2018 IEEE Intelligent Vehicles Symposium (IV)
Changshu, Suzhou, China, June 26-30, 2018

LIDAR Based Altitude Estimation for Autonomous Vehicles using Elevation Maps

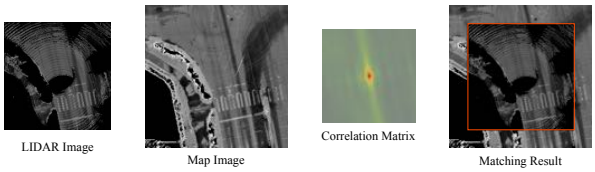
Ryo Yanase[1], Mohammad Aldibaja, Akisue Kuramoto, Kim Taehyon, Keisuke Yoneda and Naoki Suganuma

***Abstract*— This paper presents a strategy to estimate the vehicle altitude using predefined elevation maps. The map images in multilayers roads must be distinguished and separated with respect to the driving segments during the autonomous driving. This can be achieved by continuously estimating the altitude value and retiring the map images accordingly. The main influence is to increase the lateral and longitudinal localization based on matching the LIDAR and retrieving map images. Accordingly, a framework is designed and evaluated in complicated environments of common driving areas between bridges, tunnels and ground roads. The experimental results have verified that the proposed strategy is very reliable and provides precise altitude estimation with averaged accuracy of 10cm.**

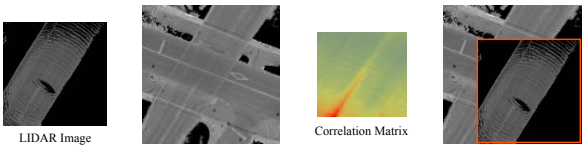
I. INTRODUCTION

Self-localization is a challenging issue of estimating the vehicle position in the real world [1] [2] [3]. We are developing an autonomous driving system using multiple sensor-data such as GNSS/INS, LIDAR, Radar and camera. Accordingly, a localization method has been implemented based on two-dimensional maps [4]. The predefined maps are generated in advance to describe the road surface and the surrounding environment in a gray level of intensity [5][6]. The vehicle position is calculated by matching the map and online-LIDAR data during the autonomous driving as illustrated in Fig. 1a. Two matching strategies, holistic and feature based, are combined to significantly utilize each pixel in LIDAR and map images in order to calculate the similarity score [7]. Therefore, the vehicle has been accurately localized with the range of 10cm along many courses in different cities in Japan. In addition, the calculation of the matching score has been enhanced in terms of processing time to enable autonomous driving at speed of 60Km/h. One can observe in the right image of Fig. 1a that the map image is perfectly overlapped by the LIDAR image based on the maximum matching peak in the correlation matrix. The lateral matching accuracy can be evaluated by the alignment degree of the lane lines between the map and LIDAR images whereas the longitudinal accuracy can be checked by matching the road context with respect to the drivable direction.

Regardless to the high localization accuracy throughout three test years, we observed that the altitude estimation must be integrated to the system. This is because the 2D map images may encode multiple road patterns at bridges, tunnels and multi-floor parking areas. In such situations, a considerable difference between the LIDAR and map images occurs and a wrong matching result is obtained accordingly. In Fig. 1b, the vehicle is on a bridge and the LIDAR image describes the road surface. The bridge is encoded in the corresponding map image with the ground road. Consequently, the matching result is wrong and a localization error in both longitudinal and lateral direction has occurred.



(a) Matching result on one-layer road.



(b) Wrong matching in two-layer road.

Fig. 1 Importance of altitude estimation.

