Object-Detection-Aided GNSS and Its Integration With Lidar in Highly Urbanized Areas

Weisong Wen, Guohao Zhang, and Li-Ta Hsu

*Are with the Hong Kong Polytechnic University, China.
E-mail: weisong.wen@connect.polyu.hk; guohao.zhang@connect.polyu.hk;
lt.hsu@polyu.edu.hk*

Abstract—Positioning is a key function for autonomous vehicles that requires globally referenced localization information. *Lidar-based mapping*, which refers to simultaneous localization and mapping (SLAM), provides continuous positioning in diverse scenarios. However, SLAM error can accumulate through time. Besides, only relative positioning is provided by SLAM. The Global Navigation Satellite System (GNSS) receiver is one of the significant sensors for providing globally referenced localization, and it is usually integrated with lidar in autonomous driving. However, the performance of the GNSS is severely challenged due to the reflection and blockage caused by buildings in superurbanized cities, including Hong Kong, China; Tokyo; and New York, resulting in the notorious non-line-of-sight (NLOS) receptions. Moreover, the uncertainty of the GNSS positioning is ambiguous, leading to the incorrect tuning of its weight during GNSS–lidar integration. This article innovatively employs lidar to identify the NLOS measurement of the GNSS receiver using point-cloud-based object detection. Measurements from satellites suffering from NLOS reception will be excluded based on the proposed fault detection and exclusion (FDE) algorithm. Then, GNSS-weight least-square positioning is conducted based on the surviving measurements from FDE. The noise covariance of the GNSS positioning is calculated by considering the potential location errors caused by the NLOS and the remaining LOS measurements. The improved GNSS result and its corresponding noise covariance are integrated with lidar through a graph-based SLAM-integration framework. Experimental results indicate that the proposed GNSS–lidar integration can obtain improved positioning accuracy in a highly urbanized area in Hong Kong.

Autonomous vehicles [1], [2] receive increasing attention due to their immense potential market. To achieve fully autonomous vehicles, globally referenced, meter-level positioning is required in all scenarios. Lidar is a commonly used sensor for autonomous driving, not only for object detection [3], [4] but to provide continuous positioning [5]. In lidar-based positioning, the simultaneous localization and mapping (SLAM) [6], [7] algorithm is usually employed to calculate the transformation between the consecutive point clouds provided by lidar. However, the localization from SLAM can introduce accumulated error through time, and only relative positioning is obtained, which cannot satisfy autonomous vehicles' requirement. With the rise of the multi-Global Navigation Satellite System (GNSS), the availability of satellites has been significantly enhanced, which makes it possible to receive transmissions from enough satellites for the GNSS positioning, even in an urban canyon. The GNSS is currently a significant source that provides continuous global positioning. It is usually integrated with lidar-based localization to take advantage of both positioning sources [8]–[12].

Based on the principle of sensor fusion, sensor-integration methods can be divided into two groups: filtering based and smoothing based. The symbolic filtering-based sensor-integration method is the Bayes filter, including the Kalman filter [13], [14], information filter [15]–[17], and particle filter [18]–[20]. The Bayes filter-based sensor integration estimates the current state from the current observation and the previous state approximation, failing to make use of all the states before the previous ones. This is because of the assumption of the first order of the Markov model [21], which is one of the key suppositions of the Bayes filter. Conversely, the smoothing approaches [22]–[25] estimate the pose and the map by considering the full sets of measurements, from the first epoch to the current one. The most well-known smoothing method is graph-based SLAM [26].

These GNSS–lidar integration solutions can obtain decent positioning in sparse areas, as shown in Figure 1. However, their execution can be severely challenged in superurbanized areas due to the poor performance of, and a large uncertainty in, the GNSS positioning. The GNSS can achieve a positioning accuracy of 5~10 m in open areas and sparse scenarios based on conventional single-point positioning. However, the positioning error can significantly increase to ~50 m in superurbanized areas [27] because of the reflection and blockage from the surrounding buildings. The reflection can cause extra travel delays in the pseudorange domain, thus causing the well-known multipath effects and non-line-of-sight (NLOS) receptions. Moreover, the uncertainty of the GNSS positioning is also greatly increased by severe NLOS conditions.

According to a recent survey [28], the NLOS is the dominant component to blame for positioning errors in dense urban areas. Numerous studies [29]–[33] have been conducted to identify NLOS receptions. Due to the fact that the NLOS is caused by reflection from buildings, 3D city models are employed to identify NLOS receptions [34]–[39]. With the aid of the 3D city models, the possible blockage from buildings can be detected, and the corresponding NLOS is obtained. The NLOS measurements are subsequently excluded from the GNSS positioning. However, this method relies heavily on the availability of the 3D city models, which is the main problem for its implementation. Range-based, 3D-map-aided GNSS [29], [31]–[33] is one of the most mature techniques to mitigate the positioning errors from NLOS receptions. It innovatively employs the ray-tracing simulation to represent the possible transmission routes of the GNSS signals. Thus, the travel delay can be calculated based on the simulated signal transmission route. The NLOS measurement is also corrected and used in the GNSS-positioning calculation. However, this method introduces a heavy computational load due to the ray-tracing simulation. Moreover, 3D city models are also needed, and those map-aided GNSS-positioning methods rely heavily on the initial guess of the receiver.

The other problem in GNSS–lidar integration is the large uncertainty of the GNSS positioning in urbanized areas. The uncertainty is referred to as the *noise covariance*, which is essential in the GNSS–lidar integration. Satisfactory performance can be obtained using the GNSS–lidar integration scheme on the condition that each instance of

FIG 1 (a) The performance of GNSS stand-alone positioning. The red circles indicate the GNSS-positioning result, and the black circles represent the ground truth. (b)The GNSS–lidar integration, where green points signify the point cloud map and pink circles denote the trajectory generated by the GNSS–lidar integration. The background image of Figure 1(a) is captured from Google Earth.

sensor noise is well modeled. However, the research in [8]–[10] and [12] models the GNSS-positioning uncertainty as a Gaussian distribution. This rough modeling of the GNSS-positioning uncertainty can work in places where the GNSS positioning is robust and accurate with few NLOS receptions. However, the GNSS-positioning error is not subject to Gaussian distribution in urbanized areas [27]. Conventional constant, Gaussian distribution-based covariance cannot model the actual performance of the GNSS positioning. As a result, the GNSS–lidar integration result can introduce additional positioning errors.

A GNSS covariance-estimation solution based on satellite numbers and the signal-to-noise ratio (SNR) can obtain improvements [40] compared with the conventional constant covariance solution. However, this scheme cannot effectively model the positioning error from the NLOS. Taking advantage of the 3D lidar sensor, the 3D point cloud map is employed to identify the NLOS measurement [41]. This method can effectively detect the NLOS signals, and it is similar to the techniques in [35] and [42], where 3D city models are used. The NLOS receptions are then excluded from the GNSS positioning. The point cloud map plays a role similar to that of 3D city maps. However, the implementation of this method is subject to the availability of a fully 3D point cloud map of buildings, which is difficult to obtain. Moreover, the GNSS positioning uncertainty is simply estimated based on the new horizontal dilution of precision (HDOP), and the actual NLOS errors are not effectively modeled. Figure 2 presents a highly urbanized scenario in Hong Kong, where both sides of the street are lined with tall buildings. As a result, the GNSS receiver may obtain many NLOS measurements that contain only reflected signals.

This article innovatively employs 3D lidar to facilitate the GNSS positioning and the corresponding covariance estimation using real-time point-cloud-based object detection. Then, the improved GNSS positioning is integrated with lidar odometry in a graph-based SLAM framework. The flowchart of the GNSS–lidar integration solution is shown in Figure 3. First, the building boundary is detected by using the algorithm proposed in the previous work [43] of our research team. The point clouds are fixed to the GNSS frame based on the orientation obtained from lidar odometry (shown in Figure 3). The satellites and the building boundary are both projected to a GNSS skyplot [44]. Second, NLOS discovery is conducted with a proposed NLOS-detection algorithm. GNSS measurements that suffer from both the NLOS and a low elevation angle are excluded based on a proposed fault-detection and exclusion

(FDE) algorithm. Then the GNSS positioning is conducted using the surviving GNSS measurements. Third, the GNSS-positioning covariance is calculated by considering the potential location errors caused by NLOS and LOS receptions. Finally, the improved GNSS-positioning result and the corresponding covariance are integrated with the lidar odometry through a graph-based SLAM framework.

Lidar Odometry and Its Covariance Estimation

Lidar Odometry

The principle of lidar odometry [45] is to track the transformation between two successive frames of point clouds by matching the two frames (called a *reference point cloud* and an *input point cloud* in this article). The matching process is also called *point cloud registration*. The objective of point cloud registration is to obtain the optimal transformation matrix to match or align the reference and input point clouds. The most well-known method of point cloud registration is the iterative closest point (ICP) [46]. The ICP is a straightforward approach to calculate the transformation matrix between two consecutive scans by iteratively
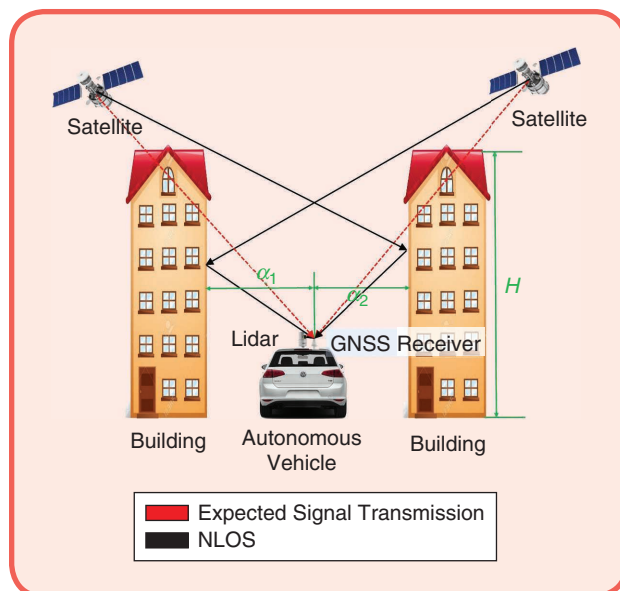

The other problem in GNSS–lidar integration is the large uncertainty of the GNSS positioning in urbanized areas. The uncertainty is referred to as the *noise covariance*, which is essential in the GNSS–lidar integration.

