

Article

Accurate Mobile Urban Mapping via Digital Map-Based SLAM [†]

Hyunchul Roh ¹, Jinyong Jeong ², Younggun Cho ² and Ayoung Kim ^{2,*}

¹ Robotics Program, Korea Advanced Institute of Science and Technology, Daejeon 34141, Korea; rohs_@kaist.ac.kr

² Department of Civil and Environmental Engineering, Korea Advanced Institute of Science and Technology, Daejeon 34141, Korea; jjy0923@kaist.ac.kr (J.J.), yg.cho@kaist.ac.kr (Y.C.)

* Correspondence: ayoungk@kaist.ac.kr; Tel.: +82-42-350-3632

† This paper is an expanded version of Jeong, J.; Kim, A. Adaptive Inverse Perspective Mapping for Lane Map Generation with SLAM. In Proceedings of the IEEE Ubiquitous Robots and Ambient Intelligence, Xi'an, China, 19–22 August 2016.

Academic Editor: Dale A. Quattrochi

Received: 7 June 2016; Accepted: 11 August 2016; Published: 18 August 2016

Abstract: This paper presents accurate urban map generation using digital map-based Simultaneous Localization and Mapping (SLAM). Throughout this work, our main objective is generating a 3D and lane map aiming for sub-meter accuracy. In conventional mapping approaches, achieving extremely high accuracy was performed by either (*i*) exploiting costly airborne sensors or (*ii*) surveying with a static mapping system in a stationary platform. Mobile scanning systems recently have gathered popularity but are mostly limited by the availability of the Global Positioning System (GPS). We focus on the fact that the availability of GPS and urban structures are both sporadic but complementary. By modeling both GPS and digital map data as measurements and integrating them with other sensor measurements, we leverage SLAM for an accurate mobile mapping system. Our proposed algorithm generates an efficient graph SLAM and achieves a framework running in real-time and targeting sub-meter accuracy with a mobile platform. Integrated with the SLAM framework, we implement a motion-adaptive model for the Inverse Perspective Mapping (IPM). Using motion estimation derived from SLAM, the experimental results show that the proposed approaches provide stable bird's-eye view images, even with significant motion during the drive. Our real-time map generation framework is validated via a long-distance urban test and evaluated at randomly sampled points using Real-Time Kinematic (RTK)-GPS.

Keywords: 3D mapping; SLAM; digital map; urban mapping system; IPM; lane map

1. Introduction

The recent development of autonomous vehicles encompasses many key issues in robotics problems, including perception, planning, control, localization and mapping. Among these problems, this paper focuses on solutions for an accurate urban map generation, specifically targeting 3D and lane maps for autonomous cars. Having an accurate map is an important issue for self-driving cars [1–3] and many other areas such as virtual reality [4], visualization [5], recognition [6], localization and navigation [7,8].

In conventional mapping approaches, aerial sensing has been heavily investigated. In these approaches, the fusion of aerial images with aerial Light Detection and Ranging (LiDAR) and/or radar is usually applied for an accurate digital map at the cm-level, including urban structures. This line of studies focuses on undistorting building shapes from aerial sensors, which is mostly too costly (ranging from \$0.5 M to \$1.4 M). To obtain cost-effective solutions for mapping, recent advances have

appeared in mobile mapping systems. Using a car-like system with sensors mounted, 3D maps are often generated from a set of point clouds, using LiDAR sensors to actively acquire accurate and dense 3D point clouds of building façades or surfaces of objects. Mobile scanning is widely used for modeling in architecture [4] and agriculture [9], as well as for urban and regional planning [10]. Many studies combine a highly accurate GPS, a high precision Inertial Measurement Unit (IMU) and the wheel/visual odometry of the vehicle to compute fully timestamped trajectories. The absence of the GPS in constricted spaces is an exceptionally limiting factor for mobile scanning. It is more critical, in a complex urban environment, and thus it is difficult to calculate accurate positions using a GPS sensor due to blackouts or multipath problems. Generally, using a variety of sensors has become a popular strategy to complement the drawbacks of a single sensor.

We propose a SLAM-based approach to localize the mobile sensor system mounted on a car-like platform and generate an accurate urban map consisting of a large amount of 3D points. For accurate localization, we use publicly available digital maps together with cameras, Light Detection and Ranging (LiDAR)s, and inertial sensors. Using this accurate localization, motion compensated Inverse Perspective Mapping (IPM) is applied for accurate lane-map generation. Given accurate localization, projections of front-looking cameras follow the construction of the lane map. For the lane map description, we propose an adaptive IPM algorithm to obtain accurate bird's-eye view images from the sequential images of forward-looking cameras. These images are often distorted by the motion of the vehicle; even a small motion can cause a substantial effect on bird's-eye view images.

To use a digital map in Simultaneous Localization and Mapping (SLAM), we incorporate a shape file to extract structural and elevation information. Although digital maps with 1:1000 and 1:5000 scales offer sub-meter global average accuracy, only a subset of map data (e.g., buildings and road boundaries which are globally static) are available for vehicle localization. In this paper, we introduce a measurement model for both GPS and structural information, and leverage the digital map data as observations to actively fuse digital maps in SLAM when GPS signals are sporadic.

The contribution of the proposed method can be summarized as follows.

- Digital map-based SLAM: Digital map information was incorporated with a building model, road network and elevation model. This paper introduces a measurement model to add global measurements for full 3D SLAM. Instead of addressing this problem as localization to a prior map, our modeling handles the structural information in the digital map as sporadic spatial measurements.
- Development of online mobile mapping system with sub-meter accuracy: Typical LiDAR-based mapping system known to be slow, suffering from association problems. Despite a large number of nodes and a long trajectory, the entire method runs in real-time using only 40% of the total mission time. Not only the computational time, our algorithm provides memory-efficient implementation. Wall information obtained from point cloud turns into an urban signature, which then can effectively be used in the matching phase. Fully integrated single step implementation allows us to build a 3D map over 9 km while driving.
- Thorough analysis for resulting 3D and lane maps accuracy: We perform a thorough analysis on the resulting maps to evaluate the effects of digital maps. The addition of elevation and buildings from digital maps enables accurate 3D urban mapping. We performed quantitative evaluation on random sample points. Using RTK-GPS with 10-mm-accuracy, we analyzed the accuracy of the points from buildings and lane maps.

2. Related Works

Conventional approaches in digital map generation rely on aerial sensors, including high-resolution aerial images, Light Detection and Ranging (LiDAR)s, and radars. Toward accurate building detection, point cloud data from aerial LiDAR has been widely used for a Digital Building Model (DBM) [11–14]. One line of study used only aerial imagery to extract building

information [15,16], while merging the high-resolution aerial imagery and aerial LiDAR [17] has been introduced.

The aforementioned approaches use costly aerial sensors and have limitations in generating maps inside tunnels and multi-layered roads. For a cost-effective mobile mapping system, urban mapping systems similar to ours have been introduced in the literature [18–21]. Blanco et al. [18] presented a collection of outdoor datasets using a large and heterogeneous set of sensors comprising color cameras, several laser scanners, precise GPSs, and an IMU. Their posterior research [19] introduced a dataset gathered entirely in urban scenarios using a car equipped with one stereo camera and five laser scanners, among other sensors. Elseberg et al. [20] used a commercial data logging system and collected data with an experimental platform constructed by the RIEGL scanner [22]. Bok et al. [21] used a mobile mapping sensor system mounted on a ground vehicle. A vertical 2D LiDAR scans structures and the reconstruction is done by accumulating scanned data. All of the conventional approaches cover a small region, assuming a 2D environment without taking altitude information into account.

For these mobile mapping systems, cameras are the most popular sensors in detecting lanes. For visual road-mark detection, IPM is often used in the vision-based perception of roads and lanes. IPM produces bird's eye-view images that remove the perspective effect by using information about camera parameters and the relationship between the camera and the ground. The results of IPM can provide post-processing algorithms with more efficient information, such as lane perception, mapping, localization, and pattern recognition. Many researchers have studied IPM in many applications, such as distance detection [23], production of bird's eye-view images of a spacious area using a mosaic method [24], provision of appropriate bird's eye-view images for parking assistance [25], and lane-level map generation [26].

The challenge for mobile mapping systems is to achieve a consistent map. Researchers investigated the SLAM algorithm for accurate vehicle localization [27] and 3D mapping [28] using perceptual sensors mounted on a vehicle. Using a digital map has also been recently investigated by many researchers [29–31]. Schindler et al. [29] first proposed lane map generation and representation methods using RTK GPS-based localization and showed high-precision digital map-based self-localization in [30]. They showed an efficient Monte Carlo localization approach using a map model based on smooth arc splines. Floros et al. [31] proposed an approach for global vehicle pose estimation that combines visual odometry with map information from OpenStreetMaps [32] to provide accurate estimates of the vehicle's pose. Similarly, Pink and Stiller [33] described landmark (lane-based) generation from aerial images with image classification geometric representation. This method used aerial images for prior maps and focused on the way to match orthographic lane images to a global lane map from aerial images. Guo et al. [26] presented a system for lane-level map generation from local IPM images and the global OpenStreetMap (OSM) database. They locally constructed an orthographic lane map, matched it with a segmented aerial map, and finally generated a global lane graph of real-world roads.

As an accurate localization aspect, some previous works [34–36] used a lane map as a local vehicle position estimation by matching pre-built local lane maps from IPM images and current incoming raw images. Napier and Newman [34] proposed a vehicle localization method based on synthesized local orthographic lane maps. They generated local orthographic images from a first run and localized the vehicle by Mutual Information (MI) between live-stream images and synthesized images. Schreiber et al. [35] constructed prior lane-based maps with extended sensors such as GPS/INS and matched road markings and curbs for lane-level localization. Rose et al. [36] used lane maps for lateral position estimations of a vehicle. These studies only applied localization or pose estimation algorithms; no cases have generated 3D maps using the SLAM algorithm with a digital map. In this paper, we are interested in using publicly available digital maps in a SLAM framework to leverage for accurate lane map and 3D map generation.

A similar approach to ours was reported in [37] which incorporated bundle adjustment using monocular vision. Although extracting buildings from digital maps and using them as constraints were introduced and conceptually similar to ours, their implementation relied mostly on vision, covering a relatively small urban area. Unlike their approach, proposed algorithm focuses on using LiDAR accomplishing real-time performance with a single-step mapping algorithm that can build an urban map while driving.

3. System Overview

For accurate urban mapping, we developed a sensor suite called Urban Mapping System (UMS), which is mountable to a car-like platform. As shown in Figure 1a, the platform used for mapping was equipped with three Light Detection and Ranging (LiDAR) sensors, four cameras, a Global Positioning System (GPS), an Inertial Measurement Unit (IMU), an altimeter, and wheel encoders. The detailed specifications are summarized in Table 1. As in Figure 1, coordinates of each sensor were represented with respect to the vehicle center coordinates (red, green and blue arrows).

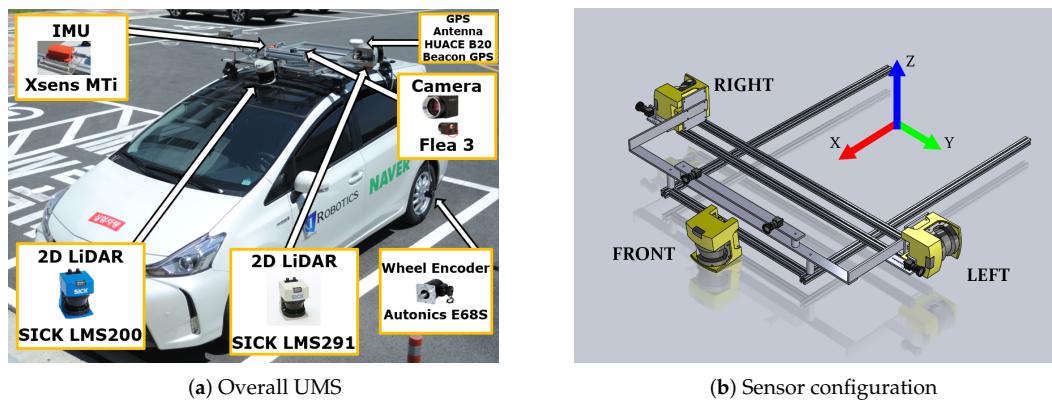


Figure 1. Sensor system description. Our sensor suite includes two 2D LiDARs, four cameras, a GPS, two wheel encoders, an IMU, and an altimeter. Note that the front LiDAR in (b) is only for moving object detection, and thus excluded in Simultaneous Localization and Mapping (SLAM) framework.