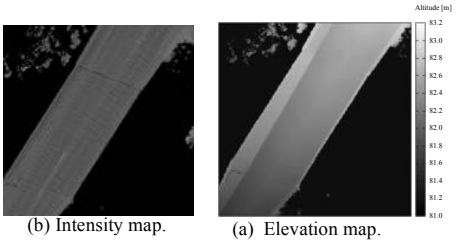
In order to improve the localization accuracy, some methods have been proposed to magnify the matching features in LIDAR 3D point cloud [8][9][10]. Yoneda et al. have combined the road surface patterns that obtained at different heights. The map is divided into multiple layers and Iterative Closest Point (ICP) scan matching is applied to each layer to fuse the pose estimation result with respect to the value of the error function.

In [10], the 3D point cloud is clustered into groups. Normal Distribution Transform (NDT) is then applied to approximate the point distribution in each group. Scan matching is performed on the approximated groups using ICP algorithm and the position is estimated accordingly. This method has provided a robust performance with reliable processing time and memory allocation.

M. A. Brubaker et al. have proposed a visual odometry-based localization method using Open Street Map [11]. The method provides accuracy in the range of several meters without using GNSS data. Thus, it is not qualified to be applied for localization. Since it does not require GNSS information, it is very effective to roughly estimate the initial position before activating the autonomous driving mode.

In this paper, a 3D pose estimation method using 3D point clouds and laser reflection intensity is proposed. The predefined intensity and elevation maps are automatically

Institute for frontier science initiative, Kanazawa University, Japan
[1]ryanase@staff.kanazawa-u.ac.jp



(b) Intensity map. (a) Elevation map.

Fig. 2 The mapping strategy.

generated from the LIDAR point clouds. The position in the lateral and longitudinal directions is estimated by calculating the matching score between the intensity-map and LIDAR images as in Fig. 1. Based on the matching area, the height information in the corresponding elevation-map and LIDAR-altitude images is used to estimate the altitude value in Z direction. The estimated z position is continuously updated and employed to retrieve the intensity maps. Therefore, the situation in Fig.1b can be overcome because the intensity map of the bridge can be distinguished from the ground road and retrieved accordingly.

II. Map Structure

The map is stored as 3D point clouds and every point possesses the 3D coordinates (x, y, z) and the intensity value. The intensity map encodes the road surface whereas the elevation map contains the corresponding altitude of each pixel as shown in Fig. 2, respectively. The resolution is set to 0.125m per pixel and the image size is 256×256.The pixels range in a gray scale from 0 to 255. In particular, 0 is a point where the laser beam of LIDAR was not irradiated and represents an unknown value. The map images are saved with global labeling IDs in the absolute coordinate system. These IDs can be calculated based on the 3D vehicle position X_t with respect to the pixel resolution Rse, the image size ($SS_w \times SS_h$) and the threshold Thr as in (1). Hence, the map images are retrieved during the autonomous driving continuously.

$$ID_t(x) = floor\left(\frac{\mathrm{X}_t(x) - Rse*\left(\frac{SS_w}{2}\right)}{Rse*SS_w}\right)$$
$$ID_t(y) = ceil\left(\frac{X_t(y) - Rse*\left(\frac{SS_h}{2}\right)}{Rse*SS_h}\right) \quad (1)$$
$$ID_t(z) = floor\left(\frac{\mathrm{X}_t(z)}{Thr}\right)$$

III. Proposed Self-Localization Method

A. Localization Strategy Overview

Dead reckoning is utilized to estimate the vehicle position $\left[x^v_{DR,t}, y^v_{DR,t}, z^v_{DR,t}\right]^T$ by time integration of the linear velocity. However, the estimation accuracy decreases proportionally to the driven distance from the initial position because of the error accumulation. The error is produced by the sensor noise and the localization redundancy. Therefore, the proposed method reduces the error by calculating an offset vector $\left[\Delta x_{DR,t}, \Delta y_{DR,t}, \Delta z_{DR,t}\right]^T$ in order to estimate the correct vehicle position $[x^v_t, y^v_t, z^v_t]^T$ as in (2).