FIG 2 GNSS NLOS receptions caused by surrounding buildings.

searching pairs of nearby points in the two scans and minimizing the sum of all point-to-point distances. The objective function can be expressed as follows [46]:

$$C(\hat{\boldsymbol{R}}, \hat{\boldsymbol{T}}) = \arg\min \sum_{i=1}^{N} ||(\boldsymbol{R}\boldsymbol{p}_i + \boldsymbol{T}) - \boldsymbol{q}_i||^2, \qquad (1)$$

where N indicates the number of points in one scan $\boldsymbol{p}$, and $\boldsymbol{R}$ and $\boldsymbol{T}$ designate the rotation and translation matrices, respectively, to transform the input point cloud ($\boldsymbol{p}$) into the reference point cloud ($\boldsymbol{q}$). Objective function $C(\hat{\boldsymbol{R}}, \hat{\boldsymbol{T}})$ stands for the error of the transformation. The main drawback to this method is that the ICP can easily get into the local minimum problem. The NDT [47] is a state-of-art method to align two consecutive scans with models of points based on a Gaussian distribution. The NDT innovatively divides the point cloud space into cells. Each cell is continuously modeled by a Gaussian distribution. In this case, the discrete point clouds are transformed into successive continuous functions. In this article, the NDT is employed as the point cloud registration method for lidar odometry. Assume that the transformation between two consecutive frames of point clouds can be expressed as $\overrightarrow{\boldsymbol{pose}_6} = [t_x \; t_y \; t_z \; \phi_x \; \phi_y \; \phi_z]^T$. Here, t_i indicates the translation on the x-, y-, and z-axis, respectively, and ϕ_x represents the orientation angle of the

roll, pitch, and yaw, respectively. The steps to calculate the relative pose between the reference and input point clouds are as follows:

1) Fetch all the points $\boldsymbol{x}_{i=1...n}$ contained in a 3D cell [48]. Calculate the geometry mean $\boldsymbol{q} = (1/n)\Sigma_i \boldsymbol{x}_i$. Calculate the covariance matrix

$$\Sigma = \frac{1}{n}\sum_i (\boldsymbol{x}_i - \boldsymbol{q})(\boldsymbol{x}_i - \boldsymbol{q})^T. \qquad (2)$$

2) The matching score is modeled as

$$f(\boldsymbol{p}) = -\text{score}(\boldsymbol{p}) = \sum_i \exp\left(-\frac{(\boldsymbol{x}_i' - \boldsymbol{q}_i)^T \Sigma_i^{-1}(\boldsymbol{x}_i' - \boldsymbol{q}_i)}{2}\right), (3)$$

where $\boldsymbol{x}_i$ indicates the points in the current frame of scan $\boldsymbol{p}$, $\boldsymbol{x}_i'$ denotes the point in the previous scan that is mapped from the current frame using $\overrightarrow{pose_6}$, and $\boldsymbol{q}_i$ and Σ_i indicate the mean and the covariance of the corresponding normal distribution to point $\boldsymbol{x}_i'$ in the NDT of the previous scan.

3) Update the pose using the quasi-Newton method based on the objective function to minimize the score $f(\boldsymbol{p})$.

With all points in one frame of point clouds being modeled as cells, the objective of the optimization for the NDT
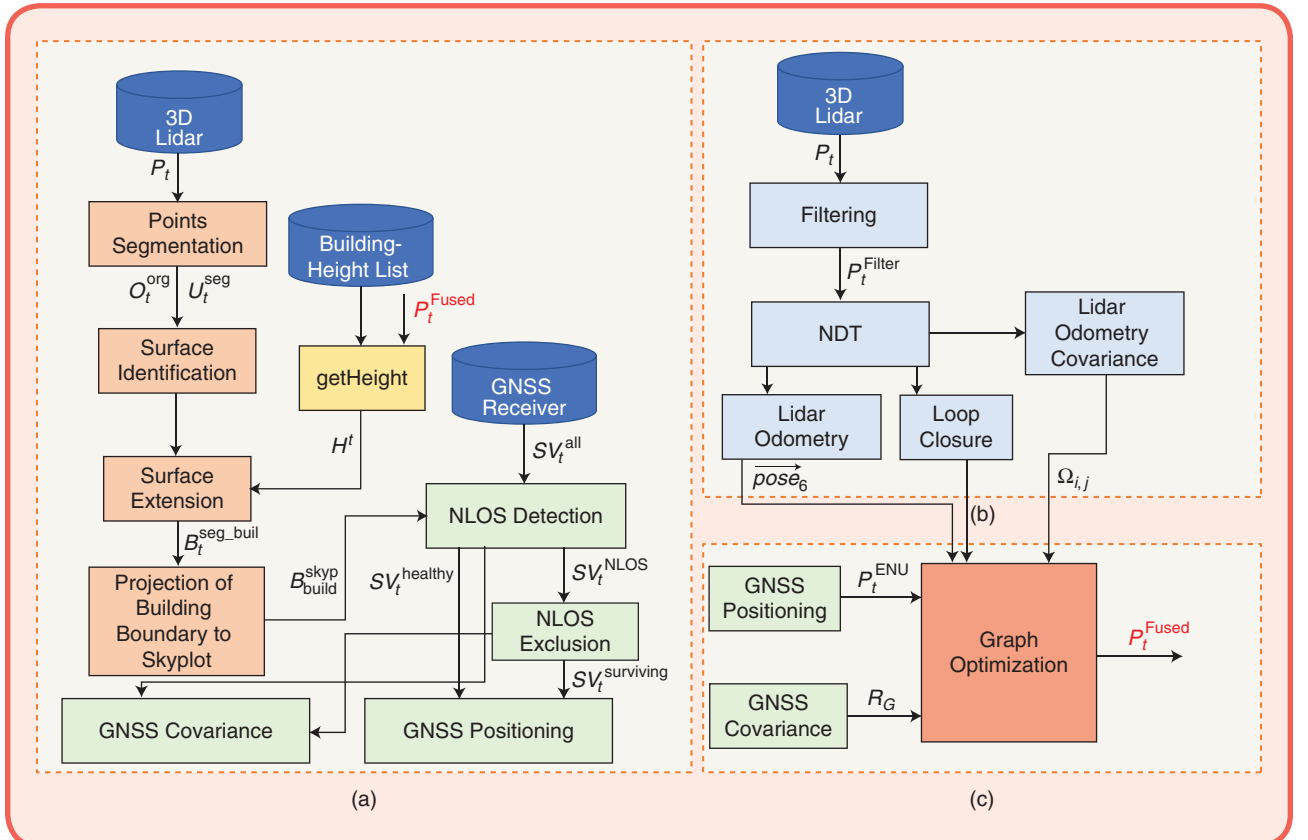

FIG 3 The proposed GNSS–lidar integration method. Three parts are included: (a) GNSS positioning and its covariance estimation, (b) lidar odometry and its covariance estimation, and (c) graph-based optimization. NDT: normal distribution transform.

is to match the current cells to the previous cells with the highest probability. The optimization function $f(\boldsymbol{p})$ can be found in [47]. In each cell that contains several points, the corresponding covariance matrix can be calculated and represented by $\boldsymbol{\Sigma}$. The shape of the cell (circle, plane, or linear) is indicated by the relationship among the three eigenvalues of the covariance matrix [47]. In this case, compared with the conventional ICP algorithm, the NDT innovatively optimizes the transformation by considering the features of the points. The loop-closure detection is conducted based on these shape features [49].

Covariance Estimation of Lidar Odometry
Lidar odometry can provide a continuous relative pose estimation, $\overrightarrow{\boldsymbol{pose}_6}$. The associated covariance of this pose estimate is essential for the later integration with the GNSS positioning. During the NDT process, the covariance of the pose estimation is related to the uncertainty of the matching between the reference and the input point clouds. In graph-based optimization, which will be introduced in the "Graph-Based GNSS–Lidar Integration" section, the covariance is indicated as the inverse of the information matrix $\boldsymbol{\Omega}_{ij}$. In each matching process between a point from the reference point cloud and a point from the input point cloud, we model the degree of matching as

$$dm_{ij} = \frac{1}{n} \sum_{k=1}^{n} \sqrt{\Delta x_k^2 + \Delta y_k^2 + \Delta z_k^2}, \qquad (4)$$

where dm_{ij} represents the degree of matching between the reference and input point clouds, n denotes the number of points in the input point cloud, Δx_k indicates the positional difference on the x-axis between input and reference points after the convergence of NDT is obtained, and Δy_k and Δz_k indicate the positional differences on the y- and z-axis, respectively. Thus, the information matrix $\boldsymbol{\Omega}_{ij}$ of the degree of matching between the reference and the input can be expressed as

$$\boldsymbol{\Omega}_{ij} = \begin{bmatrix} \boldsymbol{\Omega}_{ij}^p & 0 \\ 0 & \boldsymbol{\Omega}_{ij}^r \end{bmatrix}, \qquad (5)$$

$$\boldsymbol{\Omega}_{ij}^p = \boldsymbol{I}/(C_p^2 dm_{ij}), \qquad (6)$$

$$\boldsymbol{\Omega}_{ij}^r = \boldsymbol{I}/(C_r^2 dm_{ij}), \qquad (7)$$

where $\boldsymbol{I}$ indicates the identity matrix and C_p^2 is a coefficient that is heuristically determined. In this case, the covariance for lidar odometry is correlated with the degree of matching. We can see from Figure 2 that lidar matching can obtain a relative pose estimate on the lateral direction crossing the building. However, the longitudinal pose estimate is not as accurate because the building surface tends to be plain and featureless. Thus, the covariance should be adaptively changed according to the degree of matching in different scenarios.

