\delta) & \sin(2\delta) & 0 \\ -\sin(2\delta) & \cos(2\delta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{5}$$

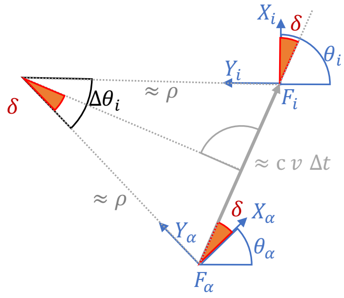

Fig. 3. Movement (of the LiDAR sensor) from a shooting angle position α to the final position i, which is the reference position of the full point cloud. Absolute coordinates are denoted with X and Y. Θ is the orientation. The curvature of the movement $\kappa = \rho^{-1}$, with ρ being the instantaneous radius of the motion.

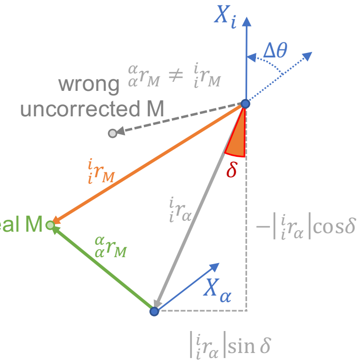

Fig. 4. Wrong placement of measurement M due to movement of sensor. The orientation is changed from pose α to pose i by orientation increment $\Delta\Theta$. The required correction involves a rotation of $\Delta\Theta_{\alpha \to i}$ of the uncorrected measurement and a displacement along ${}^i_i r_\alpha$.

The displacement ${}^i_i r_\alpha$ can be defined by using the vector of negative traveled distance, since frame F_α is reached from frame F_i by *looking to the past*. Assuming the sensor moves along its local x-coordinate and the motion is according to Fig. 4, the displacement correction is $[-s, 0, 0]^T$, with $s = c\, \bar{v}_i\, \Delta t$, rotated by the angle $-\delta$, since movement is along this direction. In this case the rotation is used as active rotation since the frame stays the same and the vector itself gets rotated. The displacement is therefore:

$$\begin{aligned} {}^i_i r_\alpha &= \begin{bmatrix} \cos(-\delta) & -\sin(-\delta) & 0 \\ \sin(-\delta) & \cos(-\delta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} -c\, \bar{v}_i\, \Delta t \\ 0 \\ 0 \end{bmatrix} = \\ &= s \begin{bmatrix} -\cos(\delta) \\ \sin(\delta) \\ 0 \end{bmatrix} \end{aligned} \tag{6}$$

Up to now the traveled distance s was calculated assuming the path directly connects frame F_α and frame F_i by a straight line. In fact, the sensor moves along the arc with the radius ρ (see Fig. 3) instead along the secant, resulting in an increased traveled distance. From Fig. 3 the relation between the length of the arc $s_{\text{arc}} = \rho\, 2\, |\delta|$ and the length of the secant $s_{\text{secant}} = 2\, \rho\, |\sin(\delta)|$ can be derived to

$$s_{\text{secant}} = s_{\text{arc}} \cdot \begin{cases} 1, & \text{if } \delta = 0 \\ \frac{|\sin(\delta)|}{|\delta|}, & \text{otherwise} \end{cases} \tag{7}$$

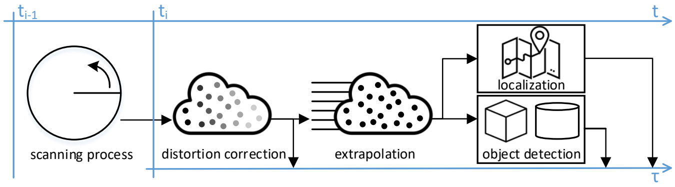

Fig. 5. Providing up-to-date LiDAR data: distortion correction for the duration of the scan and the computation time of the correction itself as well as extrapolation of the point cloud considering application dependent time delays.

The calculation of s for moving along the arc, results in a shorter distance virtually traveled along the secant and the adapted relation for s as

$$s = c\,\bar{v}_i\,\Delta t\,\frac{|\sin(\delta)|}{|\delta|}. \tag{8}$$

Note that the final correction of the full point cloud needs specific angle dependent correction of each shooting angle involving (1) to be calculated once and (2) to (8) for every shooting angle.