Using a geometrical relationship between the vehicle center and LiDAR, the position of the camera is computed using extrinsic calibration results of LiDAR and cameras. For more details, refer to [38] the system configuration.

Table 1. Specifications of Urban Mapping System (UMS).

Item	Specification
Dimensions	1.67 m × 1.36 m × 0.31 m (L × W × H)
Dry weight	35.8 kg
LiDAR	SICK LMS291,200 (35 Hz)
Imaging sensor	Point Grey Flea3, 1380 × 1024 pixel, 12-bit CCD (30 Hz)
GPS	HUACE B20 (1 Hz)
IMU sensor	Xsens MTi (100 Hz)
Altimeter	WITHROBOT myPressure (1 Hz)
Wheel encoder	Autonics E68S, rotary encoder type (100 Hz)
Processor	Intel(R) Core(TM) i7-3790 CPU@3.4 GHz
Battery	Delkor 80 Ah, 12 V, lead-acid type

In the proposed configuration, LiDAR sensors are facing each side of the vehicle, while the sweeping direction is orthogonal to the ground. This configuration allows us to capture surrounding data line-by-line. Motion induces accumulation of lines, and a 3D point cloud is obtained by stacking a

sequence of point cloud lines. Researchers using similar configurations recommended a vehicle speed of approximately 15–30 km/h [39], which we also adopted in the experiments.

The objective of the proposed platform is to build an accurate 3D and lane map given various sensors and digital maps. The process of generating urban lane maps with 3D point clouds involves four main steps as shown in Figure 2: (1) sensor data logging; (2) generating a node and the factor; (3) optimizing using an Incremental Smoothing and Mapping (iSAM) SLAM back-end; and (4) generating the final lane map based on the optimized trajectory from the SLAM results and IPM images. Measurements from odometry, digital map information, elevation, and GPS are all used.

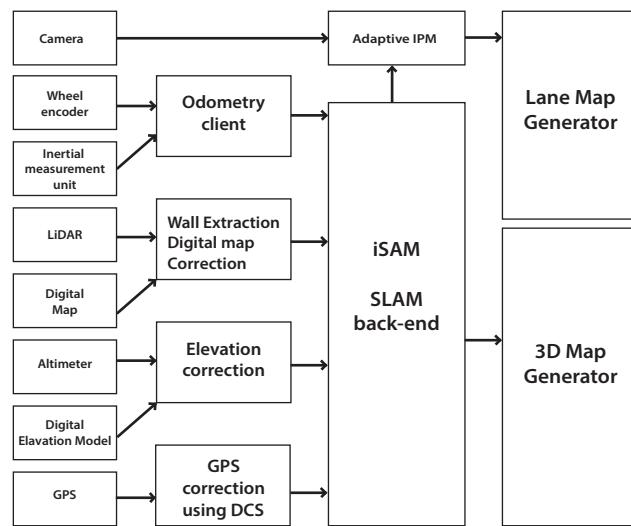


Figure 2. Software design diagram. The 3D point cloud map generation involves four main steps.

4. Pose-Graph Simultaneous Localization and Mapping (SLAM)

4.1. State Definition

We estimate the vehicle's full 6-degree of freedom (DOF) pose, $\mathbf{x} = [x, y, z, \phi, \theta, \psi]^T$. The augmented state representation is expressed as follows for n keyframes,

$$\mathbf{X} = [\mathbf{x}_1^T, \dots, \mathbf{x}_i^T, \dots, \mathbf{x}_n^T]^T \quad (1)$$

while each of the pose samples (\mathbf{x}_i) correspond to the time instance t_i of a keyframe. For the SLAM back-end, we use iSAM [40] to find an optimized solution of the trajectory, from which we build the urban map. Factors from each sensor measurement are collected as in Figure 3. The attitude measurements and altitude are given as absolute factors per node, with all nodes sequentially connected by odometry measurements. Loop-closing factors are mainly generated by walls that we generate from LiDAR when we add building measurement from digital maps. GPS availability is quite sporadic in urban area, and measurements are intermittently applied to a node using Dynamic Covariance Scaling (DCS).

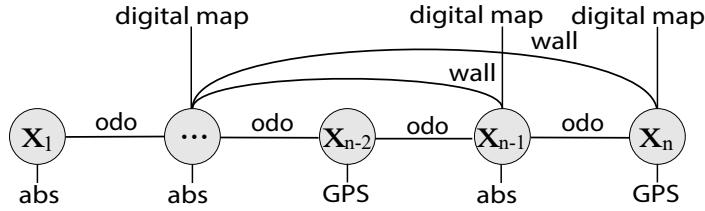


Figure 3. Depiction of the pose-graph SLAM factors. Odometry constraints (odo) are sequential whereas wall loop constraints (wall) can be non-sequential. Absolute constraints (abs) includes digital map correction and altimeter. GPS factors may occur irregularly. When wall is created, we also provide a digital map correction factor.

4.2. Odometry Modeling

The system has two rotary encoders, as shown in Figure 1a, which are mounted to record wheel revolution counts and the resulting rotation angles (C_l and C_r) of each wheel. We define vehicle pose in 2D space, $x = [x, y, \theta]^\top$ for odometry measurements. We calculate the relative pose difference as a function of left/right wheel diameter (D_l, D_r), left and right wheel rotation angle (C_l and C_r), wheel diameter (D_l, D_r), and wheel base (W_b).

$$\mathbf{z}_{odo} = \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta \theta \end{bmatrix} = \begin{bmatrix} L_{avg} \cos(\Delta\theta) \\ L_{avg} \sin(\Delta\theta) \\ L_{diff} W_b^{-1} \end{bmatrix} \quad (2)$$

Here, L_{avg} is the average and L_{diff} is the difference of two travel distances from the left and right wheel.

4.3. Altitude and IMU Modeling

We also provide z-directional measurements from the altitude sensor. When mapping hilly terrain, this leverages the localization performance by providing both absolute and relative measurements on height. We used the WITHROBOT MyPressure (WITHROBOT, Seoul, Korea) altitude sensor [41], which has accuracy within 1 cm. Attitude information was obtained via IMU. A MTI Xsens [42] was mounted into the system to provide roll, pitch, and yaw in 100 Hz. Altitude and attitude information is used as absolute measurement factors and fed into the SLAM back-end.

The proposed implementation produces loop-closure on height term based on wall information Section 5.3. When a loop-closure happens in wall-to-wall matching (i.e., when a wall is revisited), we apply a relative constraint on two associated nodes so as for them to be the same height.

4.4. GPS Modeling

GPS becomes extremely unreliable in highly complex urban environments. Signals are lost or deteriorate in urban canyons. Despite these limitations, GPS still provides valuable information to ground vehicles. For our GPS measurement error modeling, we mainly used the GPS Pseudorange Noise Statistics (GPGST), which include the standard deviation of longitude/latitude errors. The error measurement is the most sensitive, depending on the complexity of the environment, showing high error values in complex urban area.

Outlier handling in SLAM determines the robustness of the entire framework, since a single wrong measurement may critically deteriorate the results. To tackle this issue, Sünderhauf and Protzel [43] introduced Switchable Constraints (SC) and switching variables ($S_{ij} \in [0, 1]$) for the robust SLAM back-end. This research area was further developed by Agarwal et al. [44], who solve for a closed form solution to determine these switching variables $S_{ij} = \min \left(1, \frac{2\Phi}{\Phi + \chi_{ij}^2} \right)$. As can be seen, the choice depends on χ_{ij}^2 , which represents the error for each loop closing constraint. By considering the induced error by a factor, the associated covariance is dynamically scaled. This substantially improves the

results when a certain measurement may be unreliable during the mission, such as GPS. In this implementation, we adopted DCS to cope with intermittent and unreliable GPS signals.

5. Digital Map-Based SLAM

We introduce using a digital map with Light Detection and Ranging (LiDAR) sensor to correct the navigation error even under unreliable Global Positioning System (GPS). We first create walls from a 3D point cloud generated by LiDAR. Using this point cloud as a prior reference, the building edges are automatically detected to correct navigation error. In this section, we introduce the generation of walls, matching with a digital map, and matching between walls for loop-closure.

5.1. Digital Map

We leverage publicly available digital maps for SLAM application. Specifically, we focus on vectorized digital maps widely used in a geospatial information system. Studies that automatically generate the digital map have been continued in recent years [11–17,45–48] but it has mostly relied on aerial sensors. For the digital map data format, we chose to use a shapefile format developed by ESRI [49]. The shapefile stores non-topological geometry and attribute information for the spatial features as a set of vector coordinates. Vector features including points, lines, and polygons are described in the shapefile format. The format also represents area features via closed loop and double-digitized polygons. Each attribute record has a one-to-one relationship with the associated shape record. Among many types of records, we focus on polygons that contain information about structural buildings.

As illustrated in Figure 4a, a polygon consists of one or points that form a closed, non-self-intersecting loop. A polygon may contain more closed curves. A closed curve is a connected sequence of four or more multiple outer closed curves. The order of vertices or orientation for a closed curve indicates which side of the closed curve is the interior of the polygon. Recording contents of this polygon data type are shown in Table 2. Figure 5b presents an example of a building extracted from polygon data in a shapefile. In this paper, shapefiles from National Geographic Information Institute (NGII) are used for building extraction.

Table 2. Description on polygon record contents. The fields for a polygon type are box, numParts, numPoints, parts, and points. Box means the bounding box for the polygon stored in the order Xmin, Ymin, Xmax, Ymax. The number of closed curves in the polygon is described by NumParts. NumPoints is the total number of points for all closed curves. parts means an array of length numParts. For each closed curve, the index of its first point stored in the points array. Points are array of length numPoints. The points for each closed curve in the polygon are stored end to end.