Object-Detection-Aided GNSS Positioning and Its Covariance Estimation
In this section, the detection of a building boundary is first presented. The NLOS FDE method is subsequently discussed. Then, the GNSS positioning is implemented based on the NLOS FDE. Finally, the innovative covariance estimation of the GNSS positioning is introduced.

Building-Boundary Detection
To identify which satellite is blocked by the surrounding buildings, the pose of the building boundaries relative to the GNSS receiver is needed. Since 3D lidar can provide sufficient points that represent the environment, our previous work in [43] presents the detection of a double-decker bus and a dimension-extension algorithm founded on lidar-based object detection. Building-boundary detection is based on a similar approach, described as follows:
1) Segment the point clouds into clusters to represent different objects.
2) Identify the building surface and extend its dimensions to the actual measurements by using Algorithm 1.
3) Obtain the bounding box that indicates the building surface, and find the corresponding top boundary.
4) Calculate the pose of the building relative to the GNSS receiver.

The inputs of Algorithm 1 are $\boldsymbol{U}_t^{\mathrm{seg}}$ and $\boldsymbol{O}_t^{\mathrm{org}}$, which are obtained from the segmentation based on the work in [43]; the point number threshold num_{thres}; the length threshold len_{thres} and height threshold hei_{thres}; the building-height

---

Algorithm 1. Building-surface identification from bounding box sets and height extension.

Input: Bounding box sets $\boldsymbol{U}_t^{\mathrm{seg}} = \{\boldsymbol{U}_1, \boldsymbol{U}_2, \ldots, \boldsymbol{U}_i, \ldots \boldsymbol{U}_n, t\}$, organized point clusters $\boldsymbol{O}_t^{\mathrm{org}} = \{\boldsymbol{O}_1, \boldsymbol{O}_2, \ldots, \boldsymbol{O}_i, \ldots \boldsymbol{O}_n, t\}$, point number threshold num_{thres}, length threshold len_{thres}, height threshold hei_{thres}, building-height list $\boldsymbol{H}_{\mathrm{build}}$, receiver position $\boldsymbol{P}_t^{\mathrm{fused}}$, and yaw angle Yaw_r.

Output: Bounding box set represents building surfaces $\boldsymbol{B}_t^{\mathrm{seg\_buil}} = \{\boldsymbol{B}_1, \boldsymbol{B}_2, \ldots, \boldsymbol{B}_i, \ldots \boldsymbol{B}_n, t\}$.

1   set up an empty-clusters list $\boldsymbol{B}_t^{\mathrm{seg\_buil}}$ to save to the bounding box
2   **for** all bounding boxes $\boldsymbol{U}_i$ in $\boldsymbol{U}_t^{\mathrm{seg}}$ do
3   **if** $\mathrm{Num}(O_i) > num_{\mathrm{thres}}$
4         $\boldsymbol{U}_i \leftarrow [x_i^c, y_i^c, z_i^c, roll_i^c, pitch_i^c, yaw_i^c, d_i^{\mathrm{len}}, d_i^{\mathrm{wid}}, d_i^{\mathrm{hei}}]$
5   **if** $d_i^{\mathrm{len}} > len_{\mathrm{thres}}$ **and** $d_i^{\mathrm{hei}} > hei_{\mathrm{thres}}$
6           $d_i^{\mathrm{hei}} \leftarrow getHeight(\boldsymbol{H}_{\mathrm{build}}, \boldsymbol{P}_t^{\mathrm{fused}}, \boldsymbol{U}_i, Yaw_r)$
7           $\boldsymbol{B}_i \leftarrow \boldsymbol{U}_i$
8   **end if**
9   **end if**
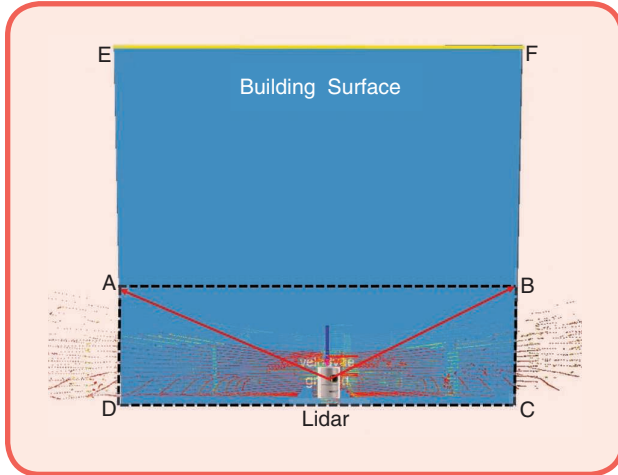10   **end for** $\boldsymbol{U}_t^{\mathrm{seg}}$

FIG 4 The point sets segmentation and building surface identification. Blue box ABCD represents the initially detected building surface. Blue box CDEF signifies the extended building surface.
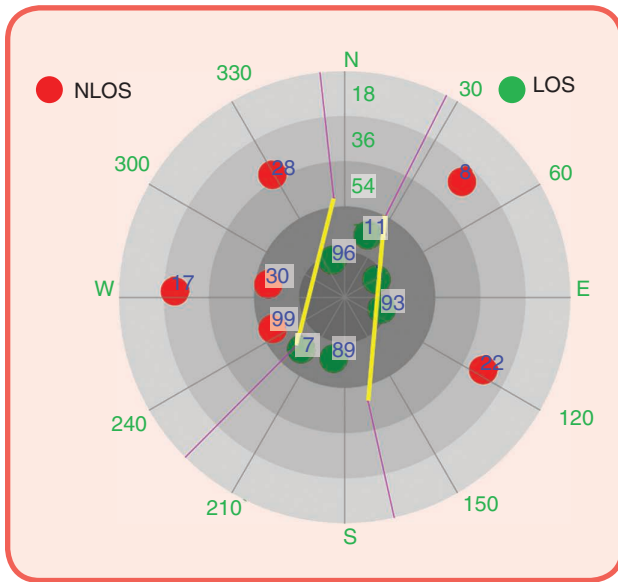

FIG 5 The GNSS skyplot indicates the satellite distribution and the building boundary.

list H_{build}; the receiver position P_t^{fused}; and the yaw angle Yaw_r from the GNSS–lidar integration. The output is the bounding box set $B_t^{\text{seg\_buil}}$, which specifically represents the building surface. Each bounding box is indicated by $U_i = [x_i^c, y_i^c, z_i^c, roll_i^c, pitch_i^c, yaw_i^c, d_i^{\text{len}}, d_i^{\text{wid}}, d_i^{\text{hei}}]$. The function *Num* mentioned in Algorithm 1 is used to calculate the points number of each cluster O_i. The function *getHeight* is employed to search the information in a saved building-height list that contains rough data. To determine the actual height of the identified building surface, P_t^{fused}, U_i, and Yaw_r are also needed. Here, P_t^{fused} indicates the GNSS position given by the previous epoch-positioning result, which was obtained from the GNSS–lidar integration.

The relative position between the GNSS receiver and the detected building can be obtained from U_i. Moreover, the yaw angle can be acquired from the GNSS–lidar integration. For each bounding box B_i, the distance α_i from the receiver to the detected building surface can be calculated as follows:

$$\alpha_i = \sqrt{(x_i^{c^2} + y_i^{c^2} + z_i^{c^2})}. \tag{8}$$

Thus, the bounding box with an extended height that represents the building surface can be identified using Algorithm 1. The bounding box is extended from rectangle ABCD to CDEF, as seen in Figure 4. Bounding boxes ABCD and CDEF indicate the initially detected dimensions and the extended dimensions of the building, respectively. Then, the boundary parameters for bounding box B_i, which corresponds to the building surface, are indicated by the line segment $\overline{EF}$, which is the bus-boundary matrix and denoted B_{build}^{3d}. To represent the building, two points, E and F, are required. The B_{build}^{3d}, which is relative to the lidar coordinate system, is structured as follows:

$$B_{\text{build}}^{3d} = \begin{bmatrix} x_{3dE} & y_{3dE} & z_{3dE} \\ x_{3dF} & y_{3dF} & z_{3dF} \end{bmatrix}. \tag{9}$$

In this case, the top boundary of the building is detected, and it will be used for the NLOS detection in the following section. The distance between the receiver and the building surface is calculated as α_i, which will be used for the covariance estimation of the GNSS positioning in the "GNSS-Positioning Covariance Estimation" section.

NLOS Detection and Exclusion

The boundary of the building is detected as B_{build}^{3d}. The satellites and the building boundary can be projected into a GNSS skyplot, which is shown in Figure 5. The circles indicate the satellites, and the associated number represents the satellite index. The yellow line describes the building boundary projected into the skyplot. The NLOS is indicated with a red circle in Figure 5. Assume that the initial satellite set is $SV_t^{\text{all}} = \{SV_1, SV_2, \ldots, SV_i, \ldots SV_s\}$, where $SV_i = \{az_i, el_i, SNR_i, \rho_i\}$; az_i and el_i denote the azimuth and elevation angles of a satellite, respectively; SNR_i indicates the satellite SNR; and ρ_i denotes the pseudorange measurement. The satellite visibility classification based on the satellite information and boundaries is introduced in the previous work [50] of our research team. According to Figure 5, we can have two satellite sets. One is $SV_t^{\text{nlos}} = \{SV_1, SV_2, \ldots, SV_i, \ldots SV_n\}$, which contains only the NLOS satellites. The other is the LOS satellite set $SV_t^{\text{los}} = \{SV_1, SV_2, \ldots, SV_i, \ldots SV_l\}$, and $s = n + l$ is satisfied.

