Modeling and Understanding LiDAR Data for Absolute and Relative Positioning

Daniela E. Sánchez, *ISTA*
Harvey C. Gómez, *ISTA*
Prof. Thomas Pany, ISTA

BIOGRAPHY (IES)

Daniela E. Sánchez studied Telematics Engineering at Instituto Tecnológico Autónomo de México (ITAM) in Mexico City. She also holds a Master's degree in satellite applications engineering from the Technical University Munich (TUM). She has been a research associate at the Institute of Space Technology and Space Applications (ISTA) since 2017. Her main research area is sensor fusion. Her current research focuses on LiDAR, sensor fusion between LiDAR and GNSS/INS, and relative and absolute navigation algorithms particularly for terrestrial applications.

Harvey C. Gómez Martínez studied Aeronautical Engineering in Universidad de San Buenaventura, in Bogotá, Colombia. He holds a M.Sc. in Aeronautics and Space Technology from Technische Universität München, in Germany, and a M.Sc. in Astronautics and Space Engineering from Cranfield University, in England. He has been a research associate and PhD candidate at the Institute of Space Technology and Space Applications (ISTA) since 2012. His research area is visual relative navigation based on active and passive cameras, and LiDAR, mainly for aerospace applications.

Prof. Thomas Pany is with the Universität der Bundeswehr München at the faculty of aerospace engineering where he teaches satellite navigation. His research includes all aspects of navigation ranging from deep space navigation to new algorithms and assembly code optimization. Currently he focuses on GNSS signal processing for Galileo second generation, GNSS receiver design, and GNSS/INS/LiDAR/camera fusion. To support this activities, he is developing a modular GNSS test bed for advanced navigation research. Previously he worked for IFEN GmbH and IGASPIN GmbH and is the architect of the ipexSR and SX3 software receiver. He has around 200 publications including patents and one monography.

ABSTRACT

This work is composed by two main sections. Firstly, the paper presents how our system, consisting of a GNSS receiver antenna, an IMU and a LiDAR, is used in order to obtain high precision maps through the georeferencing of LiDAR point clouds. An accuracy assessment of the system is conducted, which also gives us insights on the quality of LiDAR range measurements for autonomous driving applications. The assessment is done by georeferencing the obtained point clouds of extracted buildings and comparing them against a supporting measuring system like a total station. The building extraction is done by performing an approximation of the mathematical model of a plane to the facades that composes the building in both, the LiDAR and the supporting measurement system data.

Additionally, the paper also indicates the proposed pose determination method of a mobile agent using LiDAR data. Thanks to the advantages of active, 3D sensors, diverse objects in the environment can be detected as individual point sets, or clusters. Each of segmented objects can be used as a landmark to figure how the agent is located with respect to those structural elements. The algorithm is capable of detecting the clusters in one point cloud, and finding the most alike point set on a subsequent scan. This is achieved by comparing global descriptors for point cloud data. The Ensemble of Shape Functions (ESF) is selected as the cluster descriptor. The cluster matching is performed by comparing the clusters one-to-one, calculating the minimum Chi-squared distance among their descriptors. Smaller this distance, greatest the probability of being the same cluster in distinct epochs. The resultant cluster correspondences for the whole point cloud allow finding the rigid transformation between the point clouds. An initial coarse alignment among the clouds based on the centroids of each matched cluster was performed, followed by a fine alignment in order to reduce errors by the use of the Iterative Closest Point (ICP) algorithm. This approach is valid for urban environments, or for those where many objects can be segmented as clusters.

Finally, a practical case is described in order to show how it is planned to use the outcome of the highly precise georeferenced point clouds and the pose estimation method using LiDAR.

INTRODUCTION

Precise localization for autonomous driving is a highly important topic due to the requirements of the application itself. It brings robustness to the driving assisting functions, aids safety features like collision avoidance and is intended to be used for controlling and improving traffic conditions, among other advantages [1]. The Global Navigation Satellite Systems (GNSS) position solution is universally used for this purpose and although this technology continues under development, it still has the disadvantage of having outages or signal degradation (intentional/unintentional) in certain environments (forests, tunnels, cities or indoor areas). Inertial Navigation Systems (INS) have played an important role to decrease this effect, nevertheless, due to the nature of the sensor, its navigation solution tends to drift after a short time [2].

To correct the measurement errors from the INS, different researcher groups have tried to include information provided by other sensors, e.g. [3], [4], [5]. The most common sensors used are based on feature extraction from visual and LiDAR systems [6]. Although visual systems are more mature and cheaper than LiDAR, they have the disadvantage of being passive sensors, i.e. they extensively rely on the lighting conditions of the environment. Instead, LiDAR, an active sensor, operates independently of the availability of natural or artificial light. Moreover, the fast improvements in LiDAR technology are making it more promising because its initial drawbacks are being solved. For example, LiDAR is switching from being a mechanical device (steering mirrors) to an electronic one (steering electronically the lasers), which in turn will make it smaller, cheaper and more robust.

The scope is to model and understand how LiDAR data can be used for absolute and relative positioning. As for further studies, it is intended to use this outcome to achieve a deep sensor fusion that integrates GNSS, INS and LiDAR measurements in order to provide a precise and robust position for autonomous navigation.

ABSOLUTE POSITIONING

The key aspect to discuss within the current section is to conclude if the accuracy achieved in the implemented direct georeferencing method of the LiDAR point clouds can be used for autonomous driving. The georeferencing is done by a sensor-driven approach, according to the classification in [7], it is also worth to mention that the procedure does not use control points information as suggested in [8]. Moreover, the intention is to use the georeferenced data to compute the absolute position of the LiDAR sensor and give it as an initial state to the relative navigation algorithm to keep computing the absolute position of the car, when the estimated position error provided by the GNSS-INS system exceeds a certain threshold.

For this section, a short description of the sensors and its function is presented. Additionally, the implemented direct georeferencing procedure is explained. Moreover, the reference system used to assess the accuracy of our system is detailed, and to conclude, the results are shown.

GNSS-INS-LiDAR system

The LiDAR selected to be mounted in our vehicle is the Velodyne Puck VLP-16. With this sensor, range measurements up to 100 meters can be achieved. It consists of 16 lasers with a 903 nm wavelength, separated vertically by 2°. The vertical FOV is 30° and the horizontal FOV is 360° as shown in Figure 1[left].

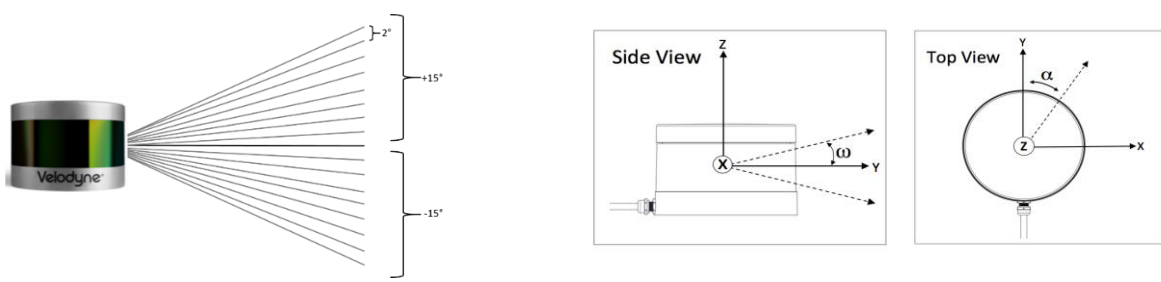

Figure 1. Velodyne Puck VLP-16 channel configuration scheme [left] and its spherical coordinate frame [right] (courtesy of Velodyne LiDAR, Inc.)