$$\begin{bmatrix} x^v_t \\ y^v_t \\ z^v_t \end{bmatrix} = \begin{bmatrix} x^v_{DR,t-1} \\ y^v_{DR,t-1} \\ z^v_{DR,t-1} \end{bmatrix} - \begin{bmatrix} \Delta x_{DR,t} \\ \Delta y_{DR,t} \\ \Delta z_{DR,t} \end{bmatrix} \quad (2)$$

Fig. 3 shows the flowchart of computing the offsets by two main steps. The first step estimates the offset $\left[\Delta x_{DR,t}, \Delta y_{DR,t}\right]$ in the XY-direction by calculating the matching score between the LIDAR and intensity-map images. This estimation is used in the second step to compute the offset $\Delta z_{DR,t}$ in Z-direction based on the height difference between the LIDAR altitude image and the elevation map. As the first

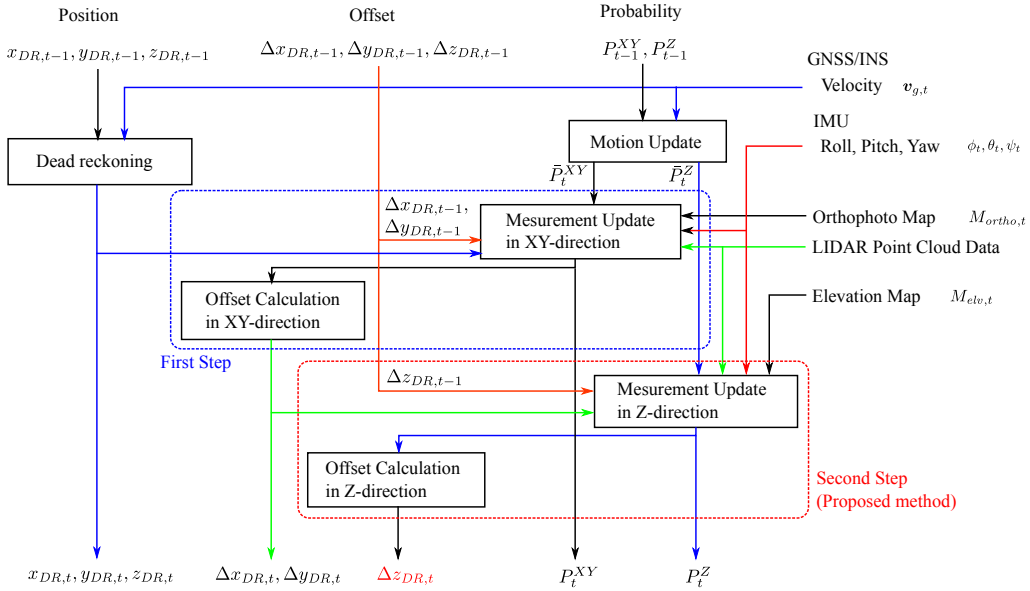
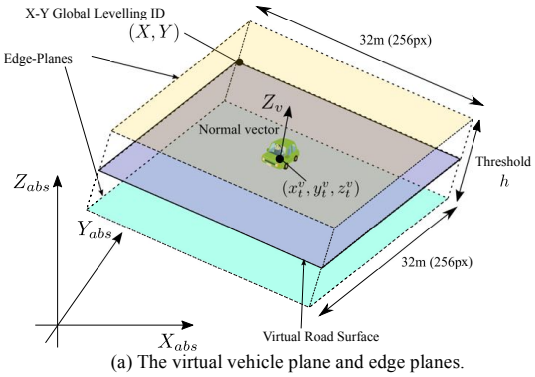
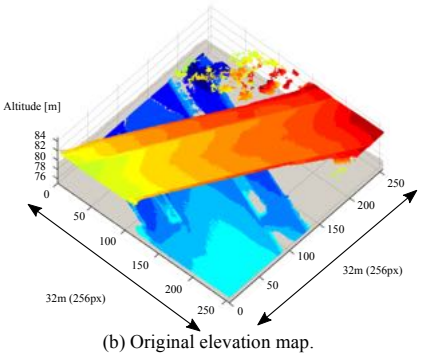


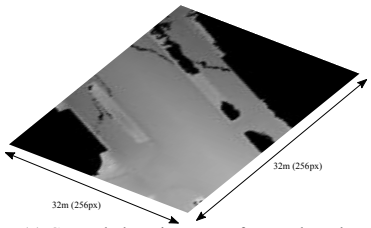
Fig. 3 Overview of the proposed method.