We can see from the skyplot in Figure 5 that half the satellites are blocked (six out of 12). Essentially, only the satellites with an elevation angle of more than 72° are not blocked. The exclusion of all NLOS receptions can

result in a significant increase in the HDOP, which will magnify the pseudorange errors in the GNSS positioning. In other words, full exclusion of the NLOS receptions will conversely deteriorate the GNSS-positioning result. In previous work, we analyzed NLOS errors in [27]. The pseudorange error is smaller, while the elevation angle is higher, and the distance from the GNSS receiver to the reflector is smaller (refer to α_i in Figure 2). In other words, an NLOS with a lower elevation angle can introduce a larger GNSS-positioning error. The relationships between the satellite elevation, α_i and the pseudorange error are presented in [27].

Inspired by this result [51], this article proposes to exclude the measurement based on the satellite elevation angle and the HDOP of the satellite distribution. The proposed satellite-exclusion method is shown in Algorithm 2. The inputs of the algorithm include the NLOS satellite information sets $SV_t^{\mathrm{nlos}} = \{SV_1, SV_2, \ldots, SV_i, \ldots SV_n\}$ and SV_t^{los}. Only the satellites blocked by buildings are contained in satellite set SV_t^{nlos} (for example, satellites 8, 17, 28, 22, 30, and 39 shown in Figure 5). The thresholds for the elevation angle and the HDOP are also inputs to Algorithm 2. The output of Algorithm 2 is the satellite set that remains after the NLOS-exclusion process, indicated by $SV_t^{\mathrm{los\_nlos}} = \{SV_1^s, SV_2^s, \ldots, SV_i^s, \ldots SV_m^s\}$. After the proposed NLOS exclusion, some of the NLOS measurements that have low elevation angles are omitted. The surviving NLOS and LOS measurements are saved in $SV_t^{\mathrm{los\_nlos}}$. This satellite set is incorporated in the GNSS positioning through the weighted least-squares (WLS) method that is discussed in the following section.

GNSS WLS Positioning

Satellites in $SV_t^{\mathrm{los\_nlos}}$ are employed for the GNSS-positioning calculation. The system clock bias between the GNSS receiver and the satellites is contained in the pseudorange measurements. The equation linking the receiver position and the range measurements can be structured as the following least-squares method:

$$\hat{x} = (G^T G)^{-1} G^T \rho, \tag{10}$$

where G represents the observation matrix and is structured by unit LOS vectors between the GNSS receiver's position and the satellite's position. Here, $\hat{x}$ indicates the estimated receiver position, and ρ denotes the pseudorange measurements.

Conventionally, to better represent the reliability of each measurement based on the information from the receiver, weightings of each computation are needed. The function to calculate the weighting by integrating the measurement SNR and the satellite elevation angle is expressed as W [52]. Finally, the GNSS receiver position can be estimated using the WLS method:

> **Algorithm 2. The proposed NLOS exclusion.**
>
> **Input:** Satellite information set $SV_t^{\mathrm{nlos}} = \{SV_1, SV_2, \ldots, SV_i, \ldots SV_n\}$, $SV_t^{\mathrm{los}} = \{SV_1, SV_2, \ldots, SV_i, \ldots SV_l\}$, elevation angle threshold $el_{\mathrm{threshold}}$, HDOP threshold H_{thres}.
>
> **Output:** Corrected satellite set after NLOS exclusion: $SV_t^{\mathrm{los\_nlos}} = \{SV_1^s, SV_2^s, \ldots, SV_i^s, \ldots SV_m^s\}$.
>
> Step 1: sort the satellites set in SV_t^{nlos} based on the elevation angle from small to large.
>
> Step 2: exclude satellite SV_i from SV_t^{nlos} **if**:
> - its elevation angle is smaller than $el_{\mathrm{threshold}}$
> - the HDOP of the remaining satellites (including the remaining satellites in SV_t^{nlos} and the satellites in SV_t^{los}) is smaller than the HDOP threshold H_{thres}.
>
> Step 3: repeat step 2 until all the conditions in step 2 cannot be fully satisfied.
>
> Step 4: save the remaining satellites in SV_t^{nlos} and SV_t^{los} to $SV_t^{\mathrm{los\_nlos}} = \{SV_t^{\mathrm{nlos}}, SV_t^{\mathrm{los}}\}$.

$$\hat{x} = (G^T W G)^{-1} G^T W \rho. \tag{11}$$

In this article, the GNSS-positioning result is represented as $P_t^{\mathrm{ENU}} = [x_E \ y_N \ z_u]$ in the East, North, Up (ENU) coordinate system [53].

GNSS-Positioning Covariance Estimation

The GNSS uncertainty is usually modeled by considering the SNR, satellite numbers, and HDOP if the NLOS satellites are not identified [40]. This rough modeling can work only in open-sky environments where there are few NLOS receptions. A team from the University of Illinois [42] proposed to model the GNSS-positioning uncertainty based solely on the SNR [42] after identifying and excluding the NLOS receptions. However, full NLOS exclusion is not acceptable in superurbanized areas, as it can result in a significant increase in the HDOP. In this article, we propose to model the covariance matrix of the GNSS positioning, which consists of two parts, the NLOS and the LOS, as follows:

$$R_G = R_C + R_E. \tag{12}$$

Here, R_C is the covariance matrix that indicates the uncertainty of the GNSS positioning, with the assumption that all satellites used for positioning in $SV_t^{\mathrm{los\_nlos}}$ are LOS; it is calculated as follows:

$$R_C = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} HDOP_{xy} \sigma_{\mathrm{UERE}}^2, \tag{13}$$

where σ_{UERE} indicates the user-equivalent range error, which is experimentally determined in this article, and $HDOP_{xy}$ is the HDOP of the GNSS positioning.

In addition, $\boldsymbol{R}_E$ is the covariance matrix that indicates the extra uncertainty of the GNSS positioning that is caused by the NLOS. It is calculated as follows:

$$\boldsymbol{R}_E = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} HDOP_{xy}\,\sigma_{\mathrm{NLOS}}^2, \qquad (14)$$

where σ_{NLOS} indicates the extra uncertainty caused by the NLOS receptions. According to [27], the pseudorange error for each NLOS measurement can be roughly modeled as follows:

$$\gamma = \alpha(\sec\theta_{\mathrm{elc}}(1 + \cos 2\theta_{\mathrm{elc}}) + \sec\theta_{\mathrm{azm}}(1 + \cos 2\theta_{\mathrm{azm}})), \;(15)$$

where α represents the distance between the GNSS receiver and the reflector and is obtained from the surface detection presented in the "Building-Boundary Detection" section. Here, θ_{elc} and θ_{azm} represent the elevation and the azimuth angles, respectively. Thus, we can obtain the total uncertainty of pseudorange σ_{NLOS} for all satellites (totaling k satellites) as follows:

$$\sigma_{\mathrm{NLOS}} = \sum_{i=1}^{k} \gamma_i. \qquad (16)$$

In this case, the GNSS-positioning covariance is calculated by considering both the LOS and the NLOS measurements. The component that must be estimated in the matrix $\boldsymbol{R}_G$ is

$$\bar{\boldsymbol{R}} = HDOP_{xy}\,\sigma_{\mathrm{NLOS}}^2 + HDOP_{xy}\,\sigma_{\mathrm{UERE}}^2. \qquad (17)$$

Moreover, only the covariance in the horizontal direction is obtained. In superurbanized areas, the vertical dilution of precision is significantly larger than the HDOP. The positioning 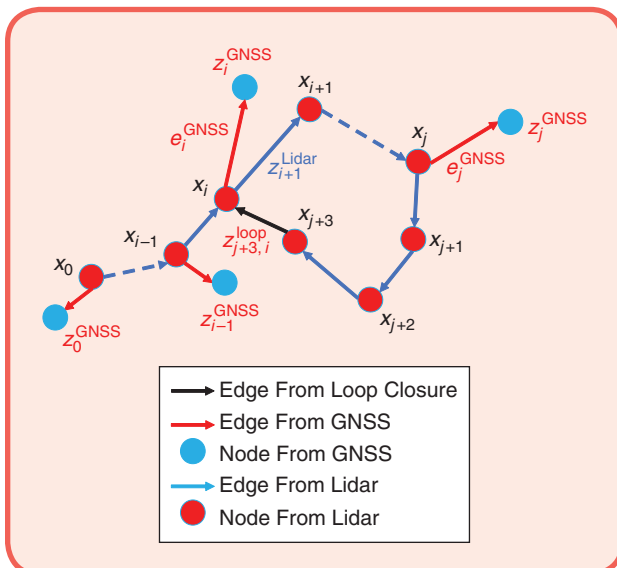error in the vertical direction can be very bad due to the distorted vertical distribution of the satellites. Thus, only the horizontal GNSS positioning and the corresponding covariance are used in the proposed GNSS–lidar integration.

Graph-Based GNSS–Lidar Integration

This section presents graph-based GNSS–lidar integration. The purpose of the pose graph optimization is to construct all measurements into a graph as constraints and calculate the best set of poses by solving a nonlinear optimization problem. In this article, the constraints are provided by both the object-detection-aided GNSS positioning and the lidar odometry. Two steps are needed to implement the graph-based GNSS–lidar integration optimization: graph generation and graph optimization.

Graph Generation

The graph is constituted by edges and vertexes [26]. Edges are provided by the observation measurements, including the GNSS and the lidar, as shown in Figure 6. Here, $\boldsymbol{x}_i$ represents the 6D pose estimation that includes the position and the orientation, $\boldsymbol{e}_{ij}$ indicates the error function that evaluates the difference between the estimated state and the observation from the sensors, and $\boldsymbol{z}_{ij}$ represents the observation. In the graph optimization, $\boldsymbol{x}_i$ is the state. The observations include three parts: the measurements from the GNSS, the loop closure, and the lidar positioning presented in the "Lidar Odometry and Its Covariance Estimation" and "Object-Detection-Aided GNSS Positioning and Its Covariance Estimation" sections, respectively. Blue circles and red lines represent the nodes and the edges, respectively, which are provided by the globally referenced GNSS positioning. Red circles and blue lines indicate the nodes and the edges respectively, which are provided by lidar odometry. The black line indicates the edge provided by the loop closure. The error function for GNSS observation is expressed as follows:

$$\boldsymbol{e}_i^{\mathrm{GNSS}} = ||\,h_i(\boldsymbol{x}_i) - \boldsymbol{z}_i^{\mathrm{GNSS}}\,||_{\boldsymbol{\Omega}}^2, \qquad (18)$$

where $h_i(*)$ is the measurement function that relates the GNSS measurement $\boldsymbol{z}_i^{\mathrm{GNSS}}$ to the state $\boldsymbol{x}_i$, and $\boldsymbol{\Omega}$ is the covariance matrix of the corresponding observation measurement.