For each laser point an azimuth angle, α, and an elevation angle, ω, are decoded as well as the range measurement, R. A detailed scheme is shown in Figure 1 [right]. Those extracted values are used to compute the coordinates of the laser point in the sensor's local reference frame by following the given equations:

$$\begin{aligned} x &= R * \cos(\omega) * \sin(\alpha) \\ y &= R * \cos(\omega) * \cos(\alpha) \\ z &= R * \sin(\omega) \end{aligned} \tag{1}$$

The interface box of the LiDAR is connected to a patch antenna to use the GPS timestamp extracted from the NMEA message and it is used to synchronize the LiDAR measurements with other sensors. In order to give an absolute position and orientation to the point clouds, a GNSS antenna receiver and a MEMS IMU are installed as well on the top of the vehicle (See Figure 2 [right]). The sensors used for this purpose are a Trimble R10 and an Xsens MTi-G-710, respectively. The IMU had connected, like the LiDAR, a patch antenna for time synchronization. A scheme of the interface setup between the sensors can be seen in Figure 2[left].

The data obtained from the GNSS antenna and the IMU is introduced to a post-processing tool that allows their integration. The IMU supports the navigation solution when there are signal outages. Having the integration of the IMU will allow us to compare in our future work the navigation solution of a) GNSS+INS versus b) only GNSS plus the LiDAR correction. Correction data from a close reference station was obtained, the baseline is approximately 9.2 km. The tool uses a loosely coupled integration architecture and the overall solution can achieve an accuracy of 2-3 cm. Nevertheless one must consider that it highly depends on certain factors like signal obstructions, multipath, satellite geometry, atmospheric conditions, among others.

Figure 2. Scheme of the interface setup [left] and the mounting of the sensors [right]

Direct Georeferencing of LiDAR point clouds

Direct georeferencing through the use of GNSS-INS systems has been very popular and has been used greatly in airborne surveys [9], as it has demonstrated to achieve a good level of accuracy using a minimum of ground control [10] [11]. Especially for the mobile mapping industry, this method has opened the opportunity to access a wide variety of uses of the data. Moreover, this method has not only been used with the traditional airborne systems but as well in other mobile agents like cars or drones. The attention in combining direct georeferencing with LiDAR measurements has increased due to the benefits that technology brings over the traditionally used camera.

As described in the previous section the sensors are mounted on plates on top of the car. The assembly is carried out in such a way that the local frames of the IMU and the LiDAR follow the X-Y-Z axis definition of the vehicle frame, which means that the Y-axis points to the driving direction, X-axis points to the right side of the car and Z-axis points up.

Table 1. Lever arm offset

IMU to GNSS antenna (Xsens center to the L1 Phase center)	x = -0.180 m y = -0.058 m z = 0.165 m
LiDAR to GNSS antenna (VLP-16 mounting point to L1 phase center)	x = -0.530 m y = -0.058 m z = 0.178 m

All the laser points that constitute the point cloud are referred to the optical center of the LiDAR, where the origin of its body frame is defined. In order to transform the measurements from the LiDAR body frame to a global reference frame, first one has to compensate the difference between the position of the L1 phase center of the GNSS antenna and the optical center of the LiDAR, where the measurements are referred to. During the post-processing, the tool that integrates the information of the IMU and the GNSS receiver, estimates the lever arm, LA, and it is consistent with the one measured and shown in Table 1.

$$DP_{VLP} = DP_{VLP} - LA + OC \tag{2}$$

The LiDAR and the IMU are rigidly mounted on the same plate (next to each other) to avoid any type of distortion due to stress. After the above translation, the following rotation has to be applied to give the correct orientation to the laser points.

$$R = R_z(-yaw) * R_x(pitch) * R_y(roll)$$
$$DP_n = R * DP_{VLP} \tag{3}$$

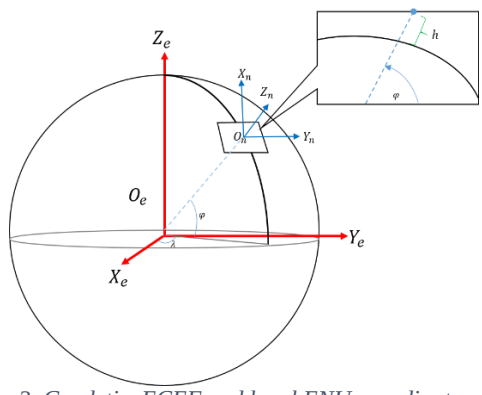

Figure 3. Geodetic, ECEF and local ENU coordinate systems

After obtaining DP_n, which is the laser point in the ENU coordinate system shown in blue on Figure 3, one can transform the coordinates to ECEF and then to any other mapping reference frame by

$$P_e = (R_{e/n} * DP_n) + P_{e,ref}, \tag{4}$$

where $R_{e/n}$ is the rotation matrix from local ENU to ECEF and $P_{e,ref}$ is the position in ECEF coordinates of the origin of the local ENU frame. The value given to $P_{e,ref}$ is the position we get from the GNSS-INS integration process, which has to be transformed to the ECEF coordinate system.

Finally P_e is transformed to WGS84 $P_{WGS84} = \begin{pmatrix} \lambda \\ \varphi \\ h \end{pmatrix}$ according to [12].

Figure 4 and Figure 5 show the results of the procedure. In the first image, just 3 different scans from the LiDAR are shown and superposed on Google Earth. On the second image, one can see more scans plotted that correspond to different epochs along the same trajectory followed by the car. Each scan is plotted with a different color and in a light blue square, one can also see the position of the car w.r.t. the GNSS receiver antenna at the moment in which the data is georeferenced.