Additional compensation of in advance known time delays is tackled separately in the following and in contrast affects all measurement the same. For example, the processing of the point cloud, its correction, sending, receiving, and application dependent operations introduce a time delay τ. Finally, the measurement data at instance $t_i + \tau$ is not up-to-date with the actual position at this instance. As a consequence, the closed-loop performance of control loops using this as an input signal will be decreased. Fig. 5 shows an exemplary pipeline for concurrent localization and object detection introducing an application dependent time delay τ.

This time delay can be anticipated: the point cloud measurement is extrapolated to a frame $F_{i+\tau}$, which is a defined time span τ ahead of frame F_i. Once velocity and turn rate are assumed to be constant and only a single delay needs to be considered, the extrapolation together with the presented distortion correction is performed in a single step. To do so, an adaptation of (1) is sufficient: $\Delta t = t_i - t_{i-1} + \tau$. If in contrast to that, updated values for velocity and turn rate and various delays need to be considered (e.g., different computation time), the extrapolation is performed in an additional step: the approach from above is repeated once for the entire already corrected point cloud. Following the same approach, the delay compensation as shown in Fig. 6 can be formulated as:

$$^{i+\tau}_{i+\tau}r_M = {}^{i+\tau}_{i+\tau}r_i + {}^{i+\tau}R_i \cdot {}^{i}_{i}r_M \tag{9}$$

with

$$^{i+\tau}R_i = \begin{bmatrix} \cos(2\delta_\tau) & \sin(2\delta_\tau) & 0 \\ -\sin(2\delta_\tau) & \cos(2\delta_\tau) & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad \delta_\tau = \frac{\bar{\omega}_i\,\tau}{2} \tag{10}$$

and

$$^{i+\tau}_{i+\tau}r_i = s_\tau \begin{bmatrix} -\cos(\delta_\tau) \\ \sin(\delta_\tau) \\ 0 \end{bmatrix}, \quad s_\tau = \bar{v}_i\,\tau\,\frac{|\sin(\delta_\tau)|}{|\delta_\tau|} \tag{11}$$

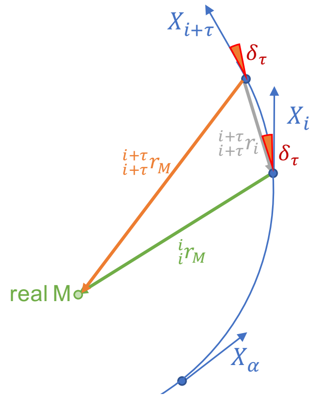

Fig. 6. Anticipation of a time delay τ: extrapolate to future position.

IV. EXPERIMENTS

The effect of LiDAR distortion and its proposed correction, presented in the previous section, is tested during Roborace's SeasonAlpha[1] in Zala-Zone[2]. Based on a LiDAR map, generated while low speed driving, a vehicle localizes itself within this map using current, corrected LiDAR measurements. To do so, we use NDT localization [13] within the automated driving stack *Autoware* [14] on top of Robot Operating System (ROS) [15]. In ROS the correction is executed in a separate node. Since τ (see Fig. 5) delays just the online availability of the measurement, extrapolation is not necessary for offline analysis as discussed in this section.

A. Setup

For the experiment the 360° 1-D scanning OS-1 LiDAR from Ouster is used [16]. It offers 16 vertically aligned rays. Together with a horizontal resolution of 1024 points, a single revolution takes $50\,\mathrm{ms}$. As shown in Fig. 7, the sensor is mounted in a height of $h = 1.1\,\mathrm{m}$ on top of a real race car, provided by Roborace: the DevBot 2.0[1]. Start and end of the scan ($\alpha = 0° = 360°$) are heading in the opposite of the driving direction (see Fig. 8). The scanning direction is clockwise. Note that the OS-1 counts angles in the opposite direction, starting from 360° and ending at 0°. This must be taken into account in (2): one needs to subtract 2π from the shooting angle α. For further explanation, we divide the 360° view in four quarters: Q1, Q2, Q3, and Q4.

During the experiment, the vehicle drives counterclockwise in a circuit containing different types of corners up to 90° (see Fig. 9). The track boundaries are given by traffic cones, that are placed every $5\,\mathrm{m}$. Additionally, on the outside of sharp corners, continuous barriers of the same height are used. The underlying map, that is used as ground truth, is generated before the experiment under slow speed driving, using the same LiDAR sensor. While driving, the vehicle reaches velocities up to $25\,\mathrm{m\,s^{-1}}$ and longitudinal and lateral accelerations up to $10\,\mathrm{m\,s^{-2}}$. A high accuracy on-board measurement system

[1]https://roborace.com/
[2]https://zalazone.hu/en/

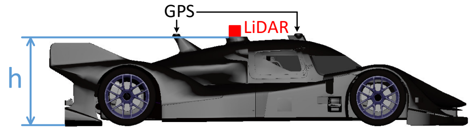

Fig. 7. OS-1 (red) positionated on top ($h = 1.1\,\mathrm{m}$) of DevBot 2.0.