Lidar odometry can provide a continuous 6D pose estimate and the corresponding covariance. The error function for lidar odometry is expressed as

$$\boldsymbol{e}_i^{\mathrm{LiDAR}} = ||\,\boldsymbol{x}_i - \boldsymbol{x}_{i-1} - \boldsymbol{z}_i^{\mathrm{LiDAR}}\,||_{\boldsymbol{\Omega}}^2, \qquad (19)$$

where the $\boldsymbol{z}_i^{\mathrm{LiDAR}}$ is the measurement from lidar odometry. The loop closure can be detected when the vehicle passes a similar or neighboring area again. The error function for the loop closure is expressed as

FIG 6 A demonstration of graph generation based on GNSS and lidar positioning.

$$e_{i,j}^{\text{loop}} = ||\boldsymbol{x}_i - \boldsymbol{x}_j - \boldsymbol{z}_i^{\text{loop}}||_{\boldsymbol{\Omega}}^2, \qquad (20)$$

where the $\boldsymbol{z}_i^{\text{loop}}$ is the measurement from the loop closure.

Only 2D horizontal positioning and the corresponding covariance are provided by the GNSS positioning in this article. The GNSS-positioning error can go up to –50 m in superurbanized areas. Since the covariance of the GNSS positioning is reasonably estimated by considering the NLOS and LOS receptions, we propose to add the GNSS results to the graph (shown in Figure 6) only when the $\sqrt{\bar{R}}$ of the GNSS positioning is smaller than a threshold $\boldsymbol{R}_{\text{threshold}}$. In this case, only the GNSS measurement that has a small covariance is applied to the graph optimization for providing the globally referenced update. This exclusion can prevent the severely biased GNSS-positioning result from being applied to the graph. The graph generation is shown in Algorithm 3.

Compared with conventional graph-based GNSS–lidar integration, this article innovatively adds the improved GNSS results and the corresponding covariance to the graph optimization. The effectiveness of this novelty is subject to the performance of the uncertainty estimation of the GNSS positioning, which is introduced in the "GNSS-Positioning Covariance Estimation" section.

Graph Optimization

Graph optimization [54] is a straightforward procedure that combines all constraints into a nonlinear optimization problem. The optimization form is

$$F(\boldsymbol{x}) = \sum_{i,j} ||h_i(\boldsymbol{x}_i) - \boldsymbol{z}_i^{\text{GNSS}}||_{\boldsymbol{\Omega}}^2 + ||\boldsymbol{x}_i - \boldsymbol{x}_{i-1} - \boldsymbol{z}_i^{\text{LiDAR}}||_{\boldsymbol{\Omega}}^2$$
$$+ ||\boldsymbol{x}_i - \boldsymbol{x}_j - \boldsymbol{z}_i^{\text{loop}}||_{\boldsymbol{\Omega}}^2, \qquad (21)$$

where $F(\boldsymbol{x})$ is the optimization function, which is the sum of the errors of all edges, and $\boldsymbol{\Omega}_{ij}$ is the information matrix that indicates the importance of each constraint in the global graph optimization. The information matrix is the inverse of the covariance matrix. The final solution to this optimization is $\boldsymbol{x}^*$ (the 6D pose estimate), which satisfies the following function:

$$\boldsymbol{x}^* = \arg\min F(\boldsymbol{x}). \qquad (22)$$

Thus, the optimization lies in solving the preceding equation to obtain the optimal $\boldsymbol{x}^*$. We can see from the optimization form $F(\boldsymbol{x})$ that the covariance of the GNSS and the lidar odometry positioning results are reflected in $\boldsymbol{\Omega}$. If the covariance of each positioning result is not properly estimated, the global optimization will be deflected, resulting in erroneous final pose sets.

Experimental Evaluation

To evaluate the performance of the GNSS–lidar integration method proposed in this article, two experiments are presented. The performance of stand-alone lidar positioning

---

Algorithm 3. The proposed graph generation.

Input: GNSS results $\boldsymbol{z}_i^{\text{GNSS}}$ and the corresponding covariance $\bar{\boldsymbol{R}}_i$. Lidar odometry observation $\boldsymbol{z}_i^{\text{lidar}}$ and loop closure $\boldsymbol{z}_i^{\text{loop}}$.
Output: Graph of nodes and vertexes.

Step 1: Initialize the estimated state using the GNSS results.
Step 2: Add the observation $\boldsymbol{z}_i^{\text{lidar}}$ from the lidar odometry to the graph if any of the following conditions are satisfied:
- the translation between the current lidar odometry and the previous node in the graph overweigh $Tran_{\text{threshold}}$.
- the rotation between the current lidar odometry and the previous node in the graph overweigh $Rot_{\text{threshold}}$.

Step 3: Add the observation of the GNSS results to the graph if
- the value $\sqrt{\bar{\boldsymbol{R}}_i}$ is smaller than $\boldsymbol{R}_{\text{threshold}}$.

Step 4: Add the observation $\boldsymbol{z}_i^{\text{loop}}$ from the loop closure to the graph if the loop closure is detected.
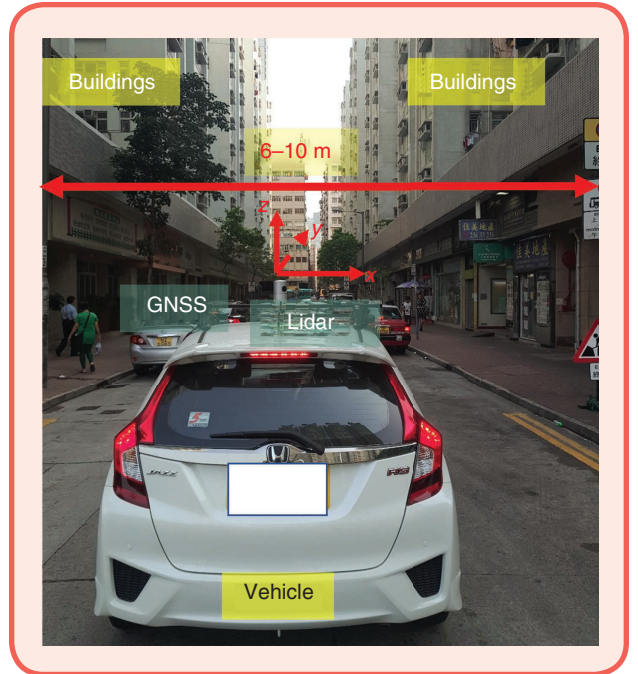**Step 5:** Repeat steps 2 and 3 until the end.

FIG 7 GNSS and lidar sensors are installed on the top of the vehicle.

in diverse urban scenarios is extensively evaluated in previous work [55] by our research team. In this section, GNSS positioning results are presented first. Then, the GNSS-lidar integration experiment results are analyzed.

Experimental Setup

Two experiments are conducted in Hong Kong. The first is implemented in a narrow street with buildings on both sides, which can be seen in Figure 7. Both sides of the road are lined with buildings, and the distance between the structures is just 7~10 m (see Figure 7). The experiment is conducted with an open-loop route. The other trial

> The GNSS positioning is evaluated by comparing the WLS-based GNSS positioning with the GNSS positioning aided by NLOS exclusion.

is conducted with a closed-loop route, and loop closure is available for the GNSS–lidar integration. The objective of this closed drive is to certify that the proposed GNSS–lidar integration solution is repeatable in diverse scenarios and that the overall performance can be well enhanced with the aid of loop closure.

In both experiments, a u-blox M8T receiver is used to collect raw GPS and BeiDou measurements. A 3D lidar sensor (Velodyne 32) is employed to provide the real-time point clouds scanned from the surroundings. Both the u-blox receiver and the 3D lidar are installed on the top of a vehicle, which can be seen in Figure 7, during the experiment. The data were collected during an approximately 5-min drive, at a frequency of 1 Hz for the GNSS and 10 Hz for the 3D lidar, using Robot Operation System [56] time to synchronize all the sensor information. The sensor setup and the corresponding coordinate system are shown in Figure 7, with the x-axis (the lidar coordinate system) pointing to the back of the vehicle. The GNSS positioning is represented through the ENU reference system. The initial position of the experiment is employed as the initial position calculation of the ENU coordinates. Moreover, the lidar coordinates are shown in Figure 7 and calibrated [55] with the GNSS in the ENU [57] coordinates at the beginning of the experiment. In addition, a NovAtel SPAN-CPT integrated GNSS–inertial navigation system (fiber optic gyroscopes) is used to provide the ground truth trajectory with decimeter-level accuracy.

Experimental Evaluation Using Data Collected With Open-Loop Route

GNSS-Positioning Evaluation
The GNSS positioning is evaluated by comparing the WLS-based GNSS positioning with the GNSS positioning aided by NLOS exclusion. The result of the GNSS positioning (two dimensions) using different methods are listed in Table 1. If all NLOS receptions are excluded from the GNSS positioning, the outcome is shown in the third column. The result obtained by the proposed NLOS-exclusion method in Algorithm 2 (WLS–NE) is shown in the fourth column. Due to the blockage from tall buildings, the majority of the measurements are NLOS.