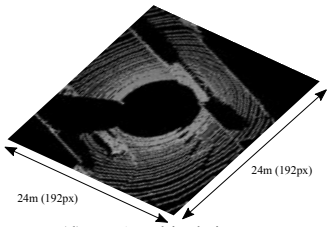
(a) The virtual vehicle plane and edge planes.



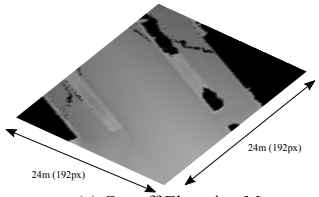
(b) Original elevation map.



(c) Created elevation map of ground road.



(d) LIDAR altitude image.



(e) Cut-off Elevation Map.

Fig. 4 Computing altitude of the road surface

step was implemented and discussed from different perspectives in the previous works [12] [13], the second step is detailed in the next section.

B. Computing Altitude of the Road Surface

In order to estimate the vehicle altitude, elevation-map image must be retrieved as well as the LIDAR altitude image must be created.

Given a rotation matrix R by IMU system at a vehicle position in the absolute coordinate system, a virtual plane with size of 32m×32m is created around the vehicle as illustrated in Fig. 4a and described by (3).

$$n_x\,(x - x_t^v) + n_y\,(y - y_t^v) + n_z\,(z - z_t^v) = 0 \quad (3)$$

where $[x, y, z]^T$ is an arbitrary position in the absolute coordinate system and $[n_x, n_y, n_z]^T$ is a normal vector of the plane $[n_x, n_y, n_z]^T = R\,[0, 0, 1]^T$. Another two edge-planes passing through the points of $(x_t^v + \frac{h}{2}n_x, y_t^v + \frac{h}{2}n_y, z_t^v + \frac{h}{2}n_z)$ and $(x_t^v - \frac{h}{2}n_x, y_t^v - \frac{h}{2}n_y, z_t^v - \frac{h}{2}n_z)$ are assumed with a threshold h. The X-Y global ID of the map image is calculated according to the vehicle position. Accordingly, the absolute coordinates of a pixel (i, j) can be obtained as follows:

$$x = (256 * 0.125)\,ID_t(x) + \ 0.125 * i \quad (4)$$

$$y = (256 * 0.125)\,ID_t(y) - \ 0.125 * j \quad (5)$$

Based on the corresponding elevation map $ID_t(z)$, the altitude of a pixel (i, j) can be conditionally obtained by (3) if z value is within the edge-planes. Fig. 4b shows an elevation map image with encoding two road layers (ground and bridge). The shaded area refers to the vehicle and edge planes. The pixels in this range are extracted to create the elevation map image A_{Map} of the current driving road as shown in Fig. 4c.

For creating the LIDAR altitude image, it is a very straightforward strategy to hold the z values of the point cloud in a matrix A_{LIDAR} (192px×192px). The index (i, j) of a pixel p' can be calculated using (6-8) with respect to the rotation matrix R.

$$p' = \begin{bmatrix} p'_x \\ p'_y \\ p'_z \end{bmatrix} = R\,p \quad (6)$$

$$i = 96 + \frac{p'_x}{0.125} \quad (7)$$

$$j = 96 - \frac{p'_y}{0.125} \quad (8)$$

The corresponding altitude of p' in the absolute coordinate system is obtained by adding the vehicle z position as in (9). Consequently, the LIDAR altitude image is formed as illustrated in Fig 4d.

$$z_{i,j} = p'_z + z_t^v. \quad (9)$$

In order to calculate the height difference, A_{Map} is cut-off to A'_{Map} with size of the LIDAR-altitude image based on the estimated offset $[\Delta x_{DR,t}, \Delta y_{DR,t}]$ as shown in Fig. 4e.