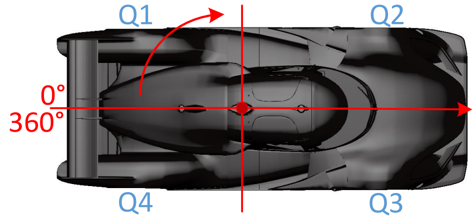

Fig. 8. Bird's-eye view of DevBot 2.0 with rotation direction, starting in Q1 and ending in Q4.

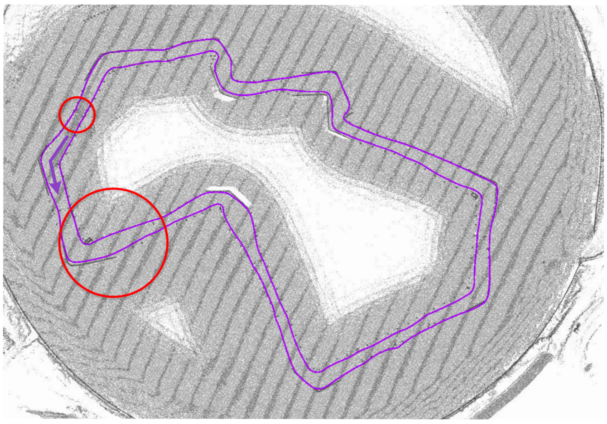

Fig. 9. Race track in Zala-Zone mapped with OS-1 [16]: $730\,\mathrm{m}$ length, cones every $5\,\mathrm{m}$, counterclockwise driving direction. The small and the large red circle identify the start/finish area (Fig. 10) and the corner used in Fig. 11 and Fig. 12.

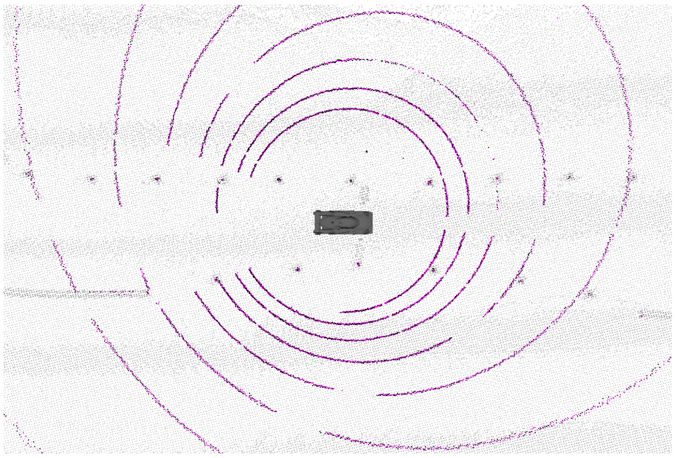

Fig. 10. DevBot 2.0 standing still at the start/finish line showing no distortion: raw point cloud (pink) and corrected point cloud (black) match.

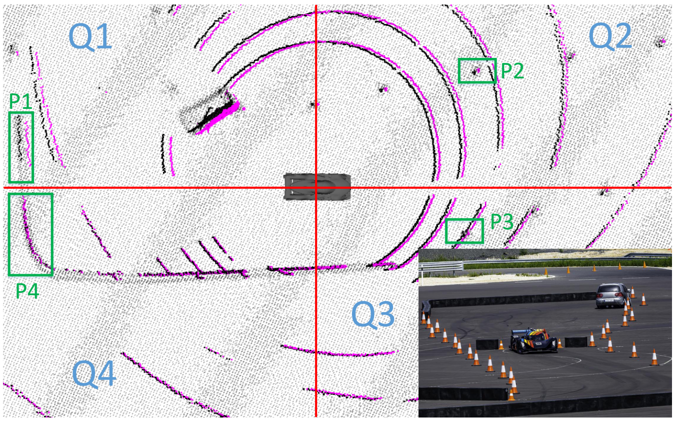

Fig. 11. DevBot 2.0 after a 90° corner shows the effect of distorted points (pink) and their correction (black). The small picture in the lower right corner illustrates this scenario.

provides odometry data (linear velocity and turn rate) every $20\,\mathrm{ms}$, used to correct the LiDAR distortion as explained in the previous section.