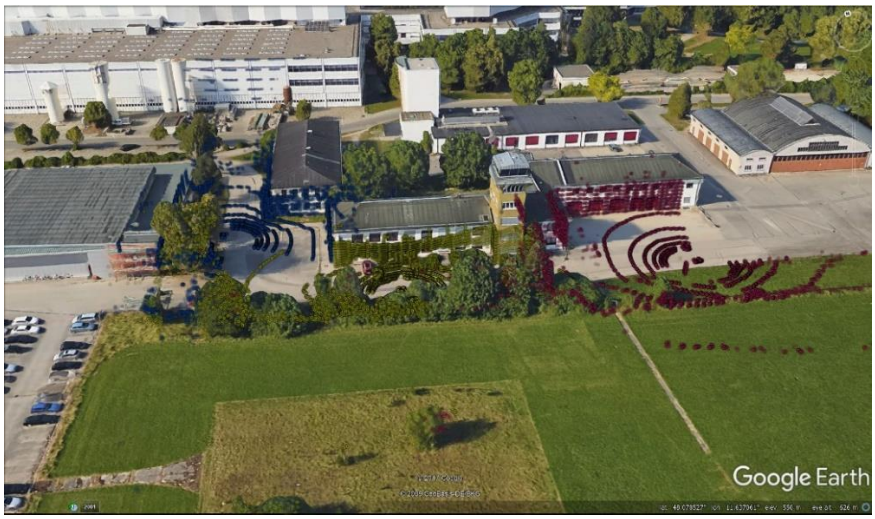

Figure 4. Rendering of 3 georeferenced LiDAR point clouds on Google Earth

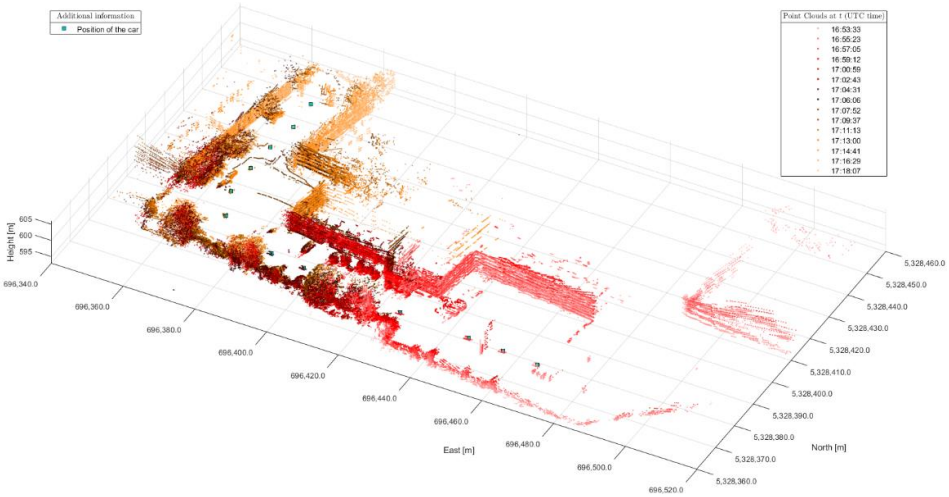

Figure 5. Direct-geo-referencing of LiDAR data at different times

Accuracy Assessment

To assess how accurate the system performs, including the LiDAR in terms of range accuracy, a supporting measurement system is used. This is done by acquiring data with our system in a measurement campaign and processing it according to the explanation given in the previous section. The georeferenced data is compared with the data coming from the reference system allowing us to conclude the overall performance of the system and thus the LiDAR measurements' accuracy.

According to [7], the most accurate and conventional method for the georeferencing of terrestrial laser scanning data is using a total station. Therefore, the Multistation MS60 is used as a reference system. The MS60 provides precise long-range scanning with range accuracy of 2mm + 2ppm when the target is between 0m to 500m.

Figure 6 shows the place in which the measurements with our system were done. The measurement campaign consisted of the following steps: 1) 15 minutes waiting time for the GNSS receiver antenna to solve ambiguities and get a fixed position; 2) 8-shapes trajectories for the IMU kinematic calibration (used in the forward processing of the GNSS-INS integration); 3) two

rounds driving with the car around the testing area for the sensors' data collection; 4) 8-shape trajectories for IMU kinematic calibration (used in the reverse processing of the GNSS-INS integration). The trajectory is depicted with green points in Figure 6. From that trajectory, 3 epochs were chosen to compare the georeferenced LiDAR data with the reference data.

The comparison consisted of choosing the same target in both datasets (the building marked with the red ellipse in Figure 6) and computing the difference in translation and rotation w.r.t. the GNSS antenna. In Figure 7 can be seen how, after the measurement campaign with our system, the MS60 was set up. It is worth to mention that the data coming from the MS60 is obtained in the UTM coordinate system. This is due to the setup performed before scanning the target with the Multistation, in which the absolute position of the MS60 (measured with the R10) is given to the instrument as well as two back sights measurements (obtained with the R10 as well) so that the MS60 can compute the right position and orientation of the scan data. This pre-step allow us to have the data in a global coordinate system instead of the instrument's local coordinate system.

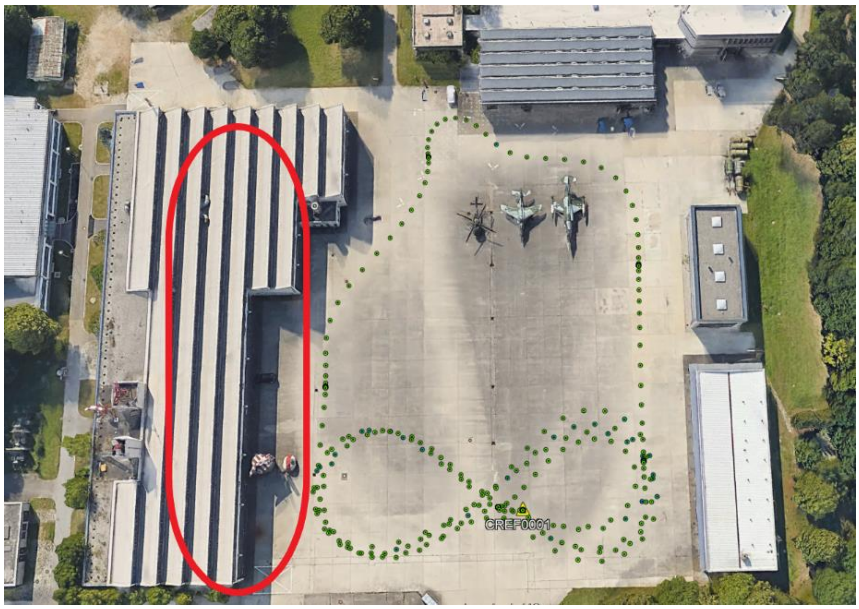

Figure 6. The testing area in the facilities of the University of the German Federal Armed Forces (UniBwM).
The red ellipse points the chosen target to extract from the LiDAR point clouds. The green dots show the trajectory done with the car.

Figure 7. Scanning of the target with the reference system (MS60)

The data coming from the Multistation is dense and equally distributed along the façade of the targeted building. The sensor is capable of retrieving structural ornaments of the façade or other elements like windows, frames or even surface reliefs of the target. On the other hand, the lasers of the LiDAR VLP16 may not intersect the whole target at certain epochs, the scan data follows the geometry of the sensor and it is not capable to retrieve millimeter-level details of the façade. Therefore, basic shapes like planes and cylinders were extracted from both sets of data in order to compare them. In the case of the MS60, when trying to find the plane that best fits the data corresponding to the wall, a maximum distance of 1cm from an inlier point to the plane is allowed. For the LiDAR VLP16, the maximum allowed distance is set to 10 cm. The restriction tries to ensure that most of the data points belong to the wall rather than any other structural ornaments.

The same criteria and values were used for the cylinder extraction. After the data is filtered and only the points corresponding to the plane that best describes the wall of the building are obtained (see Figure 8), Iterative Closest Point (ICP) algorithm [13] is used in order to obtain the affine transformation between both point clouds.