Position	Field	Value	Type	Number	Byte Order
Byte 0	Shape Type	5	Integer	1	Little
Byte 4	Box	Box	Double	4	Little
Byte 36	NumParts	NumParts	Integer	1	Little
Byte 40	NumPoints	NumPoints	Integer	1	Little
Byte 44	Parts	Parts	Integer	NumParts	Little
Byte $X\ddagger$	Points	Points	Point	NumPoints	Little

$\ddagger X = 44 + 4 \times \text{NumParts}$

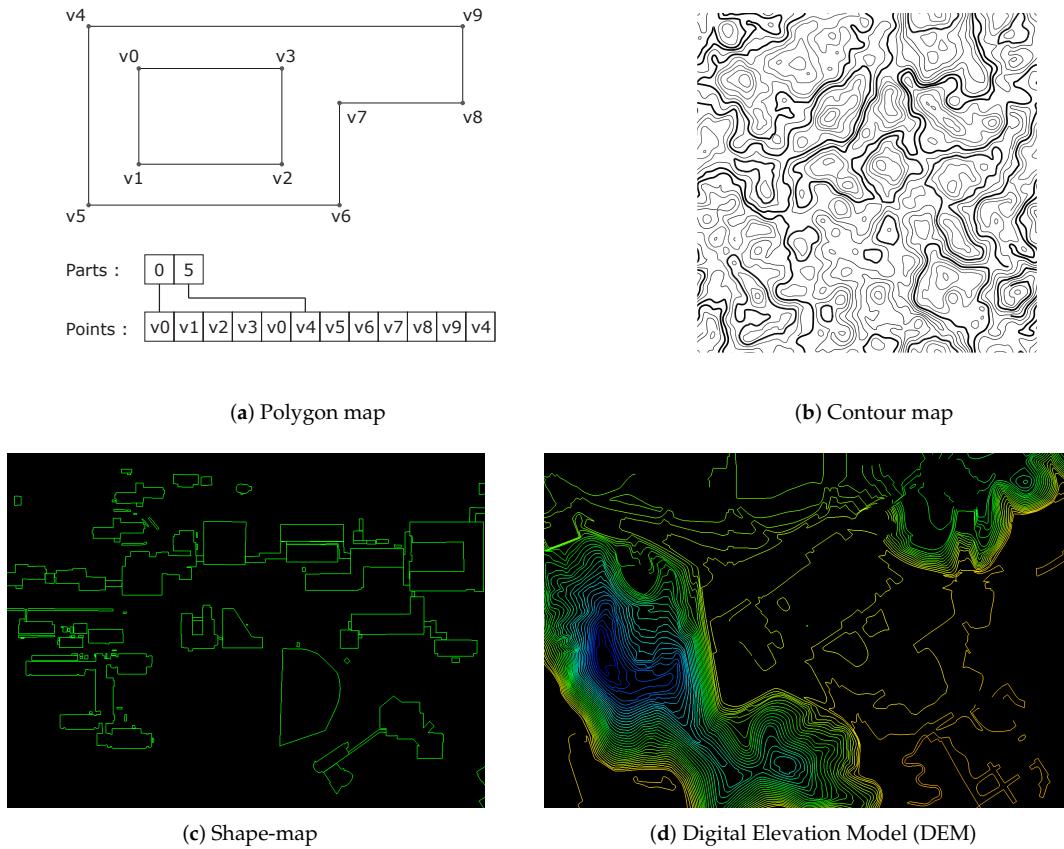


Figure 4. Shape file representations for building and contour. An example of polygon data type structure is in (a), and an example of contour map is in (b). (c) is sample shape-map (building) in the campus data. (d) is sample shape-map (DEM) in the campus data.

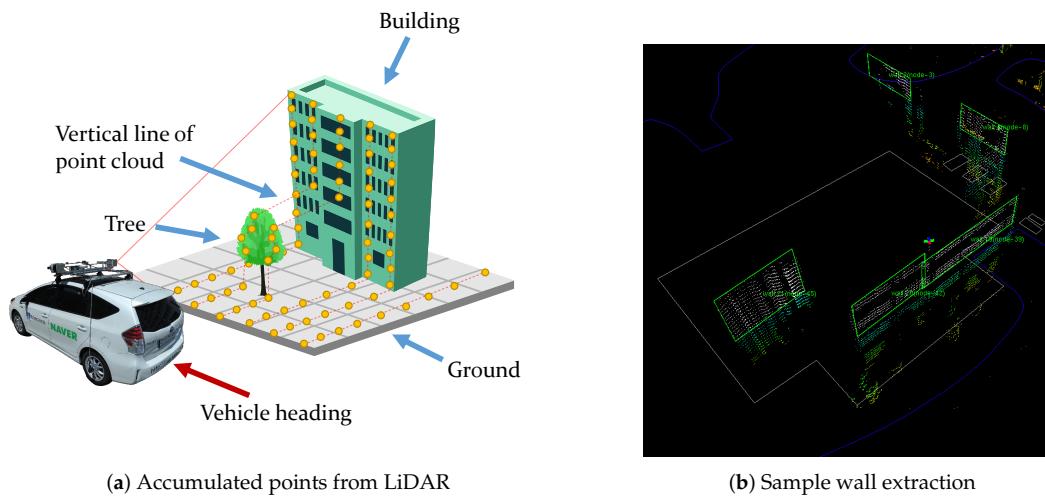


Figure 5. (a) Illustration of point cloud accumulation from vertically installed side scanning LiDAR. We only collect a single scan line per vehicle pose, and accumulate lines into a local point cloud set via motion. (b) An example of wall represented by digital map and point cloud map (white points in green box). For wall segmentation, we use 5-point Random Sample Consensus (RANSAC)-based plane fitting.

We also incorporate DEM for elevation information. For certain topographies, regions contain substantial elevation change. In general, the height of the terrain is represented by the contours. As presented in Figure 4d, points are connected by contour lines indicating the same height. The absolute height value of the contour is described in the attribute information of the shape-file. When the trajectory of the vehicle passes a contour, SLAM algorithm optimize position using absolute measurement of Z-direction from this absolute height.

5.2. Wall Segmentation

Given building information parsed from the digital maps, we now extract walls from the mobile sensor system. Building façades and other structures' surfaces are major features of the urban area. As digital maps are widely available these days, we propose a matching algorithm that uses a digital map and an accumulated local point cloud. In general, use of the point-to-point Iterated Closest Point (ICP) algorithm is popular but it is time-consuming and gives rise to issues when point clouds are not sufficiently dense.

We define a point cloud $P = \{p_1, p_2, p_3, \dots\}$ as consisting of LiDAR points, while each point indicates a 3D position $p_k = [x_k, y_k, z_k]$ in the global coordinate frame. From the LiDAR 3D point cloud in the sensor frame, (P_s) is obtained by converting each beam ranging from $i = 0$ to 360. The angular resolution of SICK LMS291 is $N_{res} = 0.5^\circ$, covering a 180° field of view for each size. Sensor frame point cloud P_s is then converted to vehicle frame point cloud P_v using a coordinate transformation.

After the transformation, we obtain a vertical line of the point cloud as described in Figure 5a. As the vehicle proceeds, these lines are accumulated and form a local point cloud set. Given a single scan line of LiDAR and odometry information from the wheel encoder and IMU, the set of the line forms a local point cloud-based on the estimated pose. During the local point cloud forming, we evaluate the wall criteria to see if the set can be segmented to a wall. When a wall is registered, the algorithm creates a node in the SLAM graph as a keyframe. Dropping the node clears the accumulated local point cloud and starts a new accumulation of incoming scan lines.

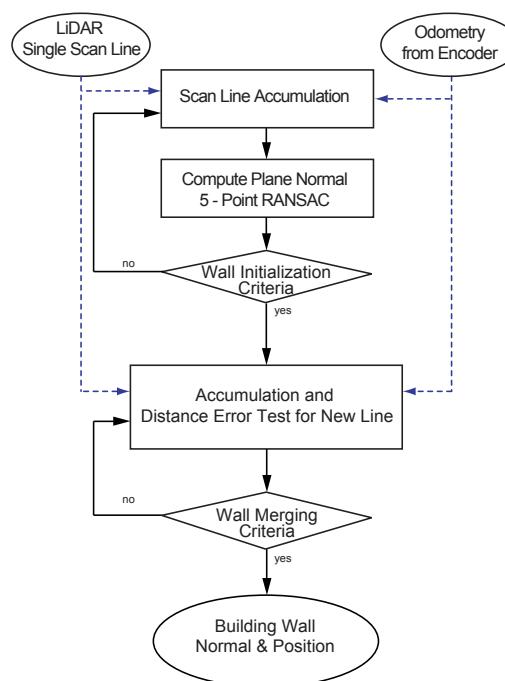


Figure 6. Block diagram for wall segmentation. The overall wall extraction algorithm can be divided into two parts. The first part is to obtain the initial normal, the second is a part of the segmentation wall to error distance test of the new incoming scan line by using initial normal information.

We propose a fast wall segmentation algorithm-based on RANSAC when segmenting a wall from a point cloud (summarized in Figure 6). For the incoming line of the point cloud, we fit a plane using a 5-point RANSAC. Our wall initialization criteria involve checking the wall width (initialization criteria in Figure 6). If the width of the created wall reaches our threshold (minimum 3 m), the algorithm initializes a wall and stores normal vector and error statistics for the wall. Once the initial wall is established, a newly incoming scan line is evaluated to be merged to the initial wall (merging criteria in Figure 6).

5.3. Wall-to-wall Loop Closing

For loop-closure, we use building wall patches by introducing a wall-to-wall measurement model. This method provides an explicit relationship between the walls in the factor graph and produces efficient maps using a data-rich 3D point cloud.

We parameterize the wall $\mathbf{B}_i = [n_i^x, n_i^y, n_i^z, p_i^x, p_i^y, p_i^z]^\top$ using normal vector $\mathbf{n}_i = [n_i^x, n_i^y, n_i^z]^\top$ and the center point of the wall $\mathbf{p}_i = [p_i^x, p_i^y, p_i^z]^\top$. Following wall coordinate transform operators \boxplus and \boxminus introduced by Paul Ozog [50] we derive our wall-to-wall measurement model. The \boxplus and \boxminus operations change the base coordinate frame that a plane is described in, which can be written as $\mathbf{B}_{jk} = \mathbf{x}_{ij} \boxminus \mathbf{B}_{ik}$ and $\mathbf{x}_{ij} \boxplus \mathbf{B}_{jk} = \ominus \mathbf{x}_{ij} \boxminus \mathbf{B}_{jk}$.

For two walls to be recognized as the same wall, they need to have the same normal vectors and the distance between a point in one plane to another plane should be zero. We compute them by evaluating the dot-product of two normals (e_n) and the distance from a center point on one wall to another wall (e_d).

$$\mathbf{z}_{\text{wall-wall}} = [e_n, e_d]^\top \quad (3)$$

Let a wall k seen at frame i to be \mathbf{B}_{ik} and the same wall revisited at frame j to be \mathbf{B}_{jk} . To compare normal vectors, \mathbf{B}_{ik} needs to be transformed to frame j using \boxminus operation. For simple notation we denote this as $\mathbf{B}'_{jk} = \mathbf{x}_{ij} \boxminus \mathbf{B}_{ik}$ with associated normal \mathbf{n}'_{jk} and the center point \mathbf{p}'_{jk} . Then the error between two normal vectors can be written as $e_n = 1 - (\mathbf{n}'_{jk})^\top \mathbf{n}_{jk}$. Similarly, the plane to point distance (e_d) is computed by measuring distance from \mathbf{B}'_{jk} to a wall center point in j frame (\mathbf{p}_{jk}).

It should be noted that we intentionally excluded the center point's coincident constraints. This is because our point cloud density strongly depends on the vehicle's motion as we accumulate line scans. When a vehicle takes turns the density may increase or decrease, and the computed center point may be varying with respect to the accumulated point cloud density.

5.4. Wall-Based Digital Map Localization

With odometry having a moderate accuracy, structural information from buildings can be used as a feature to correct the accumulated navigation error. However, buildings usually deteriorate the GPS signal reception. For a complex urban area, therefore, GPS and buildings can be exploited complementary. The digital map contains a variety of local information and there are vectorized building data among them.

An extracted wall from a point cloud represented by \mathbf{B}_i consists of normal vector $\mathbf{n}_i = [n_i^x, n_i^y, n_i^z]^\top$ and the center point of the wall $\mathbf{p}_i = [p_i^x, p_i^y, p_i^z]^\top$.

Similarly, we can calculate wall information that consists of normal vector and a center position from digital map \mathbf{d}_i in global coordinates. Using \boxplus operation and current node pose, we write extracted wall information from point cloud as $\mathbf{B}_{ik} = \mathbf{x}_{ij} \boxplus \mathbf{B}_{jk}$.

We obtain the measurement which consists of the dot-product of two normals (e_n) and the distance from a center point on one wall to another wall (e_d) in the same manner as in the Section 5.3.

$$\mathbf{z}_{\text{wall-digitalmap}} = [e_n, e_d]^\top \quad (4)$$

5.5. Elevation-based Full 3D Mapping

For full 3D mapping, we use DEM that consists of contour lines by introducing a node-to-contour measurement model. When the vehicle traverses a contour, the algorithm detects an intersection between vehicle motion vector and a line segment from the contour.