B. Results

Fig. 10 shows the racing car at stand still on the start/finish line: the ground truth map is illustrated in transparent gray, while the current measured point cloud is illustrated in pink. The point cloud after applying the presented distortion correction is marked in black. As there is no longitudinal or angular movement, the point clouds superimpose. The circles around the vehicle are reflections from the asphalt. They are not concentric, as the surface is not totally flat. Due to the position of the sensor, there are occlusion areas around the vehicle, caused by the body of the vehicle itself. Further, the two GPS antennas, and the rear spoiler (see Fig. 7) occlude small sections of the surround view in the corresponding direction.

The small picture on the bottom right of Fig. 11 shows the in Fig. 9 highlighted corner of the track the DevBot 2.0 passed. In the background one can see a dummy vehicle that is placed at the inside of the curve, as well as the barriers indicating the outside of the curve. The measurement directly after the corner is illustrated in Fig. 11: at position P1 there is a significant difference between the ground truth map and the sensor data (pink points). Due to the driving direction of the vehicle, the barrier in P1 seems to be closer than it is. For the same reason, in P2 and P3, the cones seem to be further away, although the error decreases as a result of more recent measurements. In P4 the effect of distortion almost vanishes as there the most recent measurements are available. In all cases, the correction is able to determine the real position of the objects. The effect of distortion is not only visible once objects are present but also on behalf of reflections from the floor (circles around the vehicle). The difference between circles from distorted and corrected point clouds decreases more and more from Q1 to Q4.

Fig. 12 shows a zoomed in view on Q1 of the scenario in Fig. 11. For better visualization of the distortion effect, the real borders of the dummy vehicle, based on the ground truth map, are marked in green. Again, without distortion correction, the vehicle seems to be closer than it is. Note that points that

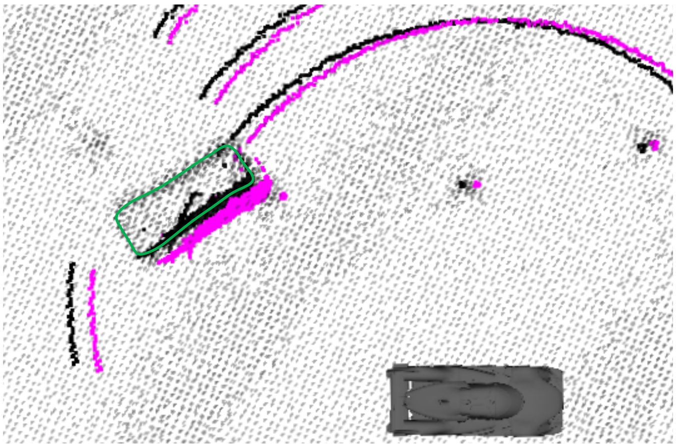

Fig. 12. Outline (green) of the dummy vehicle that is captured by the LiDAR sensor (pink points) in Q1 (see Fig. 11). Ground truth data and current measurement does not match: its correction is marked with black points.

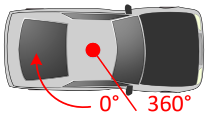

Fig. 13. Proposed sensor alignment once distortion correction is not available.

appear inside the object boundaries are reflections from the floor below the object.

If distortion correction by any reason could not be applied, e.g., due to lack of accurate odometry data, we propose to align the LiDAR sensor in a way such that the influence from the distortion is minimized. Exemplary, for a road vehicle, measurements targeting in driving direction are most critical: objects may suddenly appear and thus, need to be detected as fast and as accurate as possible. Only once this is guaranteed, an appropriate maneuver may be planned and executed in time. Therefore, a sensor alignment as illustrated in Fig. 13 may be appropriate to ensure that the most relevant measurements are taken most recently. Once a sensor scans counterclockwise, the alignment needs to be mirrored accordingly.

V. Conclusion

As LiDAR sensors are integrated in an increasing number of automated driving applications, where safety, or performance are important, the measurement has to be as accurate as possible. As shown in this article, the correction of raw LiDAR data measurement is simple and highly beneficial for moving vehicles already at medium velocities due to the comparably low revolution frequency of 360° scanning LiDARs. The correction assumes constant known velocity and turn rate during measurement and can additionally be used to convert/extrapolate the measurement to future frames in order to sync with localization. For the given example of autonomous racing the approach was successfully applied and results show the applicability. Future application of the correction is planned for enabling increased accuracy for high definition mapping using LiDAR at high driving speeds.