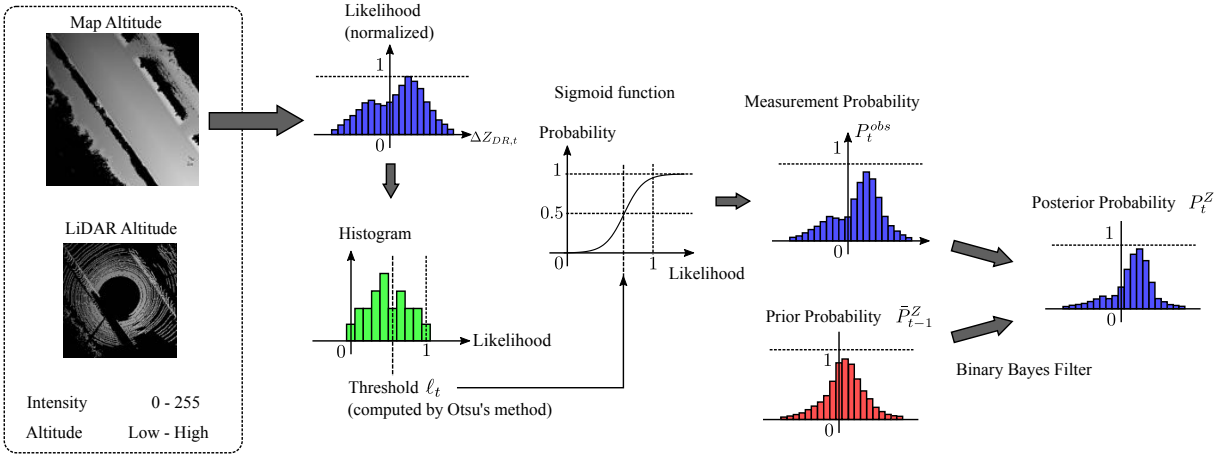


Fig. 5 Computation of posterior probability.

C. Offset Estimation in the Z-Direction

Likelihood distribution ℓ of $\Delta z_{DR,t}$ is calculated by creating a histogram on the number of common pixels (frequency) and the corresponding altitude differences between A'_{Map} and A_{LiDAR} as illustrated in Fig. 5. However, we observed that calculating simply the average value of the altitude differences is greatly influenced by the noise of the map and LIDAR data. In order to reduce the noise effects, the histogram is normalized and the Otsu's method is applied to calculate the optimal threshold ℓ_t [14]. The threshold is utilized to convert the likelihood in the histogram into an observation probability distribution P_t^{obs} using a sigmoid function as in (10).

$$P_t^{obs} = \frac{1}{1+e^{-a(\ell-\ell_t)}} \tag{10}$$

where a is a gain that assumed to be $a = 3.0$. As the distribution P_t^{obs} may change rapidly, Binary Bayes Filter (BBF) is incorporated to model the variance of the distribution with respect to the time [15]. It calculates the posterior probability P_t based on the prior probability distribution P_{t-1} and the observation probability P_t^{obs} as in (11).

$$P_t = \frac{1}{1+e^{-L_t}}, \tag{11}$$

where $L_t = \log\frac{P_t^{obs}}{1-P_t^{obs}} + \log\frac{P_{t-1}}{1-P_{t-1}}$. Hence, the offset $\Delta z_{DR,t}$ is obtained at the maximum point in the calculated posterior probability with achieving a smooth change of the distribution as demonstrated in Fig. 5.