Let the vehicle pose be $\mathbf{x} = [x, y, z, \phi, \theta, \psi]^\top$. The DEM contour line, \mathbf{c} , consists of vertices $\mathbf{c}_i = [c_i^x, c_i^y, c_i^z]^\top$. Then we can extract two line segments, vehicle motion induced line ($\mathbf{l}_x = [x_{i-1}, y_{i-1}, z_{i-1}, x_i, y_i, z_i]$) that connects vehicle current node and previous node and contour line segment ($\mathbf{l}_c = [c_{k-1}^x, c_{k-1}^y, c_{k-1}^z, c_k^x, c_k^y, c_k^z]$) from a nearby contour. Using these two segments, we calculate the intersection point of these two line segments to determine the case when vehicle traverses contour line. Given vehicle pose that meets the DEM, our proposed method produces absolute measurement, which is described in the attribute information of the shape-file (Figure 7c).

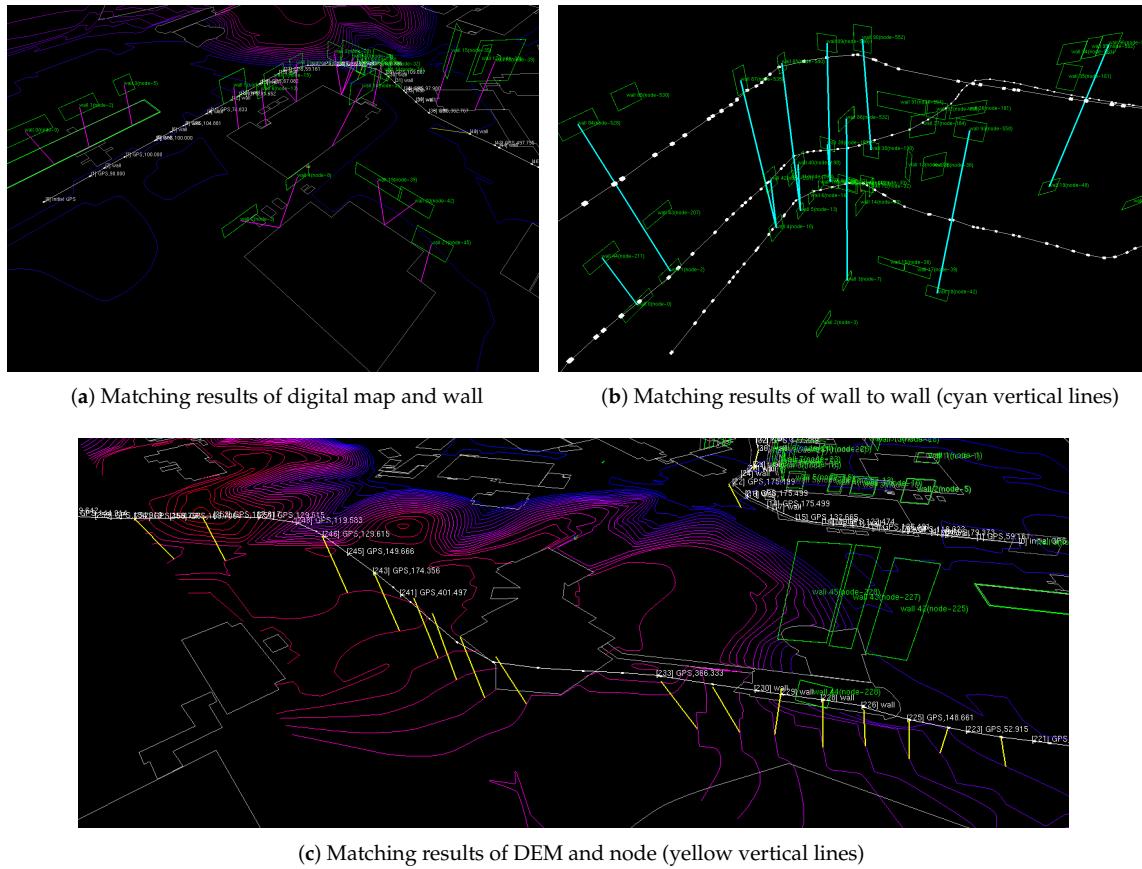
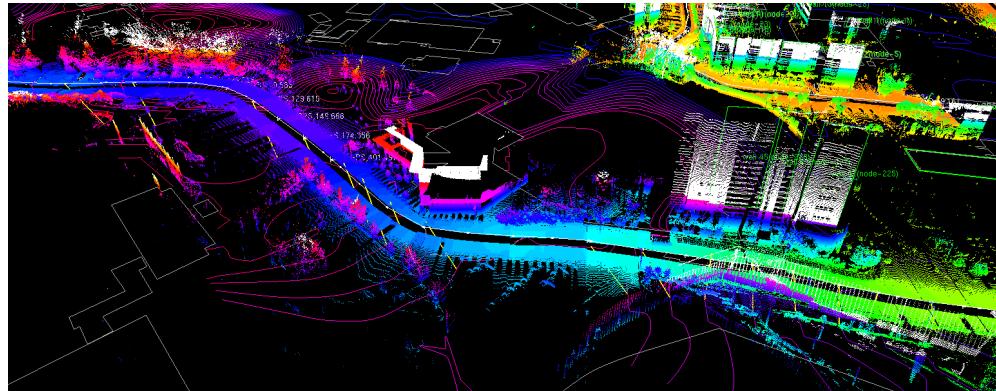


Figure 7. Cont.



(d) Matching results of DEM and node with point cloud

Figure 7. Result of digital map-based SLAM (a) Magenta line represents the correspondence of the digital map and building wall (green) extracted by point cloud. (b) Loop closure of trajectory generated using wall to wall matching showed by cyan line. (c) Correcting in the Z-axis direction, we use matching of Digital Elevation Model (DEM) and SLAM node. the yellow line represents this correspondence. (d) We can confirm the z-directional trajectory over the change of color.

6. Motion-Compensated Adaptive Lane Map Generation

Once we achieved an accurate localization by using a digital map-based SLAM, we apply IPM to back project road images onto the SLAM-induced map. This section briefly presents the basic IPM model [51] by using the physical parameters of a camera before illustrating an adaptive IPM model. IPM is a mathematical technique that relates to coordinate systems with different perspectives. More detailed derivation and explanation can be found in [52]. For lane map generation, we relate undisturbed bird's-eye view images and forward-facing distorted images (Figure 8).

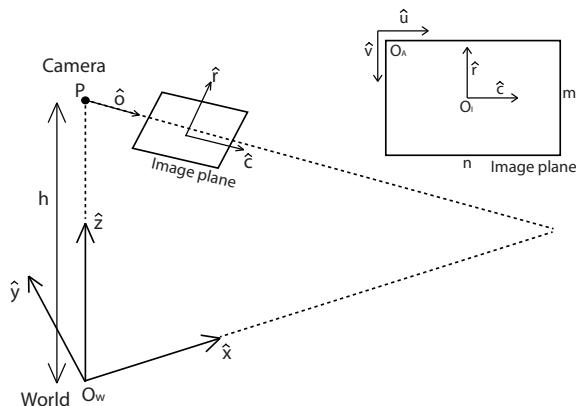


Figure 8. Coordinate the relationship between the camera, image and world. The unit of (\hat{u}, \hat{v}) is the pixel measurement, and that of (\hat{r}, \hat{c}) and $(\hat{X}, \hat{Y}, \hat{Z})$ is the meter measurement. Reprinted, with permission, from Jeong et al. URAI 2016; ©2016 IEEE.

6.1. Basic Inverse Perspective Mapping (IPM) Model

The objective is to map pixel points (u, v) to the world coordinate points (X, Y, Z) . The notation $(\hat{\cdot})$ indicates a vectored version of the variables. We first define a unit vector \hat{X} to set the camera's viewing direction. Being orthogonal to \hat{X} , we define another unit vector, \hat{Y} , that is orthogonal to the camera-viewing direction on the ground. The IPM is to find the relation between the world coordinate $(\hat{X}, \hat{Y}, \hat{Z})$ and image coordinate (\hat{u}, \hat{v}) in order to map image pixels to the world coordinate points. Note that two types of coordinates on an image are set as (\hat{u}, \hat{v}) and (\hat{r}, \hat{c}) , depending

on the unit. The relation between an image point in a pixel space (\hat{u}, \hat{v}) and the same point in a meter space (\hat{r}, \hat{c}) is defined as follows

$$\mathbf{u}(\mathbf{c}) = \frac{n+1}{2} + \mathbf{K}\mathbf{c} \longleftrightarrow \mathbf{c}(\mathbf{u}) = \frac{1}{\mathbf{K}}(\mathbf{u} - \frac{n+1}{2}) \quad (5)$$

$$\mathbf{v}(\mathbf{r}) = \frac{m+1}{2} - \mathbf{K}\mathbf{r} \longleftrightarrow \mathbf{r}(\mathbf{v}) = \frac{1}{\mathbf{K}}(\frac{m+1}{2} - \mathbf{v}) \quad (6)$$

with a scale factor between pixel and meter (px/m) \mathbf{K} , the image width \mathbf{m} , and the images height \mathbf{n} . The location of the camera's optical center (\mathbf{P}) is $[0, 0, h]$ in the world coordinate system. The unit vector of the optical axis $\hat{\delta}$ is orthogonal to the image plane.

Using the top and side view in Figure 9 we can derive Equations (7) and (8) for a fixed camera case.

$$\mathbf{X}(\mathbf{v}) = h \frac{\tan(\theta_0) \left(1 - 2 \frac{\mathbf{v}-1}{m-1}\right) \tan(\alpha_r) - 1}{\tan(\theta_0) + \left(1 - 2 \frac{\mathbf{v}-1}{m-1}\right) \tan(\alpha_r)} \quad (7)$$

$$\mathbf{Y}(\mathbf{u}, \mathbf{v}) = \left(1 - 2 \frac{\mathbf{u}-1}{n-1}\right) \tan(\alpha_c) \mathbf{X}(\mathbf{v}) \quad (8)$$

The location of $\mathbf{Y}(\mathbf{u}, \mathbf{v})$ in the world coordinate is dependent on (\mathbf{u}, \mathbf{v}) because $\mathbf{Y}(\mathbf{u}, \mathbf{v})$ includes $\mathbf{X}(\mathbf{v})$.

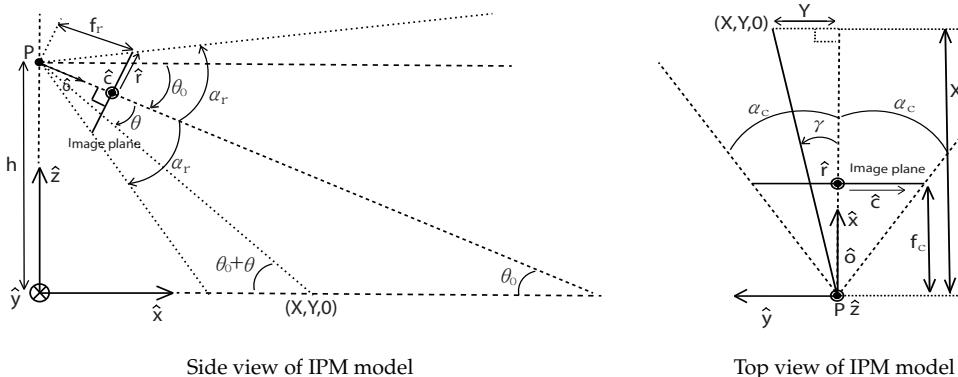


Figure 9. Side and top view of IPM model. In the illustration, f_r and f_c are the focal length of a camera; θ_0 is the pitch angle; α_r and α_c are the half angle of the vertical and horizontal field of view (FOV) respectively. Reprinted, with permission, from Jeong et al. URAI 2016; ©2016 IEEE.