The conventional WLS method can obtain 29.81 m of mean error. The error magnitude is much larger than the positioning error in [36], where the experiment is conducted in less urbanized areas. The standard deviation is 21.09, and the availability is 100% during the test. With the exclusion of all NLOS measurements, the GNSS positioning is even worse. The mean of its positioning error goes up to 30.25 m, and the standard deviation also slightly increases. Moreover, the availability of this solution decreases to 97.45%. This result shows that the exclusion of all NLOS measurements may not improve the overall performance in highly urbanized areas. This is due to the distortion of the satellite's geometric distribution; namely, a larger HDOP occurs. With the proposed method shown in Algorithm 2, the mean positioning error slightly improves from 29.81 to 27.09 m. Moreover, the availability of the GNSS solution is guaranteed (100%). The improvement is not too large because of the excessive NLOS receptions in the tested scenario.

The satellite numbers and the GNSS–positioning results are shown in Figure 8. The green curve represents the number of satellites when all NLOS receptions are excluded. The blue curve indicates the satellite number based on the proposed NLOS-exclusion algorithm (Algorithm 2). The satellite number can be reduced to fewer than five if all NLOS receptions are excluded, which can be seen in Figure 8(a). Due to the frequent NLOS exclusion based on Algorithm 2, the satellite number is slightly lower compared to the red curve. Only part of the identified NLOS is excluded; using Algorithm 2 can guarantee that there are enough satellites for the GNSS-positioning calculation. As shown in Figure 8(b), the proposed NLOS exclusion can introduce improvements sometimes instead of all the time. This is because there are too many NLOS receptions, and exclusion can also enlarge the HDOP in some ways.

The result of the covariance estimation based on the proposed method in the "GNSS-Positioning Covariance Estimation" section appears in Figure 9. The black dots represent the GNSS-positioning error using the WLS–NE method. This is the value that the estimated covariance was expected to approach. The red dots represent the

Table 1. The performance of the three GNSS-positioning methods (2D positioning).

All Data	Conventional: WLS	WLS–NE–A (Excluding all NLOS)	WLS–NE–P (Partially Excluding the NLOS)
Mean error	29.81 m	30.25 m	27.09 m
Standard deviation	21.09 m	22.28 m	19.6 m
Availability	100%	97.45%	100%

conventional GNSS covariance estimation based on the method in [41] ($\boldsymbol{R}_C$). This approach cannot model the NLOS error caused by signal reflection. We can see from Figure 9 that this covariance estimation slightly fluctu-ated due to the change in the HDOP. However, this covari-ance is far from the black dots. The blue dots symbolize the proposed GNSS-positioning covariance estimation result based on $\sqrt{\bar{\boldsymbol{R}}}$, as shown in (17). This covariance is
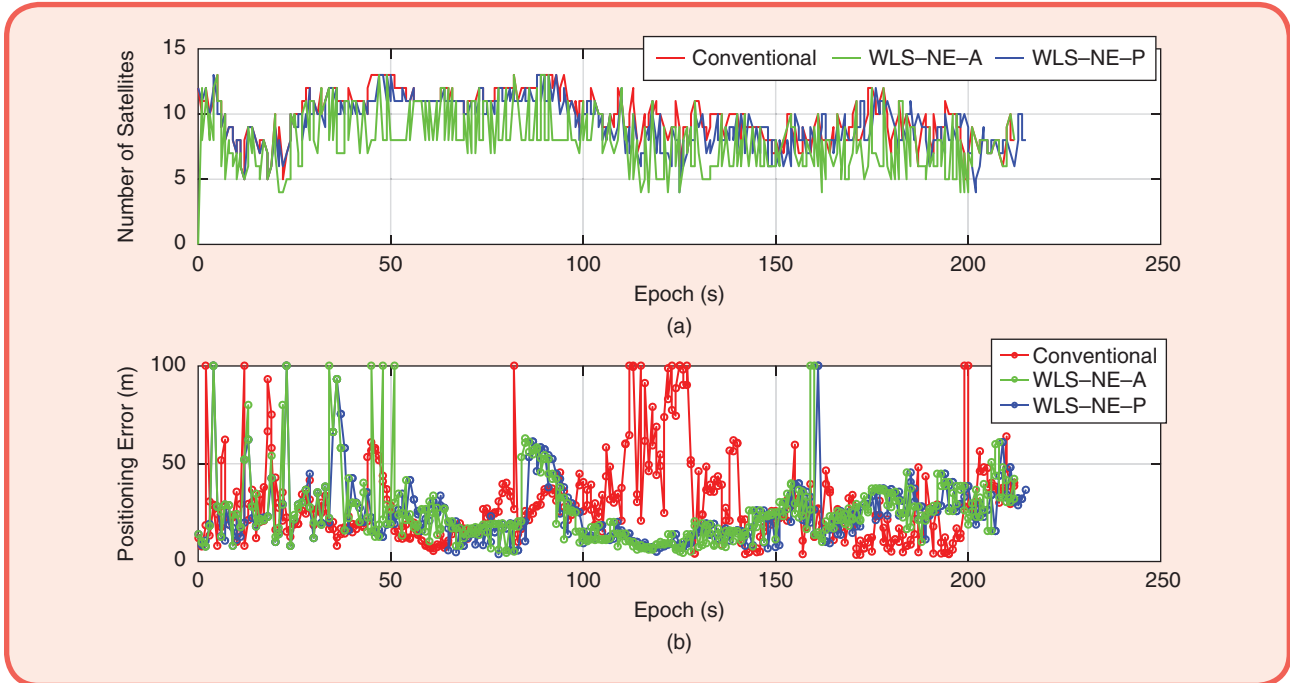

FIG 8 Experiment 1: the experimental results of the WLS and the WLS–NE, which are depicted in red dots and blue dots, respectively. (a) The number of satellites used. (b) The 3D positioning errors.
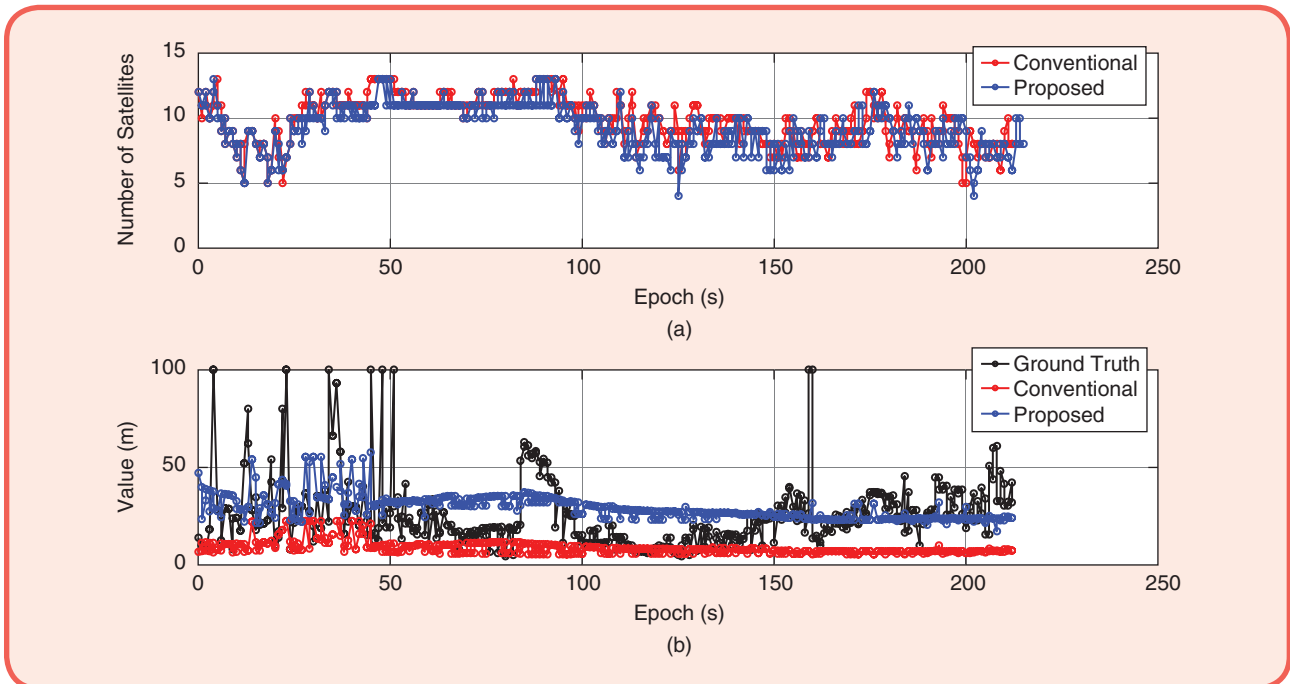

FIG 9 Experiment 1: (a) The number of satellites used in the conventional and the proposed GNSS-positioning methods. (b) The corresponding estimated covariance. The conventional and proposed covariance estimations are indicated in red and blue dots, respectively. The GNSS-positioning error when using the WLS–NE is represented by black dots (ground truth for covariance).

closer to the black dots (refer to the ground truth of the covariance) compared with that of the conventional covariance. As shown in Figure 9, the proposed covariance can effectively model the GNSS-positioning error in some epochs, especially when then GNSS-positioning error is smaller. However, some epochs encounter a large differ-

ence between the estimated covariance and the actual positioning error. This is because the proposed method can identify only the NLOS near the vehicle, which is within the detection range of the 3D lidar (commonly 120 m). The NLOS that is reflected too far from the vehicle is not modeled by the proposed covariance-estimation method. Moreover, the multipath effects are also not modeled in this covariance-estimation method.

FIG 10 Experiment 1: the trajectory of the autonomous vehicle is indicated by the green curve, and the red circles convey the GNSS-positioning result. The background image of Figure 10 is captured from Google Earth.
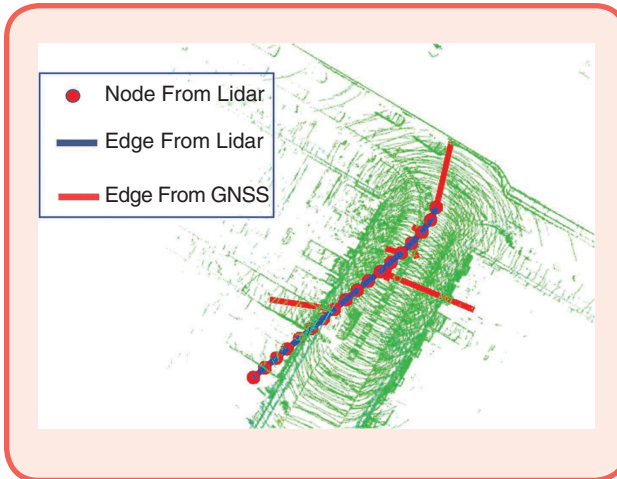

FIG 11 Graph generation in the real graph–SLAM process.