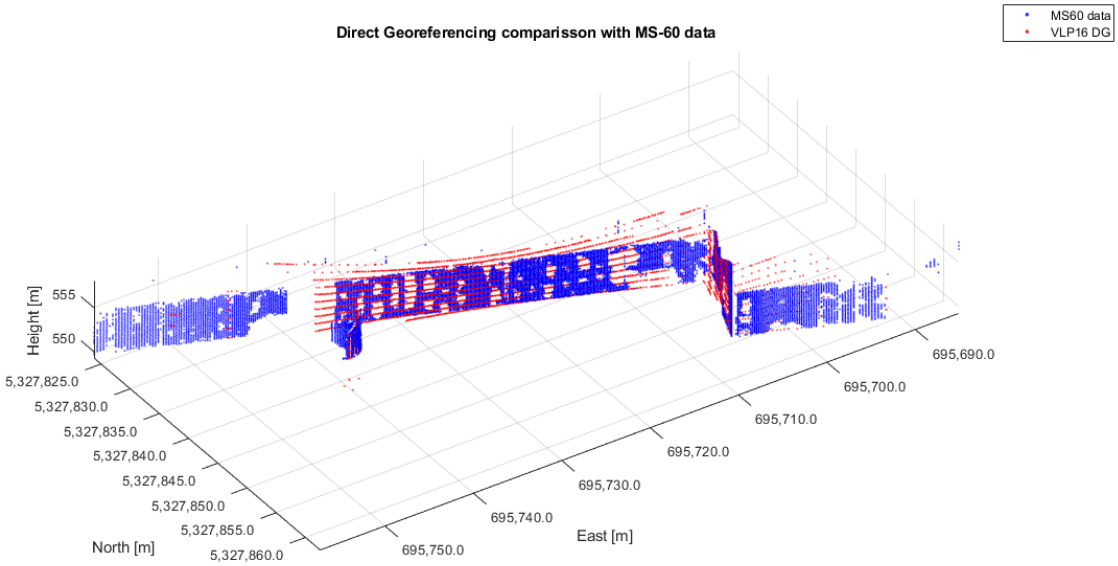

Figure 8. Comparison between VLP-16 and MS60 (filtered point clouds)

In Table 2 is presented the obtained results. As said before, different sets of georeferenced data at different epochs are examined. In Figure 9 is shown the approximate position in which the evaluated data was acquired. For P1 and P2 the results are quite similar, the errors in North and East do not exceed the decimeter level. The difference seen in the results could come from the position accuracy of the GNSS-INS solution which does not exceed 2cm along the whole trajectory or it could be as well errors in the range measurements of the VLP16, which should be in the range of 2cm-3cm. For the attitude, roll and pitch are in the range of the expected values according to the specifications of the sensor in dynamic scenarios (Typ|Max: 0.3°|1.0°); the heading value is the one showing a greater difference as a typical variation of 1° should be expected.

The acquired data in P3 did not allow a fair comparison with the reference due to the perspective in which it was taken. The cylindrical structure marked with a pink square, as it is shown in Figure 9, occlude the view of the LiDAR, meaning that not many measurements were taken from the remaining part of the wall and therefore resulted in a poor plane description of the wall. This is reflected in the results shown Table 2 in for P3.

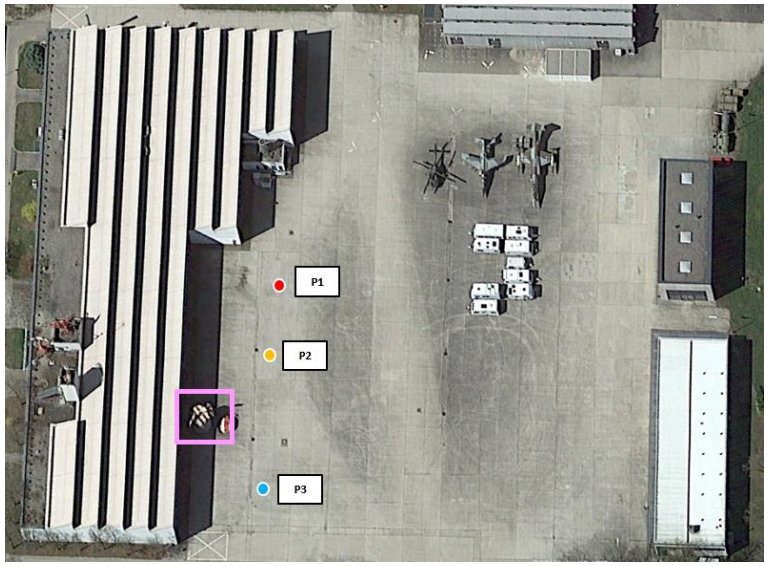

Figure 9. The approximate position of the LiDAR data acquisition along the car trajectory (P1, P2, and P3)

Table 2. Error in position and attitude of our system according to the accuracy assessment

Perspective	East error [m]	North error [m]	Roll [deg]	Pitch [deg]	Yaw [deg]
P1	-0.0629	0.0029	-0.4762	0.5133	-1.1443
P2	0.0244	-0.0524	-0.7519	0.0129	-1.7418
P3	-0.5521	0.2256	2.1439	2.0584	-1.4359

RELATIVE POSITIONING

For the second part of our work, a method for the position estimation of a mobile agent (a car) is presented. The method is divided into the following steps: 1) preprocessing and object segmentation; 2) object recognition and association and 3) point cloud registration and fine alignment

Point cloud preprocessing and object segmentation

The LiDAR is used to capture the surface of the surrounding objects. The agent (e.g. a car) moves, and the sensor records the environment. Data should be cleaned for a better outcome. Elimination of outliers is required to enhance the integrity of the navigation pipeline. For each point, the mean distance to the nearest neighbors is computed [14]. Those neighbor points whose mean distances are outside a general distance mean threshold and standard deviation threshold are removed from the point cloud. Some deliberate data discard is necessary for sake of better quality at the results, i.e. very far points may provide useless information. The ground could also be eliminated, under the assumption it is even, and it provides no advantages for the application. A range filter, based on considered distance thresholds in each sensor axis, encloses the most useful points.

It is possible to subdivide the scene scanned into smaller sets, called clusters. Each cluster is assumed to be one object in the surroundings. Different methods are used for clustering point clouds [15]. In this approach, the Euclidean segmentation method [14] is employed, where the point cloud is subdivided into smaller sets based on the Euclidean distance among the points, under a distance threshold. A cluster is composed by all the points whose distance to their closest neighbors is lower than the limit. In Figure 10, the result of clusters segmentation is shown.