6.2. Adaptive IPM Model

When images are obtained from a moving vehicle, it is difficult for them to be transformed to the accurate bird's-eye view images because of the motion of the vehicle, especially its pitch direction. To resolve this problem, oscillatory pitch angle (θ_p) of the camera is also added to original pitch angle (θ_0) in this model. This oscillatory pitch angle can be obtained either from sensors or via visual odometry. In this work, we exploited IMU sensor for oscillatory pitch angle.

$$\mathbf{X}(\mathbf{v}, \theta_p) = h \frac{\tan(\theta_0 + \theta_p) \left(1 - 2 \frac{\mathbf{v}-1}{m-1}\right) \tan(\alpha_r) - 1}{\tan(\theta_0 + \theta_p) + \left(1 - 2 \frac{\mathbf{v}-1}{m-1}\right) \tan(\alpha_r)} \quad (9)$$

$$\mathbf{Y}(\mathbf{u}, \mathbf{v}, \theta_p) = \left(1 - 2 \frac{\mathbf{u}-1}{n-1}\right) \tan(\alpha_c) \mathbf{X}(\mathbf{v}, \theta_p) \quad (10)$$

Finally, the adaptive Inverse Perspective Mapping (IPM) modeling Equations (9) and (10) can be derived by adding θ_p . $\mathbf{X}(v, \theta_p)$ is dependent on the pitch angle of the camera (θ_p) and $\mathbf{Y}(u, v, \theta_p)$ is

also dependent on it, which means that bird's-eye view images are properly compensated against the pitch angle.

6.3. Consistent Lane Map Processing

As described in Section 5, the algorithm drops nodes based upon the wall extraction events. Depending on the circumstances, structure occurrence may be sparse and thus produce a large gap between nodes. When the distance between two consecutive nodes are usually a couple of meters, inter-node inconsistency for measurements becomes substantial. This is because the SLAM algorithm only corrects navigation error for the nodes. Since only nodes are optimized through SLAM update, uncorrected measurements needs post-processing for self-consistent maps. This discrepancy statistics is shown in Table 3 that describes pose error prior to compensation. The error is computed as the distance and angle between the node position that is generated by the proposed method and is the expected pose by using the vehicle encoder. The cause of this discrepancy is encoder error from the wheel slip and modeling error, and this large discrepancy clearly indicates the need for the compensation. Smooth vehicle poses corrected with SLAM data are obtained after this compensation process.

Table 3. Discrepancy prior to the compensation (per unit meter).

	Min	Max	Average
x (m)	7.98×10^{-4}	9.37×10^{-2}	1.99×10^{-2}
y (m)	3.64×10^{-4}	1.49×10^0	4.13×10^{-2}
yaw (°)	2.08×10^{-3}	2.47×10^1	1.58×10^0

The effect of compensation is presented in Figure 10. In our algorithm, valid SLAM nodes are sparse, and thus are not compact enough to create a lane map via IPM. Interpolation is required for a dense map by updating the navigation correction between the nodes using vehicle encoder data. Figure 10a shows interpolated positions between nodes without any compensation. There are lots of errors between expected positions using encoder data and nodes because all nodes are corrected when wall extraction events occur. To overcome this problem, we apply a compensation rule, as described below.

$$E_k = \frac{\mathbf{p}_e - \mathbf{p}_n}{n} k \quad (11)$$

As shown in Figure 11, we compute the compensation value E_k per section between nodes. For a section with two nodes on each end, we compared the Dead-Reckoned (DR) position p_e and the SLAM-corrected node position p_n . When there are n number of images in the section we apply compensation value E_k for the position associated with k^{th} image. The result of this interpolation is shown in Figure 10c,d. Note that the compensated projection demonstrates a more smooth and consistent lane map.

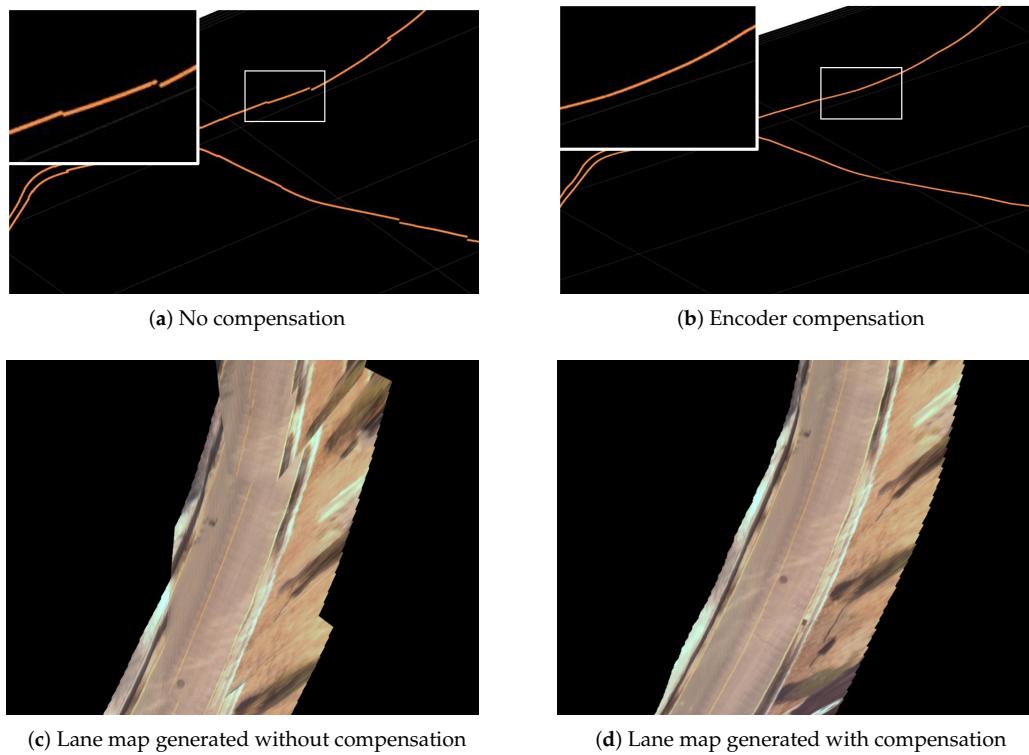


Figure 10. Result of self-consistent lane map process. Left (**a,c**) and right (**b,d**) images show the result of lane map.

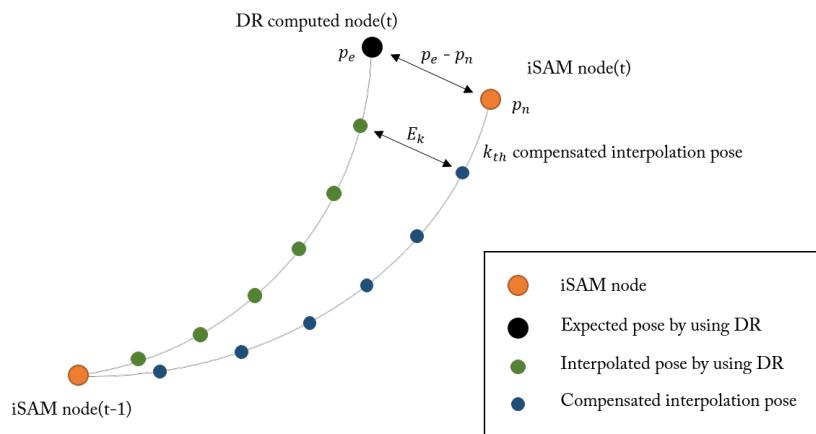


Figure 11. Compensation process of interpolated pose between DR computed node and iSAM node.

7. Experiments and Results

To validate the proposed method in the real world, we conduct an experiment covering a campus area with 9.32 km path length. The dataset is the route around a campus environment including four loop-closures by passing the center building repeatedly. The campus area is usually composed of low-rise buildings and wide roads that offer sufficient but sporadic GPS signals. As can be seen in Table 4, we have a 41.4% of GPS signal reception.

The overview of the entire campus area is as in Figure 12 depicted in the top-down and perspective view with a scale marked. In total 9.32 km of path is covered in 32 min. Overall the campus shows an altitude difference of 29.5 m between the northern and southern part. The final SLAM graph consists of 1165 nodes, and the average distance between the nodes is 8.06 m.

Table 4. Summary of Digital-Based SLAM results

Path Length	Logging Time	Computation Time	GPS Node	Wall Node	Digital Map Node	Total Nodes	No. of Point
9322.35m	1952.06s	792.04s (40.57%)	482 (41.4%)	249 (20.09%)	136 (11.67%)	1165	23,017,120

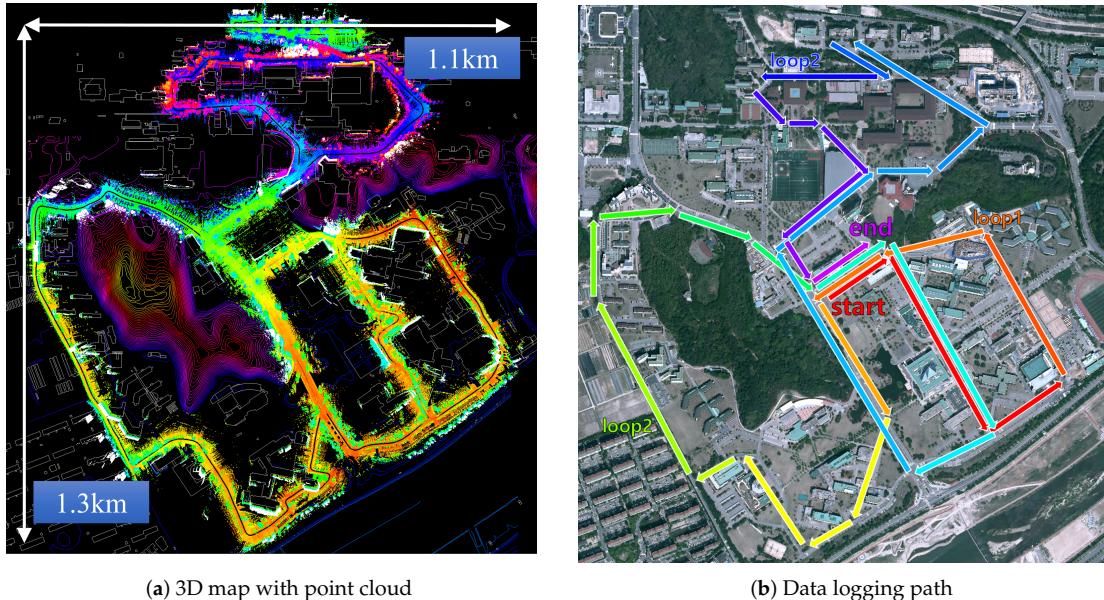


Figure 12. 3D mapping result and data logging path. **(a)** 3D map result represented by pseudo-colored point cloud in digital map background. Nodes are color-coded by height varying from red (low) to purple (high). **(b)** Illustration of data logging path performed in the experiments. Connection of the trajectory represented by color shift from red (start) to purple (end). There are three circular paths covering the eastern, the western, and the northern part of the campus.

7.1. SLAM Results

We first evaluate the SLAM results. Compared to Dead-Reckoned (DR), SLAM provides a more consistent map as shown in Figure 13b. The position is a value converted to a standard reference point ($38^{\circ}00'00''\text{N}/129^{\circ}00'00''\text{E}$), which follows the method of transverse Mercator coordinates. The algorithm was capable of making four times of loop closure successfully (Figure 13c) using a wall-to-wall loop-closing measurements. The node uncertainty propagation plot shows the SLAM effectively bounds the uncertainty (Figure 13a). We plot both the entire covariance determinant ($\sqrt[6]{\Sigma}$ in $\text{m} \cdot \text{rad}$) and the positional covariance determinant ($\sqrt[6]{\Sigma_{xyz}}$ in m). DR shows approximately 10% of the positional uncertainty, while SLAM results are mutually bounded by wall-to-wall loop closure, GPS and digital map correction algorithm. In Table 4, the computation time with respect to the total mission time is summarized. The entire method runs in real time using only 40% of the total mission time. This is due to the fact that the low-rise building allowed more GPS signal reception, because the number of the GPS and the wall node was the critical factor in terms of process time.