IV. Evaluation and Discussion

A. Setup and platform

The experimental vehicle is equipped with sensors such as GNSS/INS (Applanix POS-LV220) and LiDAR (Velodyne HDL-64E) as shown in Fig. 6. GNSS/INS measurements were postprocessed to provide very accurate position data that regarded as ground truth. LiDAR data is captured with 10Hz to estimate the 3D vehicle position. Experiments were carried out using the autonomous driving data and a numerical evaluation was conducted to verify the robustness and reliability of the proposed method.

B. Experimental Result

Two scenarios are addressed in this section to emphasize the high accuracy of estimating the z value using the predefined elevation maps. In the first scenario, the vehicle was driven in a ground road and then turned up towards a bridge. The route is represented in Fig. 7a and a common area is highlighted by the yellow rectangle. Altitude estimation was continuously updated from the starting point using (11). The error between the ground truth and the estimated altitude is provided in Fig. 7b with highlighting the common area crossing in the shaded bars. One can observe that the maximum estimation error is around 15cm which is very compatible for conducting the autonomous driving. In addition, the road surface in the common area has successfully been separated based on the estimated z value. Accordingly, the corresponding map images have been retrieved with respect to vehicle position. Figs. 7c-d show the map images for both the ground and bridge roads as well as the corresponding LIDAR images, correlation matrices and X-Y matching images. By comparing these images to the result in Fig. 1b, the lateral and longitudinal localization becomes very accurate because of representing the actual environment in the map images and eliminating the other road layers.

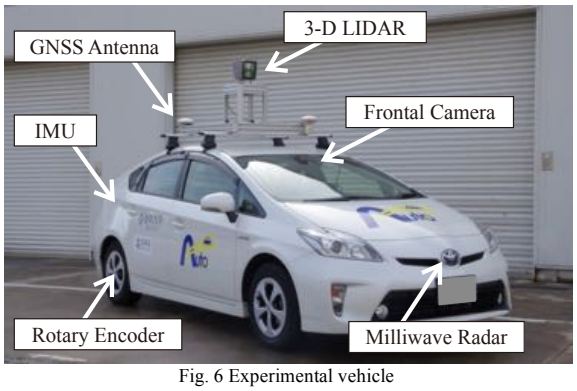
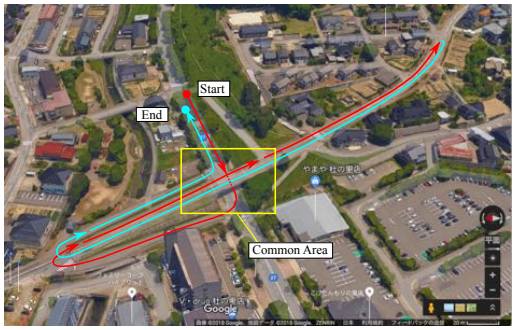
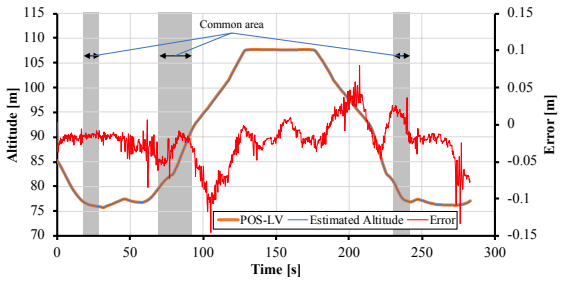


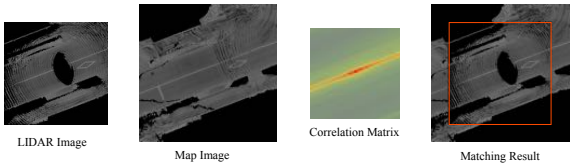
Fig. 6 Experimental vehicle



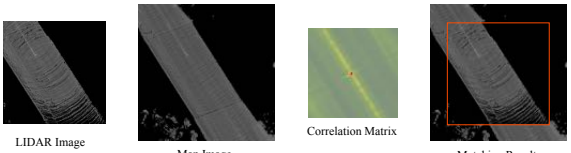
(a) Area view and common area.