GNSS–Lidar Integration Evaluation

The trajectory of the tested vehicle is shown in Figure 10. The red circles represent the GNSS-positioning results using the proposed WLS–NE method. The green curve indicates the ground truth of the tested trajectory. We can see from the figure that the majority of the epochs possess a large positioning error. In the GNSS–lidar integration, the GNSS is the only source that can provide absolute positioning information. The SLAM graph generation is shown in Figure 11, which indicates the organized point cloud, nodes, and edges for further graph-based optimization.

As we can see from Figure 8, the WLS–NE-based GNSS-positioning solution can still reach 54 m. In this section, three GNSS–lidar integration methods are compared:

- *Method A*: GNSS–lidar integration with conventional GNSS covariance estimation [41]
- *Method B*: GNSS–lidar integration with the proposed GNSS covariance estimation
- *Method C*: GNSS–lidar integration with the proposed GNSS covariance estimation; however, the GNSS positioning is integrated into the graph optimization only when $\sqrt{R}$ is smaller than the threshold $R_{\text{threshold}}$.

The GNSS–lidar integration results are given in Table 2 using the three methods. The mean error of the conventional GNSS–lidar integration is 24.07 m, and it is improved compared to the performance of the GNSS alone (27.09 m). With the aid of the proposed GNSS-positioning covariance (method B), the error of the GNSS–lidar integration is slightly decreased to 22.67 m. The standard deviation is also slightly smaller. In method B, all GNSS-positioning results and the corresponding covariance are applied in the GNSS–lidar integration. Since the majority of the GNSS positioning is erroneous, it is reasonable to use the GNSS results when they are accurate. Accurate results can be identified when the estimated covariance is less than $R_{\text{threshold}}$. Dramatic improvement is obtained after the covariance constraint is applied. The mean error and the standard deviation are lessened to 12.67 and 6.57 m, respectively. Moreover, the availability of all three methods is 100%. This improvement shows that the proposed covariance estimation can improve the performance of the GNSS–lidar integration. The GNSS–lidar integration results are shown in Figure 12. Compared to the Bayes filter-based [13], [14] sensor-fusion

All Data	Method A Conventional GNSS–Lidar Integration	Method B Proposed GNSS–Lidar Integration	Method C Proposed GNSS–Lidar Integration 2
Mean error	24.07 m	22.67 m	12.67 m
Standard deviation	14.69 m	14.48 m	6.57 m
Availability	100%	100%	100%

Table 2. Experiment 1: The performance of the three GNSS–lidar integration methods.
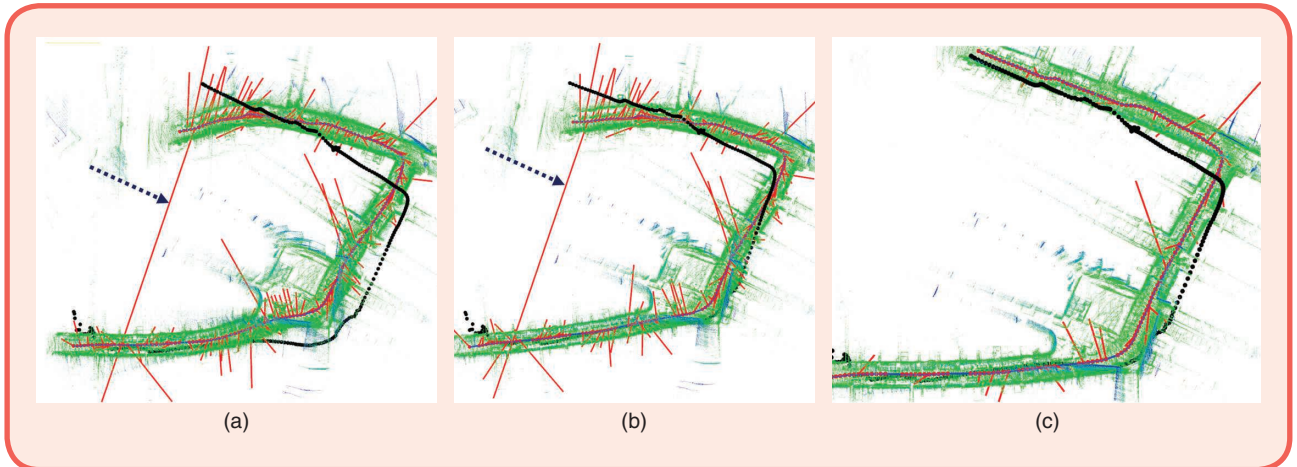
FIG 12 (a)–(c) Experiment 1: the results of the GNSS–lidar integration based on three integration methods. The blue curve consists of the optimized nodes (refer to the red node in Figure 6). The black curve indicates the ground truth of the trajectory provided by the NovAtel SPAN-CPT. The red line represents the edge from the GNSS positioning (refer to the red line in Figure 6). The green points show the organized point clouds.
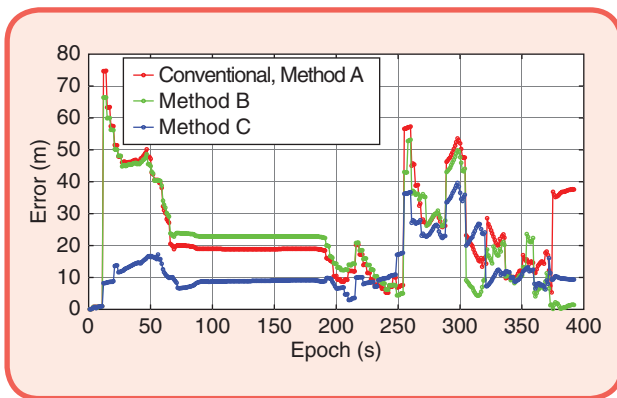

FIG 13 Experiment 1: the positioning error of the GNSS–lidar integration results obtained through the three methods. The red, green, and blue curves indicate GNSS–lidar integration methods A, B, and C, respectively.

method, the graph-based GNSS–lidar integration takes all constraints into the optimization framework. Thus, the poses of the organized point clouds, nodes, and edges changed through time. Examining Figure 12, we can conclude that

1) Proposed method C obtained the most accurate trajectory among the three methods.
2) The positioning error decreased near the end of the drive, meaning that GNSS–lidar integration can mitigate the drift of the lidar odometry.

The GNSS–lidar integration error is presented in Figure 13. We can see from the figure that method C outperforms the other two through the majority of the epochs.

Previous research [10], [40] tends to integrate the GNSS and lidar in a scenario where the GNSS-positioning error is less than 5~8 m, using the GNSS WLS. In this test scenario, the GNSS results with an enormous error are applied to the GNSS–lidar integration, which is very common in superurbanized cities, such as Hong Kong.

FIG 14 Experiment 2: the trajectory of the vehicle is indicated by the green curve, while the red circles denote the object-detection-aided GNSS-positioning results. The background image is captured from Google Earth.

Experimental Evaluation Using Data Collected With Closed-Loop Route

GNSS-Positioning Evaluation

This experiment is conducted in a superurbanized area where fewer satellites are visible compared to the first experiment. Moreover, this experiment route is a closed loop. Loop-closure [49] detection is employed in the GNSS–lidar integration process in this experiment. The experiment scene is shown in Figure 14. The height of the buildings is roughly 30 m, and the distance between the
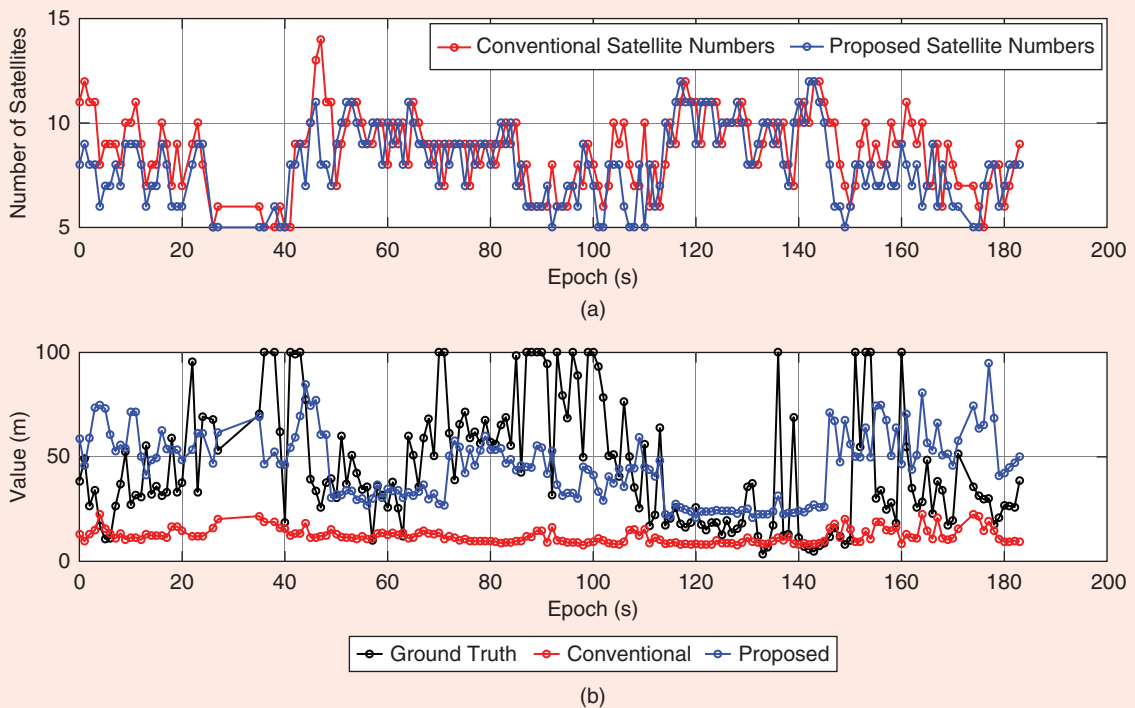
FIG 15 Experiment 2: (a) the number of satellites used in the conventional and proposed GNSS-positioning methods. (b) The corresponding estimated covariance. The conventional and proposed covariance estimations are indicated by red and blue dots, respectively. The GNSS-positioning error obtained through the WLS–NE is shown as black dots (ground truth for covariance).