Figure 14 visualizes the nodes by the sensor types for the entire trajectory. As the environment is with low level buildings, GPS signals (blue) are available for many nodes (41.4%). However, the signal is sporadic and vulnerable to the surrounding structures. When the GPS signal becomes unreliable we use complementarity available structural information and correct against the digital map (green). DEM is also effective when altitude change is substantial within a region.

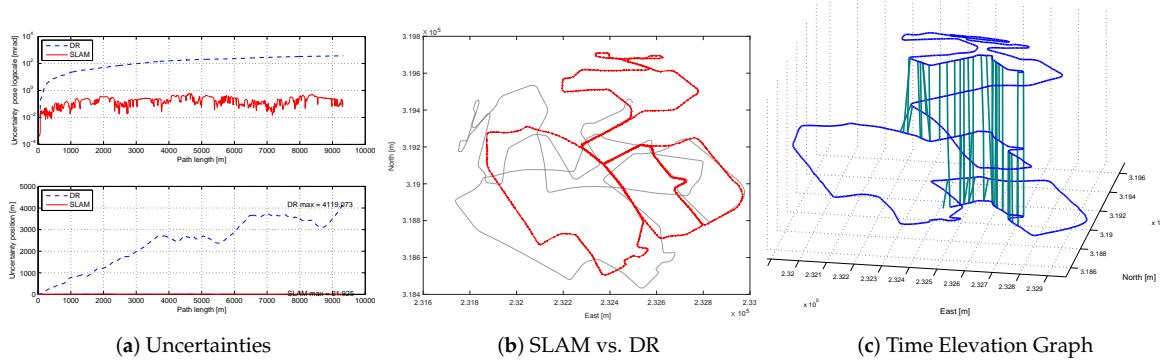


Figure 13. Proposed digital map-based SLAM results. (a) Uncertainty versus path length . Whereas the uncertainty of DR navigation (blue, dot) shows unbounded growth, the uncertainty of SLAM (red solid) is bounded from wall-to-wall loop closures. The graph at the top of the pose uncertainty with log-scale of y axis (unit: m · rad). The following [53–55], we use m · rad to show pose uncertainty), the bottom graph shows the position uncertainty (unit: m). (b) Top-down view of the SLAM estimate (red line) versus dead-reckoning trajectory (gray line). (c) The xy component of the SLAM trajectory estimate is plotted versus time, where the vertical axis represents mission time. Green lines show wall-to-wall matching for loop closure.

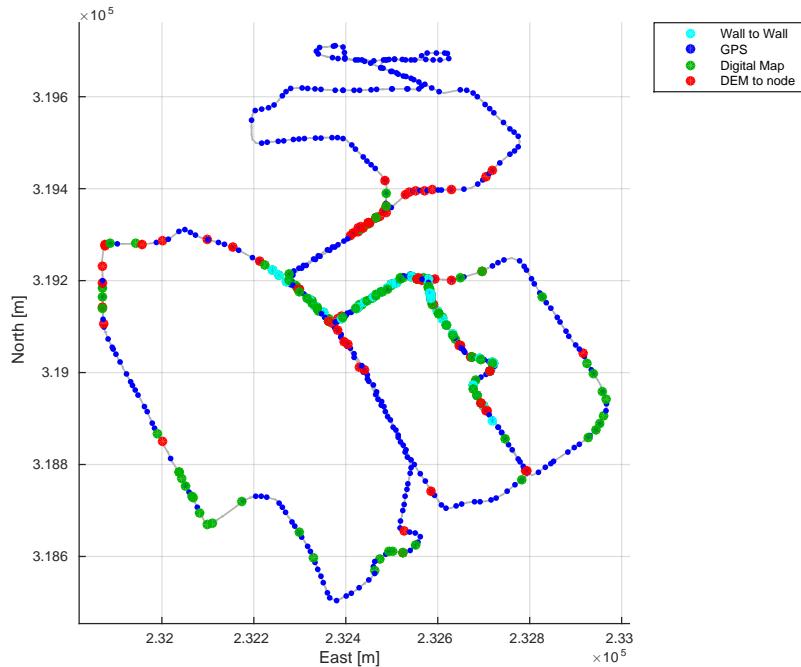


Figure 14. Sensor availability graph. The nodes are colored by the sensor type associated with the node. Blue dots are the GPS nodes and are in a smaller size for clear visualization for digital map nodes (green and red). Cyan nodes are loop-closure nodes via LiDAR comparison.

7.2. Qualitative Urban Mapping Results

For further validation of the proposed method, we back-project 3D point clouds and images using the accurate localization and mapping results. By doing so, we can generate a 3D urban map with lanes potentially for autonomous car driving. By confirming the consistency in the back-projected dataset, we validate the accuracy of the proposed method and application to urban map generation.

7.2.1. 3D Point Cloud Map

Collected LiDAR points are locally accurate with respect to the vehicle pose under exact extrinsic calibration. However, the entire map consistency may not be guaranteed if the estimated trajectory is with errors. Using the SLAM refined trajectory, we use point cloud information in order to create a 3D map of the urban environment. Figure 15 presents a qualitative result on two sample buildings created from the back-projection. As can be seen in the sample buildings of Figure 15, the proposed method is capable of detecting even highly complex walls. The accuracy of the 3D modeling is significantly affected by the trajectory estimation accuracy. The SLAM based method enables accurate estimation even under a complex motion with many turns (Figure 15c).

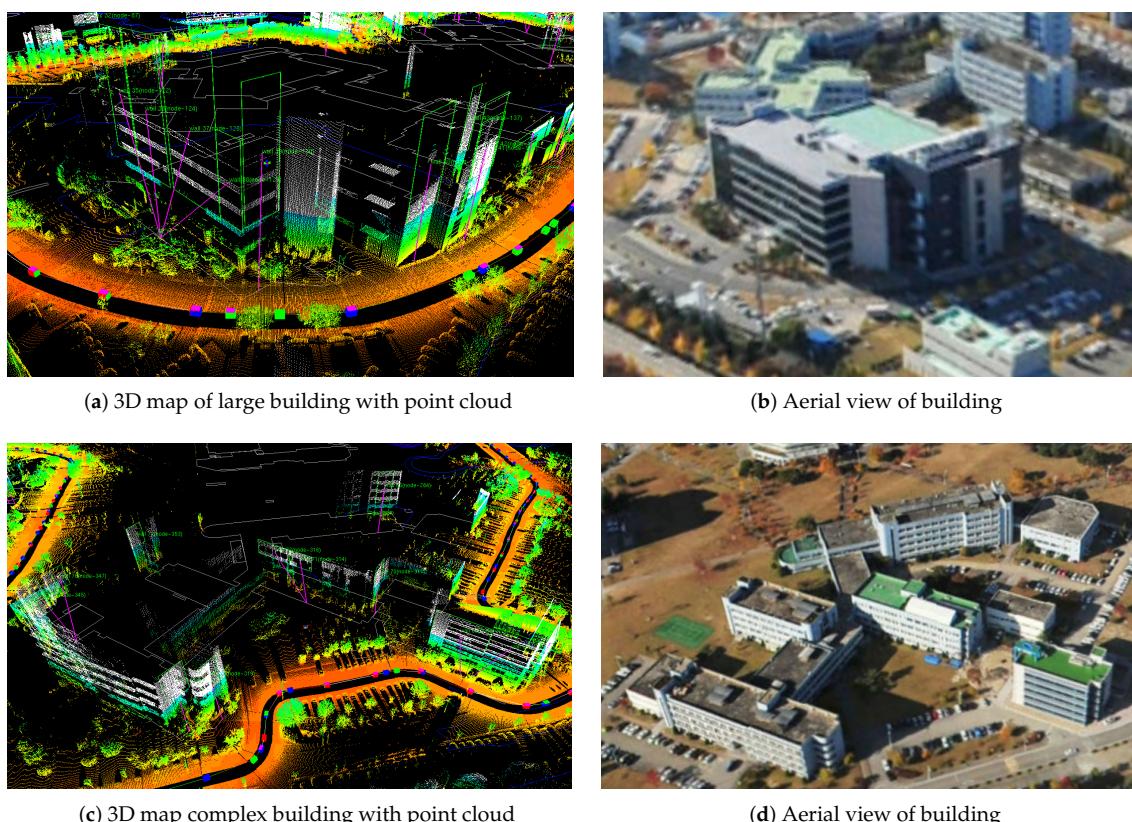


Figure 15. 3D mapping result for sample buildings. Accumulated and refined point cloud using SLAM trajectory is given. (a) and (c) are the 3D mapping of two samples. Points are colored by the height, and green squares indicate the classified building walls. (b) and (d) are the aerial view of each sample building.

7.2.2. Lane Map via IPM

We also generate a lane map for urban mapping. Using the Simultaneous Localization and Mapping (SLAM) results and the previously introduced adaptive Inverse Perspective Mapping (IPM) model, we produce precise bird's eye view images compensating for vehicle motion. To generate dense and precise lane map, additional vehicle positions are required because the average distance between nodes of SLAM result is 8.06 m. Additional vehicle position and heading angle are interpolated by using vehicle encoder data from nodes that generated by SLAM. Figure 16 shows the entire campus lane map with four representative samples (shown in a zoomed view). Note the consistency of the generated map by the IPM images back-projection.

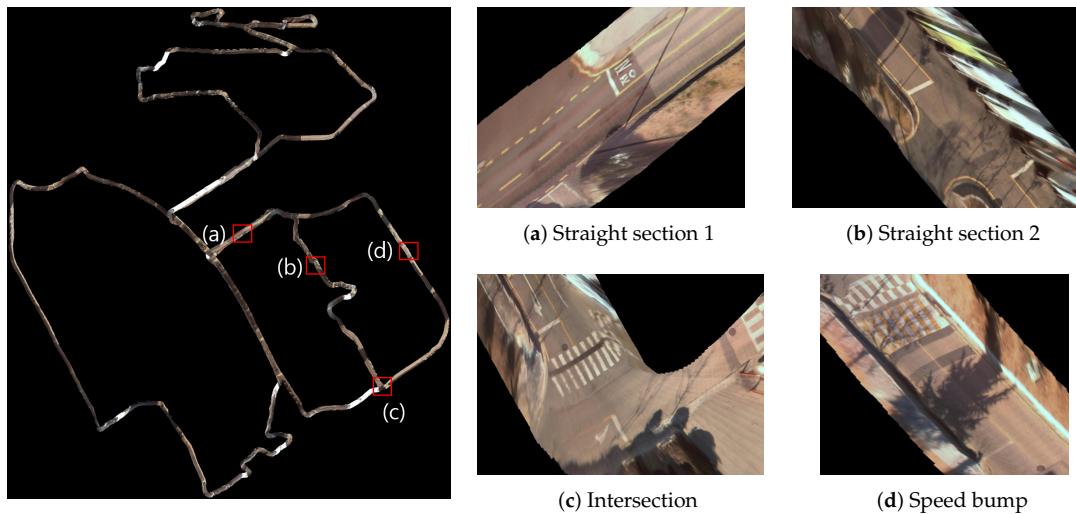


Figure 16. Generated lane map. (a) and (b) is a narrow straight section without and with parked car respectively. (c) is intersection and (d) includes speed bump and crosswalk.