(b) Altitude estimation.



(c) Lower layer (ground).



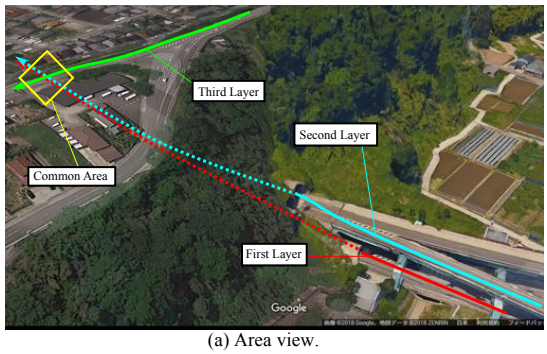
(d) Upper layer (bridge).

Fig. 7 Two-layer road (Ground and Bridge).

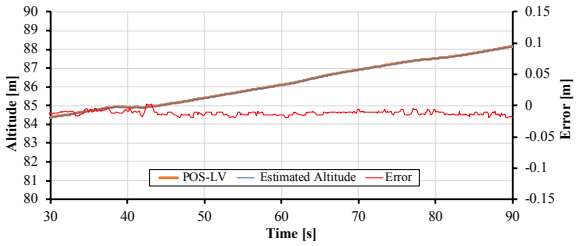
The second scenario is more complicated and illustrated in Fig. 8a. The course consists of three road layers, i.e., the first layer contains two tunnels whereas two bridges are combined in one tunnel in the second layer and the third layer possesses a straight road. Obviously, there is a common area between the three layers and the corresponding estimation error profiles are shown in Figs. 8b-d for each layer, respectively. One can observe that the largest error is 3cm which is very small and negligible compared to the height difference between any of two layers, i.e., 18m and 20m for first-second and second-third layers, respectively. This indicates the robustness of the proposed method to estimate the vehicle altitude in multilayer road environments.

V. Conclusion

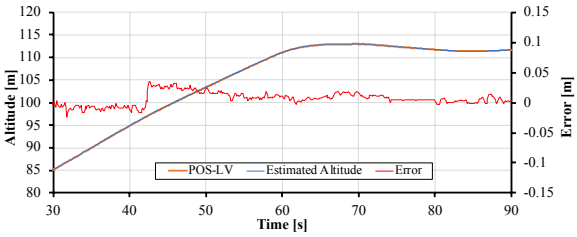
This paper explains the importance of estimating the



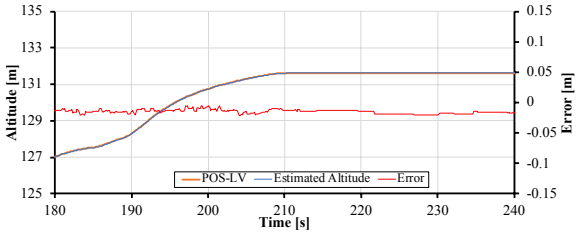
(a) Area view.



(b) First layer.



(c) Second layer.



(d) Third layer.

Fig. 8 Three-layer environment (Tunnel, Bridge and Mountain) and the altitude estimation error in the common area.

vehicle altitude to enable autonomous driving in multilayer road environments. A virtual road plane is assumed based on the current vehicle altitude. The pixels' altitudes in this plane are obtained using the predefined elevation maps. Meanwhile, the altitude observation image of the LIDAR point cloud is created and used to calculate the altitude differences with the pixels in the vehicle plane. The differences are computed in a probabilistic framework using Binary Bayes Filter. The filter guarantees a smooth change of the z value with respect to the time as well as it compensates the estimation error by modeling the system redundancy. The proposed method was evaluated in two complicated scenarios of driving in

multilayer roads. The obtained accuracy is in range of 10cm which allows to conduct autonomous driving confidently and safely. Therefore, the proposed framework is very reliable and the estimated *z* value can be used to distinguish the driving road layers precisely.