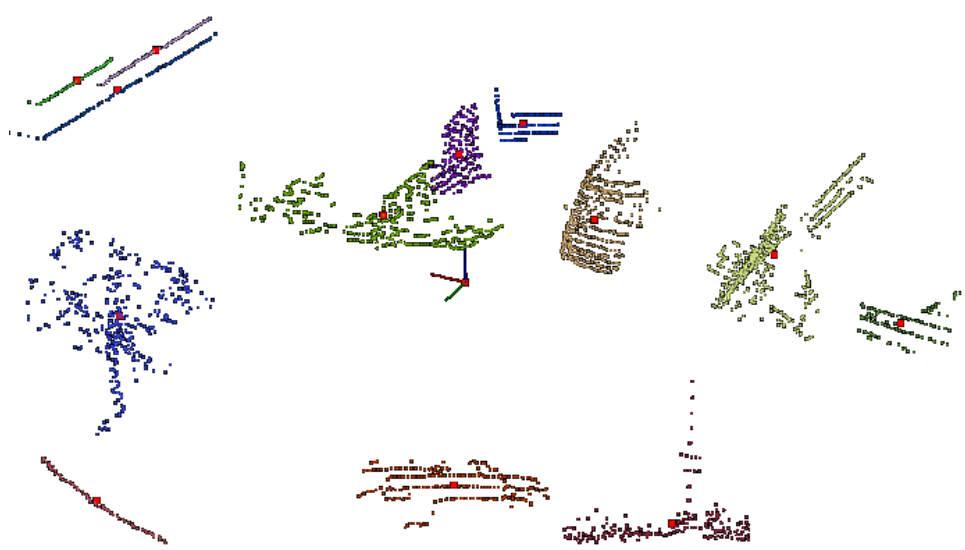

Figure 10. Computed centroid of each cluster

Object recognition and association

Global descriptors [16] play a big role in the object recognition task. In this research, the Ensemble of Shape Functions [17] is a global descriptor that does not require the normal vectors estimation of the surfaces sensed, improving the processing speed. After the cluster segmentation at time $t = k$, the ESF descriptor is calculated for each of the clusters. The descriptors are stored

for further matching against the signatures of the point cloud clusters obtained at time $t = k + 1$. In order to compare the likeliness of two descriptors, the Chi-square (χ^2) distance is calculated among all the stored signatures using the relationship

$$\chi^2 = \frac{1}{2}\sum_{i=1}^{640}\frac{(x_i - y_i)^2}{x_i + y_i} \tag{5}$$

Those descriptors, whose distance is smaller than a given threshold, are considered as nearest neighbors. That is, both descriptors belong to the same cluster calculated for different epochs. The Chi-square distance is calculated [18] by means of brute force: each cluster descriptor from the point cloud measured at $t = k + 1$ is compared against all the stored cluster descriptors from the point cloud recorded at $t = k$. Those distances greater than the threshold produce unmatched descriptors, which are discarded from both clusters lists. Two matched ESF descriptors are shown in Figure 11, as nearest neighbors using the calculation of the Chi-square distance.

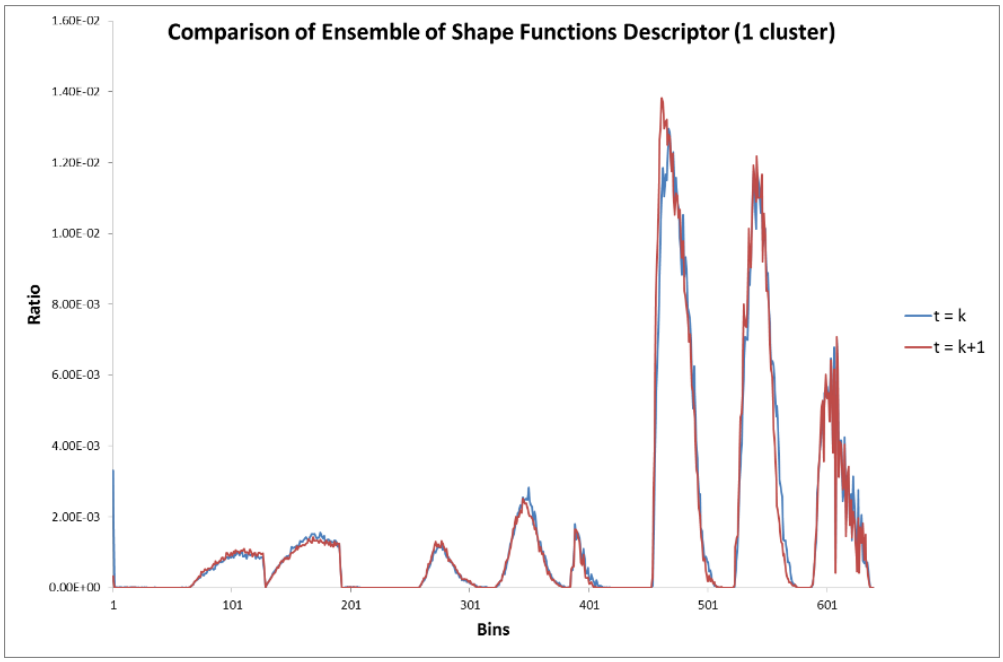

Figure 11.Matching of two ESF descriptors from the same cluster in different epochs.

Under the certainty of having a cluster identified in two consecutive point data sets, the geometric association is required in order to link the same object in both scans. The centroid of a point set is just the average value for the 3D coordinates of all the points, by means of

$$\bar{P} = \frac{1}{N}\sum_{i=1}^{N} P_i \tag{6}$$

With the centroids, each coupled signature leads to the correspondent cluster association, as shown in Figure 12.

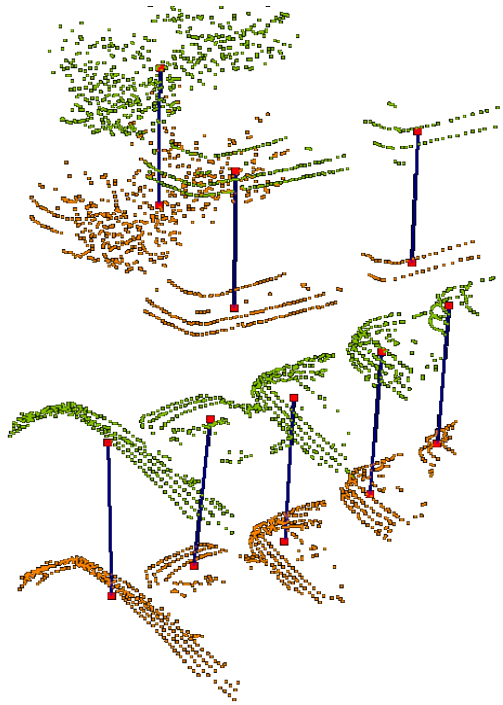

Figure 12. Association of clusters centroids based on descriptor matching. Orange cloud recorded at t=k, green cloud recorded at t=k+1.