Acknowledgment

The authors kindly acknowledge financial support of the COMET K2 – Competence Centers for Excellent Technologies Programme of the Federal Ministry for Transport, Innovation and Technology (bmvit), the Federal Ministry for Digital, Business and Enterprise (bmdw), the Austrian Research Promotion Agency (FFG), the Province of Styria and the Styrian Business Promotion Agency (SFG), and express their thanks to their partners, AVL List GmbH and to Infineon Technologies Austria AG for financial support. We express our gratitude to Roborace for the opportunity of evaluating our algorithms on their prototype as well as for their support during the testing sessions and the Zala-Zone event in Hungary. Finally, we would like to thank all members of the team *Autonomous Racing Graz* for the hard, time-of-day independent work on and next to the racing track.

References

[1] P. McManamon, *Field Guide to Lidar*, spie field ed., J. E. Greivenkamp, Ed. Bellingham, Wahington USA: SPIE PRESS, 2015. [Online]. Available: http://spie.org

[2] H. Durrant-Whyte and T. Bailey, "Simultaneous localization and mapping: part i," *IEEE robotics & automation magazine*, vol. 13, no. 2, pp. 99–110, 2006.

[3] J. Zhang and S. Singh, "Loam: Lidar odometry and mapping in real-time." in *Robotics: Science and Systems*, vol. 2, 2014, p. 9.

[4] R. W. Wolcott and R. M. Eustice, "Robust lidar localization using multiresolution gaussian mixture maps for autonomous driving," *The International Journal of Robotics Research*, vol. 36, no. 3, pp. 292–319, 2017.

[5] I. Baldwin and P. Newman, "Laser-only road-vehicle localization with dual 2d push-broom lidars and 3d priors," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2012, pp. 2490–2497.

[6] K. Kidono, T. Miyasaka, A. Watanabe, T. Naito, and J. Miura, "Pedestrian recognition using high-definition lidar," in *2011 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2011, pp. 405–410.

[7] 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," in *Field and Service Robotics*. Springer, 2015, pp. 49–60.

[8] 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," *CoRR*, vol. abs/1706.05886, 2017. [Online]. Available: http://arxiv.org/abs/1706.05886

[9] A. Heilmeier, A. Wischnewski, L. Hermansdorfer, J. Betz, M. Lienkamp, and B. Lohmann, "Minimum curvature trajectory planning and control for an autonomous race car," *Vehicle System Dynamics*, vol. 0, no. 0, pp. 1–31, 2019. [Online]. Available: https://doi.org/10.1080/00423114.2019.1631455

[10] T. Stahl, A. Wischnewski, J. Betz, and M. Lienkamp, "Ros-based localization of a race vehicle at high-speed using lidar," in *E3S Web of Conferences*, vol. 95. EDP Sciences, 2019, p. 04002.

[11] D. Caporale, A. Settimi, F. Massa, F. Amerotti, A. Corti, A. Fagiolini, M. Guiggian, A. Bicchi, and L. Pallottino, "Towards the design of robotic drivers for full-scale self-driving racing cars," in *2019 International Conference on Robotics and Automation (ICRA)*, May 2019, pp. 5643–5649.

[12] J. Betz, A. Wischnewski, A. Heilmeier, F. Nobis, T. Stahl, L. Hermansdorfer, and M. Lienkamp, "A software architecture for an autonomous racecar," in *2019 IEEE 89th Vehicular Technology Conference (VTC2019-Spring)*, April 2019, pp. 1–6.

[13] P. Biber and W. Straßer, "The normal distributions transform: A new approach to laser scan matching," in *Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003)(Cat. No. 03CH37453)*, vol. 3. IEEE, 2003, pp. 2743–2748.

[14] N. Akai, L. Y. Morales, T. Yamaguchi, E. Takeuchi, Y. Yoshihara, H. Okuda, T. Suzuki, and Y. Ninomiya, "Autonomous driving based on accurate localization using multilayer lidar and dead reckoning," in *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2017, pp. 1–6.

[15] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "Ros: an open-source robot operating system," in *ICRA workshop on open source software*, vol. 3, no. 3.2. Kobe, Japan, 2009, p. 5.

[16] *OS-1-64/16 High Resolution Imaging Lidar Software User Guide*, Ouster, Inc, 06 2019, rev. 1.12.