REFERENCES

[1] C. Urmson, J. Anhalt, D. Bagnell, et al., "Autonomous Driving in Urban Environments: Boss and the Urban Challenge", *Journal of Field Robotics*, vol. 25, no. 8, pp. 425-466, 2008.

[2] U. Franke, D. Pfeiffer, C. Rabe, C. Knoeppel, M. Enzweiler, F. Stein and R. G. Herrtwich, "Making Bertha See", *ICCV Workshop on Computer Vision for Autonomous Driving*, 2013.

[3] A.Broggi, P.Cerri, S.Debattisti, M.C.Laghi, P.P.Porta, M.Panciroli and A. Prioletti, "PROUD-Public Road Urban Driverless test: archi- tecture and results", *Proceedings of 2014 IEEE Intelligent Vehicles Symposium*, pp. 648-654, 2014.

[4] N. Suganuma, D. Yamamoto, "Map based localization of autonomous vehicle and its public urban road driving evaluation," *Proc. of 2015 IEEE/SICE International Symposium on System Integration(SII2015)*, pp. 467–471, 2015.

[5] M. Aldibaja, N. Suganuma and K. Yoneda, "LIDAR-data accumulation strategy to generate high definition maps for autonomous vehicles," *2017 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI)*, pp. 422-428, 2017.

[6] Mohammad Aldibaja, Naoki Suganuma, Keisuke Yoneda, Ryo Yanase and Akisue Kuramoto, "On autonomous driving: Why holistic and feature matching must be used in localization?," *International Conference on Intelligent Informatics and Biomedical Sciences (ICIIBMS)*, pp. 133-134, 2017

[7] C. Brenner, "Global Localization of Vehicles using Local Pole Patterns", *Pattern Recognition. Lecture Notes in Computer Science, Springer*, vol. 5748, pp. 61-70, 2009.

[8] J. Levinson and S. Thrun, "Robust Vehicle Localization in Urban Environments Using Probabilistic Maps", *Proceedings of 2010 IEEE International Conference on Robotics and Automation*, pp. 4372-4378, 2010.

[9] K. Yoneda, C. X. Yang, S. Mita, T. Okuya and K. Muto, "Urban Road Localization by using Multiple Layer Map Matching and Line Segment Matching", *Proceedings of 2015 IEEE Intelligent Vehicles Symposium*, pp.525-530, 2015.

[10] S. Kato, E. Takeuchi, Y. Ishiguro, Y. Ninomiya, K. Takeda, and T. Hamada. "An Open Approach to Autonomous Vehicles", *IEEE Micro*, vol. 35, no. 6, pp. 60-69, 2015.

[11] M. A. Brubaker, A. Geiger and R. Urtasun, "Lost! leveraging the crowd for probabilistic visual self-localization", *Proceedings of 2013 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3057-3064, 2013.

[12] K. Yoneda, N. Suganuma and M. Aldibaja, "Simultaneous Sate Recognition for Multiple Traffic Signals on Urban Road", *Proceedings of MECATRONICS-REM*, pp. 135-140, 2016.

[13] M. Aldibaja, N. Suganuma and K. Yoneda, "Improving Localization Accuracy for Autonomous Driving in Snow-Rain Environments", *Proceedings of 2016 IEEE/SICE International Symposium on System Integration*, pp. 212-217, 2016.

[14] N. Otsu, "A Threshold Selection Method from Gray-Level Histograms," in *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 9, no. 1, pp. 62-66, Jan. 1979.

[15] A. Elfes, "Occupancy grids: a probabilistic framework for robot perception and navigation," PhD thesis, Carnegie Mellon University, 1989.