7.3. Accuracy Analysis

In the previous section, we have demonstrated consistency and qualitative evaluation. We also performed quantitative evaluation on the mapping results. Using RTK-GPS with 10 mm-accuracy, we analyzed the accuracy of the points from both sample buildings and lane maps. We use GRX2 by Topcon [56] in the RTK mode to measure ground truth point sampling. As shown in Figure 17, we measure sample points from both buildings and lanes.

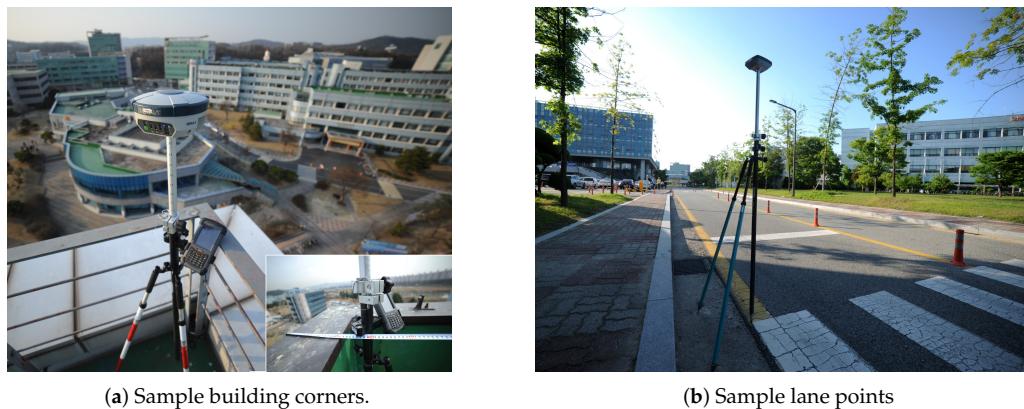


Figure 17. Accuracy analysis on sample points using VRS-GPS. (a) Four corners of the building rooftop are measured with building walls compensation. (b) Sample points on road marks are measured.

We analyze the effect of each measurement model for the entire map generation. Our evaluation process is described in Figure 18. Four building corners measured by RTK-GPS are connected to complete the boundary of a building (red square in Figure 18a). By measuring a perpendicular distance from LiDAR-extracted wall to the ground truth, we compute the Root Mean Square Error (RMSE) mapping error for 3D structures. The error analysis is summarized in Table 5. GPS-based method shows about a meter accuracy due to the sensor accuracy and unreliable availability. Compared to GPS-based mapping, the digital map-based SLAM shows substantial improvement over the conventional approach.

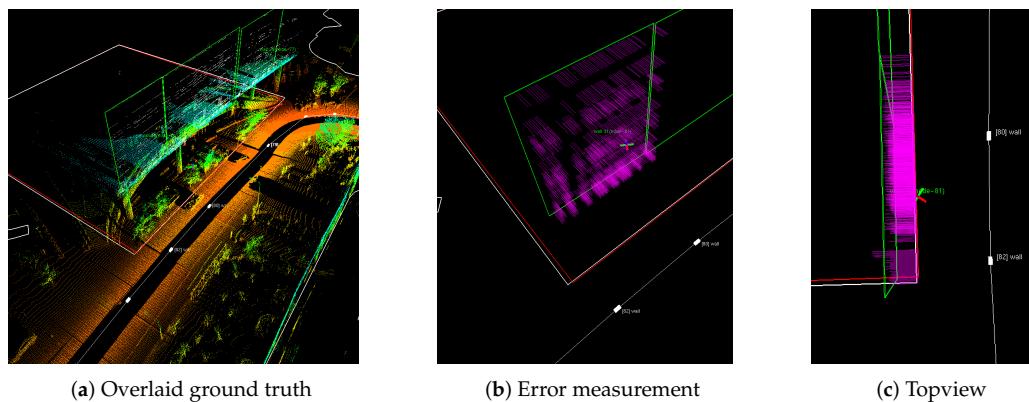


Figure 18. (a) Red square drawn from four sample ground truth corners. (b) By measuring perpendicular distance (purple line) from 3D point cloud to ground truth (red line) error is measured. (c) shows the topview for a clear illustration.

Table 5. Positional error measurement between the 3D building point cloud and RTK-GPS measured building corners. The map error is computed from average of RMSE between the ground truth wall generated from RTK measured sample points and LiDAR 3D points in the mapped wall. Note that this error is over 9.32 km of travel distance.

Sample	No. of 3D Map Points	Average RMSE	
		GPS Only [m]	Digital Map-Based [m]
Set 1	3352	0.437	0.190
Set 2	1967	1.010	0.193
Set 3	227	2.070	0.347
Set 4	3314	1.407	0.136

Similarly, we examine accuracy on a lane map from a set of sample points on road markings. In total, 16 points from five different sets are measured. Among them, Figure 19 presents validation process for a sample set with six sample points. Six RTK-GPS positions are obtained and used as ground truth (Figure 19a). In this analysis, we compare the proposed lane map with aerial image and digital map. Overlaying ground truth on aerial images (Figure 19c) reveals substantial discrepancy due to distortion without compensation from other sensors. In digital maps, road boundaries are accurate but road marking and lanes are missing. Compared to digital and aerial maps, the proposed method (Figure 19b) demonstrates substantial accuracy improvement.

Table 6 lists accuracy analysis for the proposed method for all five test sets. As in the 3D map analysis, we compare the RTK-GPS measured ground truth with aerial image and proposed method. Unlike the 3D structural map error, the SLAM error is not the only error source for this road marking error. When manually selecting a pixel from distorted and stitched road images, other factors related to image processing influences (e.g., image resolution, image quality, image distortion, manual pixel selection accuracy, and interpolated motion error). Lane parameterization and online conversion will surely improve overall lane results from fine refinement. In this work, however, we mainly aim to validate the SLAM result accuracy by examining error from structural and lane mapping results.

Table 6. Error analysis for the proposed method on road marking. Each dataset has two to six sample points respectively. For set 2, no road marking in the aerial map and excluded in the comparison. The proposed method's error is also written as a ratio to the aerial image for clear comparison.

Sample	No. of Lane Points	Average RMSE	
		Aerial Image [m]	Proposed Method [m]
set 1	2	7.180	1.000 (13.927%)
set 2	2	-	1.055 (- %)
set 3	2	8.948	1.699 (18.989%)
set 4	6	8.044	0.622 (7.727%)
set 5	2	11.261	0.610 (5.418%)

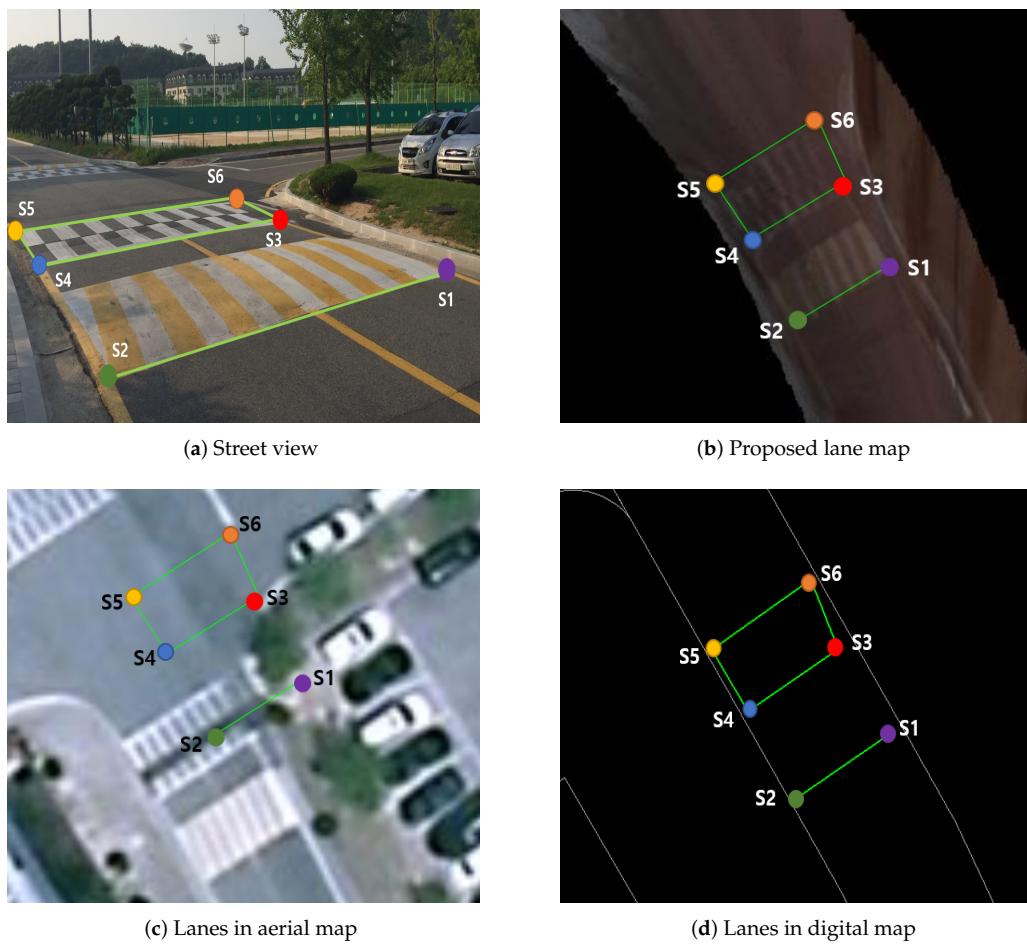


Figure 19. Six sample points having accurate global position are measured by RTK-GPS, and they are plotted on (a) street view, (b) lane map, (c) aerial map and (d) digital map. (c) Substantial position error with ground truth in aerial map, (d) Small position error with ground truth occurs, but lacking in detail such as road mark and cross walk. (b) Proposed method has position accuracy with global ground truth and detail information.

8. Conclusions

This paper proposes a seamless way to incorporate publically available digital maps and sensors mounted on a mobile platform. For accurate urban mapping, we focus on 3D structural and lane mapping. The proposed method tackles the accurate map building problem by integrating global measurements with local sensor measurements via Simultaneous Localization and Mapping (SLAM). The results validate the accuracy of the created map. As the ground truth is not available we have

conducted accuracy analysis by measuring a set of sample points using RTK-GPS. The proposed method has proven itself capable of balancing unreliable GPS and digital map information to create an accurate map.

Acknowledgments: This material is based upon work supported from ‘Development of real-time localization and 3-D mapping system with cm-level accuracy based on digital maps and vision data for autonomous driving’ project by the Ministry of Trade, Industry & Energy (MOTIE, Korea) under Industrial Technology Innovation Program (No. 10051867) and ‘3D Mapping for Intelligent Vehicle via Digital Map-Based SLAM’ project funded by Naver Corporation.

Author Contributions: Hyunchul Roh designed mobile mapping sensor system, developed digital map-based SLAM and performed data logging experiment. Jinyong Jeong modeled adaptive IPM and generated lane map using SLAM result. Younggun Cho analyzed initial lane map research and helped logging datasets. Ayoung Kim designed experiment and wrote the paper.

Conflicts of Interest: The authors declare no conflict of interest.