Point cloud registration and fine alignment

It is intended to overlap the consecutive scans in order to perform the most accurate possible alignment between them, finding the correspondent transformation matrix that allows such positioning. The centroids association gives the initial step for the registration procedure. Singular Value Decomposition (SVD) [19] is useful in point cloud processing in order to align two point clouds, finding the rotation matrix between them. For this particular case, two new point clouds, P' and Q', are composed by the correspondent centroids found in P and Q, respectively. The centroid of each new cloud is calculated, and a covariance matrix is formed using the clusters' centroids, as follows

$$C = \frac{1}{N}\sum_{i=1}^{N}\left(P'_i - \bar{P'}\right)\left(Q'_i - \bar{Q'}\right) \tag{7}$$

The SVD procedure is applied then to find 3 matrices that multiply each other:

$$C = USV^T \tag{8}$$

The purpose is to find the matrices U and V, because the rotation matrix between the two point clouds is defined by the product

$$R_{\mathrm{SVD}} = VU^T \tag{9}$$

Using the resultant matrix, the translation between the clouds is

$$t_{\mathrm{SVD}} = \bar{Q'} - R_{\mathrm{SVD}}\bar{P'} \tag{10}$$

The obtained rotation matrix and translation vector are then employed along with the original cloud Q at $t = k + 1$, to make an initial coarse alignment between this and P:

$$Q_{\mathrm{IA}} = R_{\mathrm{SVD}}Q + \mathrm{t}_{\mathrm{SVD}} \tag{11}$$

The previous process helps to reduce the distance and rotation between the point clouds, although the overlapping is far from optimal. Fine alignment aids to reduce this error, making the registration more accurate. The employment of the ICP algorithm [13] is the crucial method to obtain this enhancement. ICP reduces the rotation and translation values in a given number of iterations until the error is an admissible value for the correspondent application. When any of the boundary conditions are met, the algorithm should provide the rotation matrix, R_{ICP}, and translation vector, $\mathrm{t}_{\mathrm{ICP}}$, matching the two point clouds as best as possible

$$Q_{\mathrm{FA}} = R_{\mathrm{ICP}} Q_{\mathrm{IA}} + \mathrm{t}_{\mathrm{ICP}} \tag{12}$$

Using Equations 11 and 12, the final transformation matrix can be written in homogeneous coordinates as

$$T = \begin{bmatrix} R_{\mathrm{ICP}} R_{\mathrm{SVD}} & R_{\mathrm{ICP}} \mathrm{t}_{\mathrm{SVD}} + \mathrm{t}_{\mathrm{ICP}} \\ 0 & 1 \end{bmatrix} \tag{13}$$

And the result of the registration process can be seen in Figure 13.

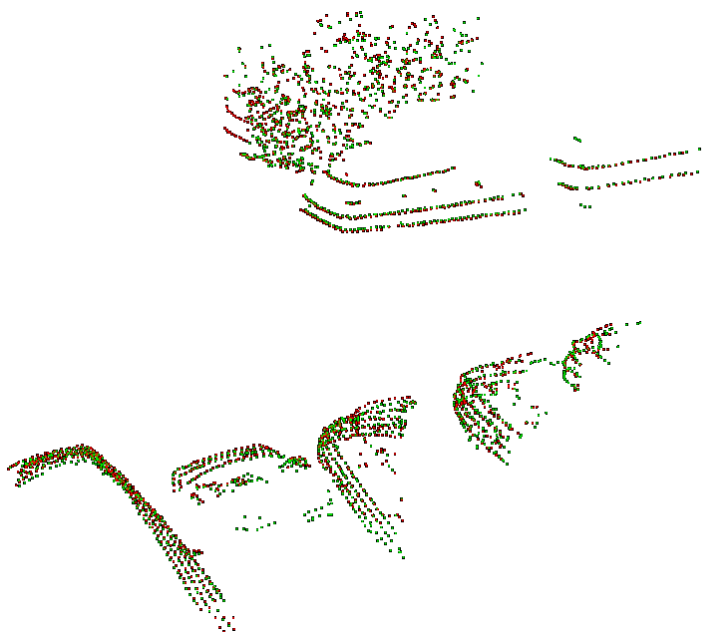

Figure 13. Point clouds final registration after coarse and fine alignments

Detection of cylindrical objects

Detecting the fixed objects in the environment provides better ancillary elements. These objects do not change their position, in comparison to cars, motorcycles, pedestrians, bikes, among others. This object detection is usually performed using the whole point cloud, taking a lot of time for the processing. Trees and poles are the desired objects to be detected, due to their cylindrical shape, and the symmetry axis is generally vertical, almost matching the Z-axis of the sensor. The method used for cylinder detection is known as RANdom SAmple Consensus – RANSAC [20], although it can also detect planes, spheres, and other 3D geometric primitives. The drawback of this algorithm is the requirement of the normal vectors estimation of the sensed surface because it evaluates the orientation of those vectors to determine if the intended geometrical model can be fitted on the point set.

The clusters as independent clouds could reduce the processing time because the number of points per cluster is much smaller than those of the entire point cloud. The RANSAC algorithm is applied for each cluster, giving as a result if a cylindrical model could be fitted to the data set or not, i.e. it detected a tree/pole. Figure 14 illustrates the detection outcome for some trees in the surroundings.

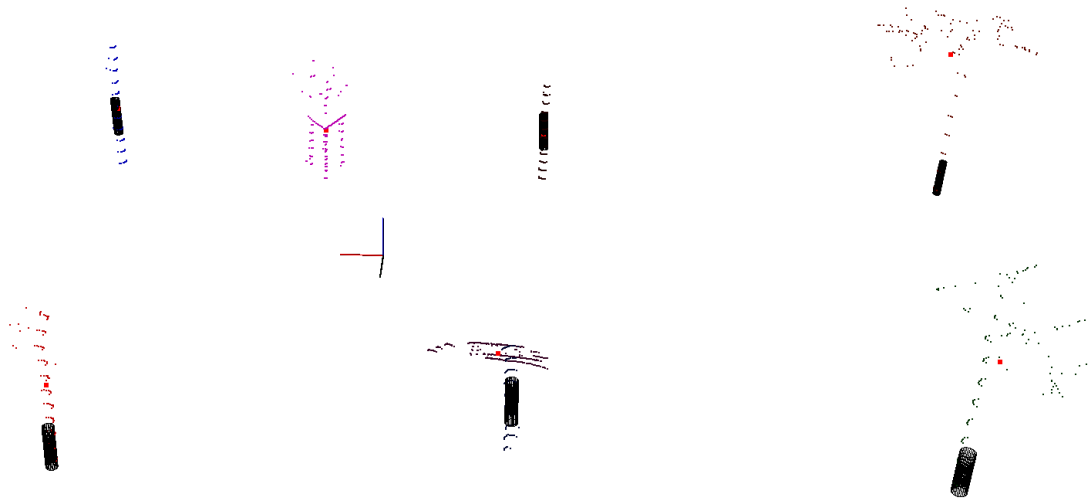

Figure 14. Detected cylinders from individual clusters. The red axis of the sensor reference frame is aligned with the car's longitudinal axis.

Experiment results

The main purpose of this workflow is to obtain the displacement vector of the agent based on solely LiDAR measurements. The multiplication of the initial position of the sensor with the final transformation matrix shown in Equation 13 provides the estimated final position. The expected result is the generation of the agent's trajectory based on consecutive scans from the LiDAR sensor. Figure 15 illustrates the sensor displacement after the processing of two consecutive point clouds. However, bad cluster matching may provoke poor results in the final position estimation of the sensor. Additionally, a registration drift exists when ca. 15-20 frames has been continuously processed, producing a position error in the vertical orientation as can be seen in Figure 16. The reasons for this drift may be caused for the absence of the ground, filtered in the first stages of the procedure and used to enhance the algorithm's processing speed. The absence of ancillary points in the z-axis of the LiDAR may produce this error accumulation.

Figure 15. Point Clouds registration and LiDAR sensor displacement between consecutive measurements. Arrow's tail is the initial position, and the head is the final position.

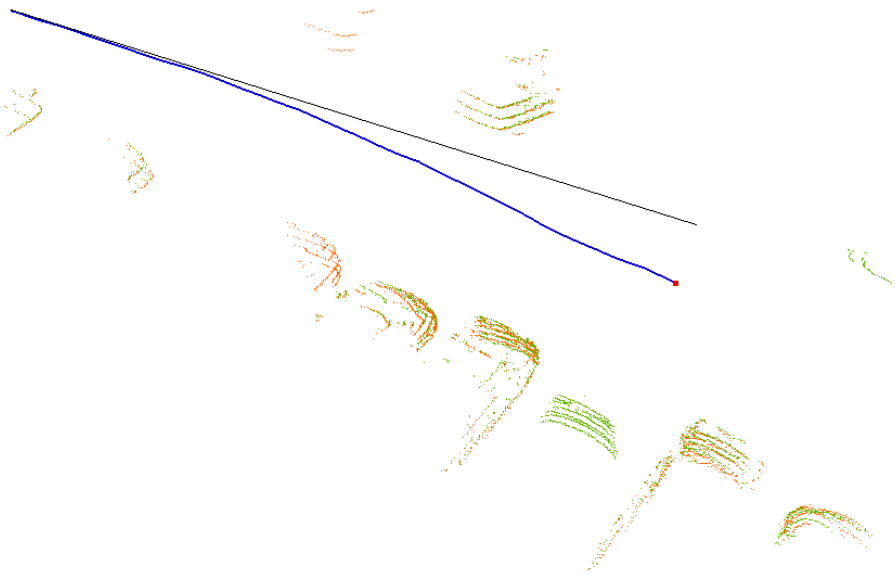

Figure 16. The vertical drift of LiDAR position after ca. 15-20 frames. The black line is the nominal path and the blue line is the actual path.

On the other hand, the cylindrical objects detection is behaving as planned, although the variation in the estimation on the normal vectors may affect the procedure from time to time. For those identified objects, a point in the symmetry axis is recorded in order to provide the position of the cylinder's centerline in the sensor reference frame. Adding the measurements of the [GNSS/IMU], the georeferencing of the detected objects allows declaring their position in world coordinates, making possible to construct a map meanwhile the position of the agent is also estimated, being the final objective of this method.

PRACTICAL CASE

As said before, it is of our interest to detect fixed structural elements of the environment that the LiDAR is sensing. Fixed objects in the environment provide ancillary elements that can be trusted to compute the displacement vector as they do not change of position as compared with cars, pedestrians, bikes among others. On the other hand, with the first part of the work described in the paper, we are able to georeference accurately point clouds as long as the GNSS-INS system is not in a challenging environment. When the GNSS-INS navigation solution cannot provide the level of accuracy that the application needs due to the different adversities that technology has to face, the position estimation algorithm (using only LiDAR) is triggered to compute the displacement vector between one scan and another. At the same time, the displacement vector can be used to correct the last trusted position of the car. Although it has been shown in the previous section that a vertical drift has been founded through the LiDAR processing, the error can be minimized with the addition of ground detection. The proposed method appears to be a good supporting system for GNSS denied environments (especially suited for urban environments) as it is very likely to find fixed objects that can be segmented in the point clouds.

The example below show images of a measurement campaign done in an urban area. One can see the detection of cylindrical objects and its given absolute position at different epochs (see the left side of Figure 17, Figure 18 and Figure 19) superposed in Google Earth. The corresponding scan from the LiDAR measurements shows how those elements were seen by the sensor (see right side of Figure 17, Figure 18 and Figure 19). As the car continued its trajectory, some elements that appeared in the scan of a certain epoch are seen in the next epoch due to the sensor is still able to get measurements of it. For example, in Figure 18 and Figure 19 is marked with a red circle, elements that must belong to the same object but that have been slightly given a different position.

Further investigation of the errors in the estimated position after a vertical drift correction is needed in order to conclude the quality of the solution.

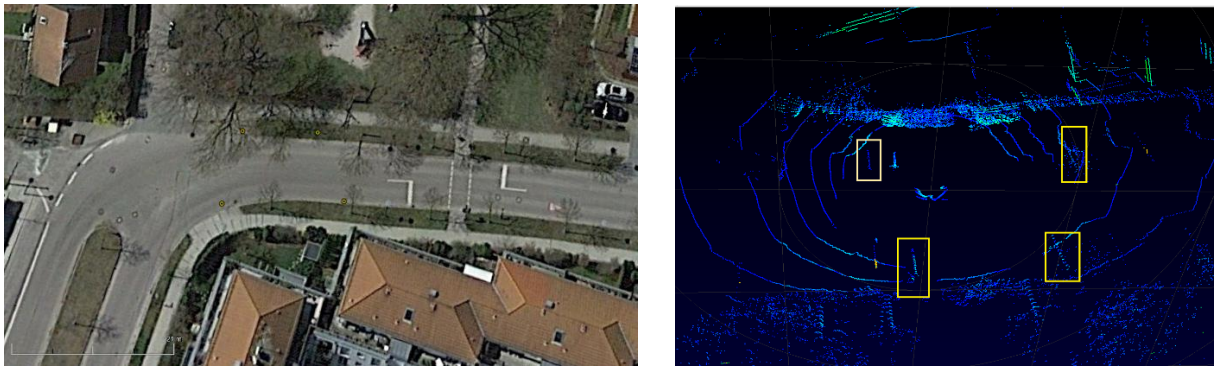

Figure 17. Georeferencing of cylinder's centroid (Frame 1787).

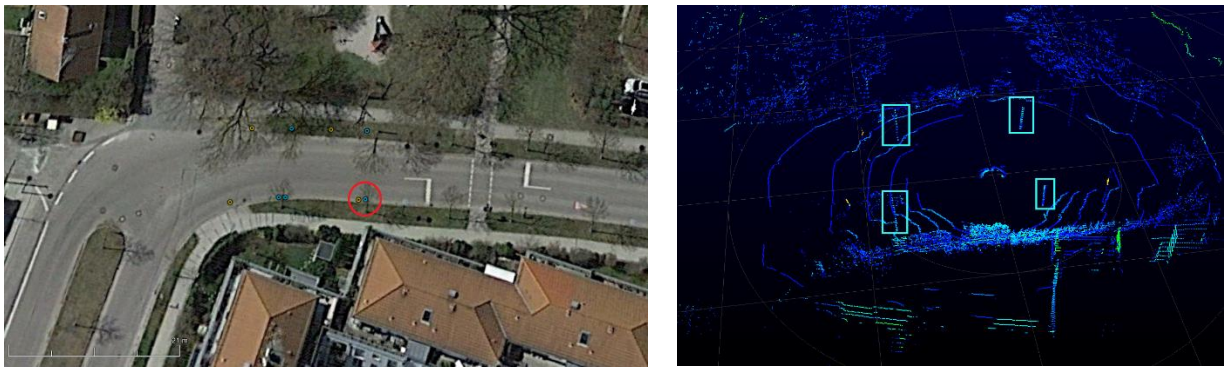

Figure 18. Georeferencing of cylinder's centroid (Frame 1792).