References

- Gu, Y.; Wada, Y.; Hsu, L.; Kamijo, S. Vehicle self-localization in urban canyon using 3D map based GPS positioning and vehicle sensors. In Proceedings of the IEEE International Conference on Conference on Connected Vehicle and Expo, Vienna, Austria, 3 November 2014; pp. 792–798.
- Stiller, C.; Ziegler, J. 3D perception and planning for self-driving and cooperative automobiles. In Proceedings of the IEEE International Multi-Conference on Systems, Signals and Devices, Chemnitz, Germany, 20 March 2012; pp. 1–7.
- Maddern, W.; Stewart, A.D.; Newman, P. LAPS-II: 6-DoF day and night visual localisation with prior 3D structure for autonomous road vehicles. In Proceedings of the IEEE Intelligent Vehicle Symposium, Dearborn, MI, USA, 8 June 2014; pp. 330–337.
- Pylvänäinen, T.; Berclaz, J.; Korah, T.; Hedau, V.; Aanjaneya, M.; Grzeszczuk, R. 3D city modeling from street-level data for augmented reality applications. In Proceedings of the IEEE International Conference on 3D Imaging, Modeling, Processing, Visualization and Transmission, Zurich, Switzerland, 13 October 2012; pp. 238–245.
- Rusu, R.B.; Cousins, S. 3D is here: Point cloud library (PCL). In Proceedings of the IEEE International Conference on Robotics and Automation, Shanghai, China, 9 May 2011; pp. 1–4.
- Golovinskiy, A.; Kim, V.G.; Funkhouser, T. Shape-based recognition of 3D point clouds in urban environments. In Proceedings of the IEEE International Conference on Computer Vision, Tokyo, Japan, 29 September 2009; pp. 2154–2161.
- Sheehan, M.; Harrison, A.; Newman, P. Continuous vehicle localisation using sparse 3D sensing, kernelised Rényi distance and fast Gauss transforms. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, Tokyo, Japan, 3 November 2013; pp. 398–405.
- Pascoe, G.; Maddern, W.; Newman, P. Direct Visual Localisation and Calibration for Road Vehicles in Changing City Environments. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 11 December 2015; pp. 9–16.
- Candiago, S.; Remondino, F.; De Giglio, M.; Dubbini, M.; Gattelli, M. Evaluating Multispectral Images and Vegetation Indices for Precision Farming Applications from UAV Images. *Remote Sens.* **2015**, *7*, 4026–4047.
- Liu, M. Robotic Online Path Planning on Point Cloud. *IEEE Syst. Man Cybern. Magn.* **2015**, *46*, 1217–1228.
- Sampath, A.; Shan, J. Segmentation and reconstruction of polyhedral building roofs from aerial LiDAR point clouds. *IEEE Trans. Geosci. Remote Sens.* **2010**, *48*, 1554–1567.
- Galvanin, E.A.d.S.; Poz, A.P.D. Extraction of building roof contours from LiDAR data using a Markov-Random-Field-based approach. *IEEE Trans. Geosci. Remote Sens.* **2012**, *50*, 981–987.
- You, R.J.; Lin, B.C. A quality prediction method for building model reconstruction using LiDAR data and topographic maps. *IEEE Trans. Geosci. Remote Sens.* **2011**, *49*, 3471–3480.
- Gonzalez-Aguilera, D.; Crespo-Matellan, E.; Hernandez-Lopez, D.; Rodriguez-Gonzalvez, P. Automated urban analysis based on LiDAR-derived building models. *IEEE Trans. Geosci. Remote Sens.* **2013**, *51*, 1844–1851.

15. Izadi, M.; Saeedi, P. Three-dimensional polygonal building model estimation from single satellite images. *IEEE Trans. Geosci. Remote Sens.* **2012**, *50*, 2254–2272.
16. Koc-San, D.; Turker, M. A model-based approach for automatic building database updating from high-resolution space imagery. *Int. J. Remote Sens.* **2012**, *33*, 4193–4218.
17. Zhou, G.; Zhou, X. Seamless fusion of LiDAR and aerial imagery for building extraction. *IEEE Trans. Geosci. Remote Sens.* **2014**, *52*, 7393–7407.
18. Blanco, J.L.; Moreno, F.A.; González-Jiménez, J. A Collection of Outdoor Robotic Datasets with centimeter-accuracy Ground Truth. *Autom. Rob.* **2009**, *27*, 327–351.
19. Blanco, J.L.; Moreno, F.A.; González-Jiménez, J. The Málaga Urban Dataset: High-rate stereo and LiDARs in a realistic urban scenario. *Int. J. Rob. Res.* **2014**, *33*, 207–214.
20. Elseberg, J.; Borrmann, D.; Nüchter, A. Algorithmic Solutions for Computing Precise Maximum Likelihood 3D Point Clouds from Mobile Laser Scanning Platforms. *Remote Sens.* **2013**, *5*, 5871–5906.
21. Bok, Y.; Choi, D.G.; Jeong, Y.; Kweon, I.S. Capturing city-level scenes with a synchronized camera-laser fusion sensor. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, San Francisco, CA, USA, 25 September 2011; pp. 4436–4441.
22. Riegl VMX-450. Available online: <http://www.riegl.com> (accessed on 17 August 2016).
23. Tuohy, S.; O’Cualain, D.; Jones, E.; Glavin, M. Distance determination for an automobile environment using inverse perspective mapping in OpenCV. In proceedings of Irish Signals and Systems Conference, Cork, Ireland, 23 June 2010.
24. Laganiere, R. Compositing a bird’s eye view mosaic. In Proceedings of the Vision Interface Conference, Montreal, Canada, 14 May 2000; pp. 382–387.
25. Lin, C.C.; Wang, M.S. A vision based top-view transformation model for a vehicle parking assistant. *Sensors* **2012**, *12*, 4431–4446.
26. Guo, C.; Meguro, J.i.; Kojima, Y.; Naito, T. Automatic lane-level map generation for advanced driver assistance systems using low-cost sensors. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Hong Kong, China, 31 May 2014; pp. 3975–3982.
27. Wolcott, R.W.; Eustice, R.M. Visual localization within LiDAR maps for automated urban driving. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, Chicago, IL, USA, 14 September 2014; pp. 176–183.
28. Tanner, M.; Pinies, P.; Paz, L.M.; Newman, P. DENSER Cities: A System for Dense Efficient Reconstructions of Cities. 2016, arXiv:1604.03734.
29. Schindler, A.; Maier, G.; Janda, F. Generation of high precision digital maps using circular arc splines. In Proceedings of the IEEE Intelligent Vehicle Symposium, Madrid, Spain, 3 June 2012; pp. 246–251.
30. Schindler, A. Vehicle self-localization with high-precision digital maps. In Proceedings of the IEEE Intelligent Vehicle Symposium, Gold Coast City, Australia, 23 June 2013; pp. 141–146.
31. Floros, G.; van der Zander, B.; Leibe, B. OpenStreetSLAM: Global vehicle localization using OpenStreetMaps. In Proceedings of the IEEE International conference on Robotics and Automation, Karlsruhe, German, 6 May 2013; pp. 1054–1059.
32. OpenStreetMap. Available online: <http://www.openstreetmap.org> (accessed on 17 August 2016).
33. Pink, O.; Stiller, C. Automated map generation from aerial images for precise vehicle localization. In Proceedings of the 13th International IEEE Conference on Intelligent Transportation Systems (ITSC), Madeira, Portugal, 19 September 2010; pp. 1517–1522.
34. Napier, A.; Newman, P. Generation and exploitation of local orthographic imagery for road vehicle localisation. In Proceedings of the IEEE Intelligent Vehicle Symposium, Madrid, Spain, 3 June 2012; pp. 590–596.
35. Schreiber, M.; Knoppel, C.; Franke, U. Laneloc: Lane marking based localization using highly accurate maps. In Proceedings of the IEEE Intelligent Vehicle Symposium, Gold Coast City, Australia, 23 June 2013; pp. 449–454.
36. Rose, C.; Britt, J.; Allen, J.; Bevly, D. An integrated vehicle navigation system utilizing lane-detection and lateral position estimation systems in difficult environments for GPS. *IEEE Trans. Intell. Trans. Syst.* **2014**, *15*, 2615–2629.
37. Larnaout, D.; Gay-Bellile, V.; Bourgeois, S.; Dhome, M. Fast and automatic city-scale environment modelling using hard and/or weak constrained bundle adjustments. *Mach. Vision App.* **2016**, *27*, 943–962.

38. Roh, H.C.; Oh, T.J.; Choe, Y.; Chung, M.J. Satellite map based quantitative analysis for 3D world modeling of urban environment. In Proceedings of the 9th Asian IEEE Control Conference (ASCC), Istanbul, Turkey, 23 June 2013; pp. 1–6.
39. Borenstein, J.; Feng, L. Measurement and correction of systematic odometry errors in mobile robots. *IEEE Trans. Robot. Automat.* **1996**, *12*, 869–880.
40. Kaess, M.; Johannsson, H.; Leonard, J. Open Source Implementation of iSAM. Available online: <http://people.csail.mit.edu/kaess/isam> (accessed on 17 August 2016).
41. Withrobot altimeter mypressure. Available online: <http://www.withrobot.com> (accessed on 17 August 2016).
42. IMU XsensMTI. Available online: <http://www.xsens.com> (accessed on 17 August 2016).
43. Sünderhauf, N.; Protzel, P. Switchable constraints for robust pose graph SLAM. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, Vilamoura-Algarve, Portugal, 7 October 2012; pp. 1879–1884.
44. Agarwal, P.; Tipaldi, G.D.; Spinello, L.; Stachniss, C.; Burgard, W. Robust map optimization using dynamic covariance scaling. In Proceedings of the IEEE International Conference on Robotics and Automation, Karlsruhe, German, 6 May 2013; pp. 62–69.
45. Sportouche, H.; Tupin, F.; Denise, L. Extraction and three-dimensional reconstruction of isolated buildings in urban scenes from high-resolution optical and SAR spaceborne images. *IEEE Trans. Geosci. Remote Sens.* **2011**, *49*, 3932–3946.
46. Gu, Y.; Wang, Q.; Jia, X.; Benediktsson, J.A. A novel MKL model of integrating LIDAR data and MSI for urban area classification. *IEEE Trans. Geosci. Remote Sens.* **2015**, *53*, 5312–5326.
47. Zhu, X.X.; Montazeri, S.; Gisinger, C.; Hanssen, R.F.; Bamler, R. Geodetic SAR tomography. *IEEE Trans. Geosci. Remote Sens.* **2016**, *54*, 18–35.
48. Brell, M.; Rogass, C.; Segl, K.; Bookhagen, B.; Guanter, L. Improving Sensor Fusion: A Parametric Method for the Geometric Coalignment of Airborne Hyperspectral and Lidar Data. *IEEE Trans. Geosci. Remote Sens.* **2016**, *54*, 3460–3474.
49. Environmental Systems Research Institute. Available online: <http://www.esri.com/> (accessed on 17 August 2016).
50. Ozog, P.; Carlevaris-Bianco, N.; Kim, A.; Eustice, R.M. Long-term Mapping Techniques for Ship Hull Inspection and Surveillance using an Autonomous Underwater Vehicle. *J. Field Robot.* **2015**, *33*, 265–289.
51. Mallot, H.A.; Bülthoff, H.H.; Little, J.; Bohrer, S. Inverse perspective mapping simplifies optical flow computation and obstacle detection. *Biol. Cybern.* **1991**, *64*, 177–185.
52. Jeong, J.; Kim, A. Adaptive Inverse Perspective Mapping for Lane Map Generation with SLAM. In Proceedings of the International Conference on Ubiquitous Robots and Ambient Intell, Xi'an, China, 9 August 2016.
53. Carrillo, H.; Reid, I.; Castellanos, J.A. On the comparison of uncertainty criteria for active SLAM. In Proceeding of IEEE International Conference on Robotics and Automation, Saint Paul, MN, USA, 14 May 2012; pp. 2080–2087.
54. Kiefer, J. General Equivalence Theory for Optimum Designs (Approximate Theory). *Ann. Stat.* **1974**, *2*, 849–879.
55. Kim, A.; Eustice, R.M. Active visual SLAM for robotic area coverage: Theory and experiment. *Int. J. Robot. Res.* **2015**, *34*, 457–475.
56. Topcon GRX2. Available online: <http://www.topcon.co.jp/> (accessed on 17 August 2016).



© 2016 by the authors; licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC-BY) license (<http://creativecommons.org/licenses/by/4.0/>).