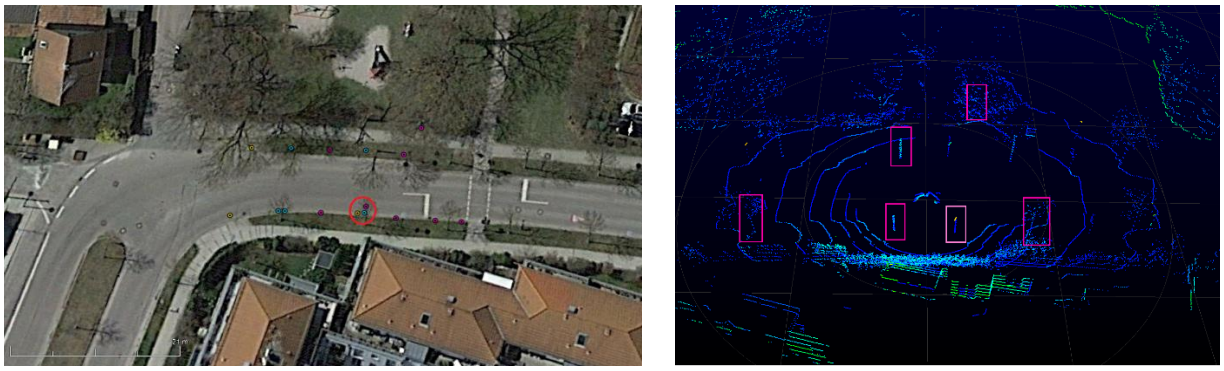

Figure 19. Georeferencing of cylinder's centroid (Frame 1795).

CONCLUSIONS

In this paper is proposed how LiDAR could be used for absolute and relative positioning. First, highly accurate georeferenced point clouds obtained with our system and the accuracy assessment of the results are presented. The accuracy obtained is the sub-decimeter level, although one must keep in mind that the problem encounter with occlusion of targets in real-time applications can lead to errors in the object detection procedure and therefore in the position estimation.

A relative navigation method was also proposed. The method can successfully detect objects with a cylindrical shape, which is our main interest as we are interested to detect no moving objects (trees, traffic signs, and post lights) from the LiDAR point clouds. When computing the displacement vector, a drift in the vertical component has been detected.

Finally, a practical case is shown in order to depict the idea of using the computed displacement vector and correct the car's absolute position when the estimated error of the GNSS-INS navigation solution crosses a certain threshold.

FUTURE WORK

Our future tasks will focus on assessing the quality of the shown practical case as well as the correction of the vertical drift. As said at the beginning of the paper, our goal is to develop a deep sensor fusion between GNSS-INS and LiDAR. Currently, we are working on the development of interfaces to our self-developed GNSS software receiver, which will allow us to keep researching and testing different integration solutions. Also, a more intelligent approach to detect no moving objects in the environment from the LiDAR data by means of machine learning is desirable.

ACKNOWLEDGMENTS

Special acknowledgment to our Master student Felix Mock for his support during the measurement campaigns. The results presented in this work were developed within the project "Autonome, optimale Fahrzeugnavigation und –steuerung im Fahrzeug-Fahrgast-Nahbereich für den städtischen Verkehr (short: AO-Car)", funded by the German Federal Ministry for Economic Affairs and Energy (BMWi) and administered by the Project Management Agency for Aeronautics Research of the German Space Agency (DLR) in Bonn, Germany (grant no. 50NA1616).

REFERENCES

[1] European GNSS Agency, *Central role for robust GNSS in autonomous driving.* [Online] Available: https://www.gsa.europa.eu/newsroom/news/central-role-robust-gnss-autonomous-driving#.

[2] M. S. Grewal, A. P. Andrews, and C. G. Bartone, *Global navigation satellite systems, inertial navigation, and integration*: John Wiley & Sons, 2013.

[3] X. Meng, H. Wang, and B. Liu, "A Robust Vehicle Localization Approach Based on GNSS/IMU/DMI/LiDAR Sensor Fusion for Autonomous Vehicles," *Sensors*, vol. 17, no. 9, p. 2140, 2017.

[4] C. Qian *et al.*, "An integrated GNSS/INS/LiDAR-SLAM positioning method for highly accurate forest stem mapping," *Remote Sensing*, vol. 9, no. 1, p. 3, 2016.

[5] J. Tang *et al.*, "LiDAR scan matching aided inertial navigation system in GNSS-denied environments," *Sensors*, vol. 15, no. 7, pp. 16710–16728, 2015.

[6] M. M. Miller *et al.*, "Navigation in gps denied environments: Feature-aided inertial systems," AIR FORCE RESEARCH LAB EGLIN AFB FL MUNITIONS DIRECTORATE, 2010.

[7] S. Schuhmacher and J. Böhm, "Georeferencing of terrestrial laserscanner data for applications in architectural modeling," 2005.

[8] J. Talaya *et al., Eds., Integration of a terrestrial laser scanner with GPS/IMU orientation sensors*, 2004.

[9] J. Hutton and M. Mostafa, "10 years of direct georeferencing for airborne photogrammetry," *GIS Business (GeoBit)*, vol. 11, no. 1, pp. 33–41, 2005.

[10] M. M. R. Mostafa, J. Hutton, and E. Lithopoulos, Eds., *Airborne direct georeferencing of frame imagery: An error budget*, 2001.

[11] A. Rizaldy and W. Firdaus, "Direct georeferencing: A new standard in photogrammetry for high accuracy mapping," *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 2012.

[12] G. Cai, B. M. Chen, and T. H. Lee, *Unmanned rotorcraft systems*: Springer Science & Business Media, 2011.

[13] P. J. Besl and N. D. McKay, Eds., *Method for registration of 3-D shapes*: International Society for Optics and Photonics, 1992.

[14] R. B. Rusu, "Semantic 3D object maps for everyday manipulation in human living environments," *KI-Künstliche Intelligenz*, vol. 24, no. 4, pp. 345–348, 2010.

[15] A. Nguyen and B. Le, Eds., *3D point cloud segmentation: A survey*, 2013.

[16] A. Aldoma *et al.*, "Tutorial: Point cloud library: Three-dimensional object recognition and 6 dof pose estimation," *IEEE Robotics & Automation Magazine*, vol. 19, no. 3, pp. 80–91, 2012.

[17] W. Wohlkinger and M. Vincze, Eds., *Ensemble of shape functions for 3d object classification*: IEEE, 2011.

[18] *The quadratic-chi histogram distance family*: Springer, 2010.

[19] G. H. Golub and C. Reinsch, "Singular value decomposition and least squares solutions," *Numerische mathematik*, vol. 14, no. 5, pp. 403–420, 1970.

[20] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.