Table 3. Experiment 2: The performance of the three GNSS-lidar integration methods.			
All Data	**Method A** Conventional GNSS–Lidar Integration	**Method B** Proposed GNSS–Lidar Integration	**Method C** Proposed GNSS–Lidar Integration
Mean error	25.68 m	8.14 m	7.49 m
Standard deviation	28.09 m	6.73 m	5.43 m
Availability	100%	100%	100%

structures is approximately 8 m. We can see from the figure that the majority of the GNSS-positioning results are located inside the buildings due to multipath effects and excessive NLOS receptions. Again, the mean error is slightly reduced from 46.62 m (conventional WLS) to 43.12 m (proposed WLS–NE).

The covariance-estimation result of the proposed method is provided in Figure 15. Compared with the covariance estimation in the first experiment, presented in Figure 9, the covariance is estimated better in this experiment. The main reason is that the majority of the satellites are NLOS due to the tall buildings, which means fewer multipath effects. As discussed earlier, multipath is not modeled in the proposed covariance-estimation method. In other words, the

proposed GNSS-positioning covariance-estimation method can obtain better performance in narrower streets.

GNSS–Lidar Integration Evaluation

The GNSS–lidar integration performance is shown in Table 3. Loop-closure detection is applied in this experiment, as the driving route is a closed loop, shown in Figure 14. The conventional solution obtains a mean positioning error of 25.68 m, with a standard deviation of 28.09 m. With the assistance of the proposed covariance estimation, the mean positioning error drastically decreased to 8.14 m. The mean positioning error is reduced to 7.49 m with the covariance-magnitude constraint applied. Moreover, the standard deviation is also smaller, at 5.43 m.

The final optimized nodes and the organized point clouds are in Figure 16. The positioning error during the test is given in Figure 17. We can see from Figure 16 that the edges of the GNSS are dramatically decreased with the application of the covariance constraint. Interestingly, the positioning error of all the methods decreased after 120 epochs. The reason is the detection of the loop closure, which is a strong constraint for further graph-based optimization. Regarding the performance of the conventional GNSS–lidar integration with no loop closure in the first experiment, the positioning error can still reach roughly 40 m at the end of the test.
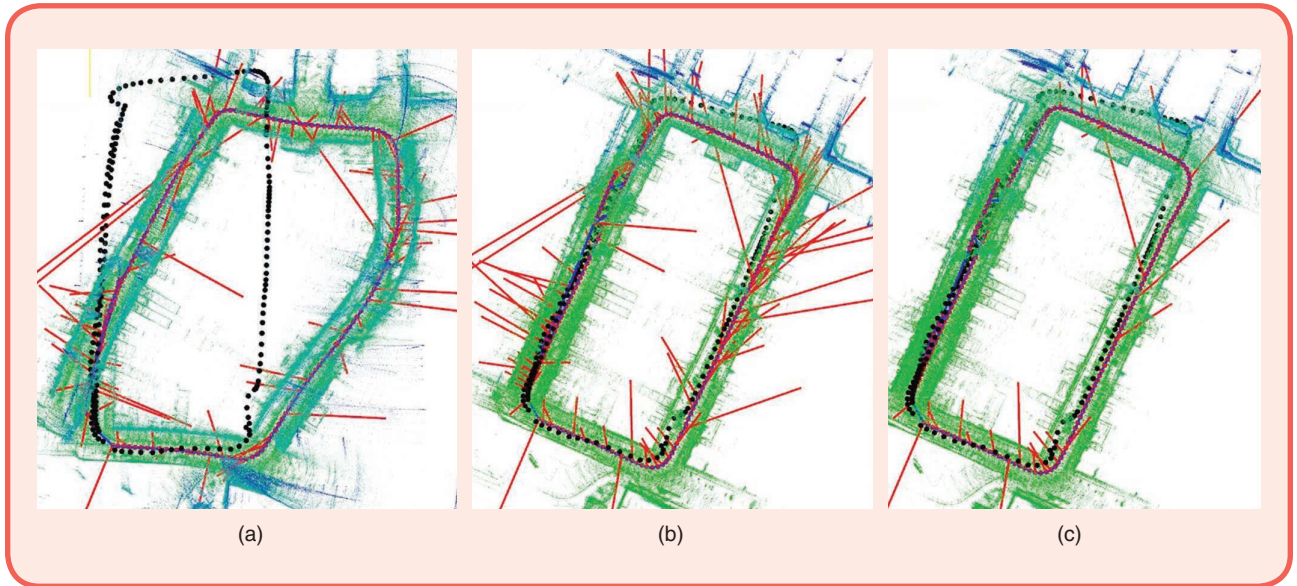
FIG 16 (a)–(c) Experiment 2: the results of the GNSS–lidar integration based on the three integration methods. The blue curve consists of the optimized nodes (refer to the red node in Figure 6). The black curve indicates the ground truth of the trajectory provided by the NovAtel SPAN-CPT. The red line represents the edge from the GNSS positioning (refer to the red line in Figure 6). The green points show the organized point clouds.

Discussion

The proposed GNSS–lidar integration method obtained improved performance with the aid of NLOS exclusion (empowered by lidar-based object detection) and the proposed covariance estimation. The proposed NLOS exclusion can obtain improvements when more satellites are available. Though the positioning performance of the GNSS is very unsatisfactory during both of the experiments, the GNSS is still indispensable for providing globally referenced positioning.

The proposed covariance estimation can capture the majority of the GNSS-positioning errors. However, the GNSS-positioning error caused by the multipath effect cannot be modeled using the proposed covariance model. In the first experiment, the mean GNSS-positioning error is less than 30 m, which is better than in the second experiment. This is because the buildings in the second experiment are even taller, which introduces more NLOS receptions. As presented in Algorithm 2, only the NLOS is modeled in the covariance. Thus, the second experiment obtains better performance regarding the GNSS-covariance estimation, which can be seen by comparing Figures 9 and 15. The multipath effects are random and difficult to model; thus, effectively representing multipath is promising work to pursue. The proposed covariance constraint applied to GNSS–lidar integration can improve performance. This novel constraint guarantees that only the accurate GNSS positioning will be applied to the integration. In other words, effectively estimated covariance can identify erroneous GNSS results.
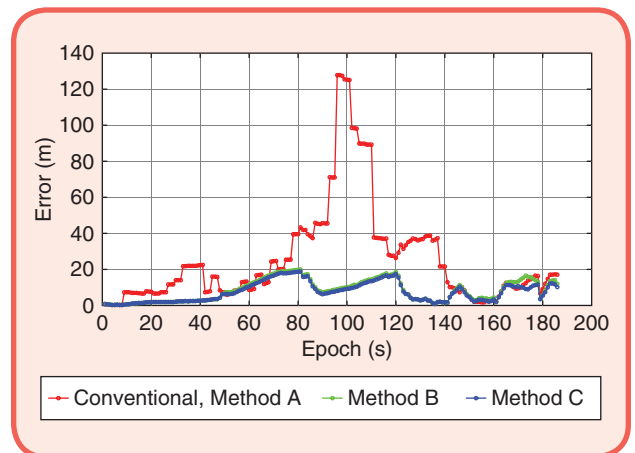

FIG 17 Experiment 2: the positioning error of the GNSS–lidar integration results based on the three methods. The red, green, and blue curves indicate GNSS–lidar integration methods A, B, and C, respectively.

Overall, the proposed GNSS-covariance estimation can improve the GNSS–lidar integration performance. The globally referenced positioning is obtained. This result proves that covariance estimation is significant for GNSS–lidar integration. However, the integrated positioning result is still large, with its best performance achieving 7.49 m of mean error in the second experiment. To realize autonomous vehicles, this kind of scenario is still a challenge for GNSS positioning. Even real-time kinematic (RTK) GNSS can suffer from severe NLOS and multipath effects. Direct NLOS exclusion will result in a big distortion of the satellites' distribution, namely, the HDOP. Thus, effectively modeling the covariance of the GNSS

positioning is a potential solution to improve the robustness of GNSS–lidar integration in superurbanized areas.

Conclusions and Future Work

With the rapid development of autonomous vehicles, the GNSS and lidar became indispensable for sensing and localization functions. The environment feature can be used to improve the GNSS-positioning performance in urbanized areas that have many tall buildings. To the best of our knowledge, this article is the first attempt to employ lidar-based object detection to improve the GNSS.

This article innovatively employs lidar perception to detect building surfaces to facilitate the covariance modeling of the GNSS positioning for GNSS–lidar integration. This study first employs lidar to provide lidar odometry based on the state-of-art NDT, and the corresponding covariance is estimated. Then, the building surface is found and identified using object detection, followed by the NLOS identification and novel exclusion. Third, the GNSS positioning is implemented using the surviving range measurements. The GNSS-positioning covariance is proposed based on an NLOS model. Fourth, the lidar odometry and the GNSS positioning are integrated through a graph-based SLAM framework. Finally, experiments are conducted to validate the proposed GNSS–lidar integration framework. The results show that the proposed method of GNSS-positioning covariance estimation can model the majority of the positioning error caused by NLOS reception. The performance of the proposed GNSS–lidar integration with adaptive covariance outperforms the conventional GNSS–lidar integration that has constant covariance. Furthermore, the remaining GNSS-positioning error caused by multipath effects will be studied and modeled to improve the performance of GNSS-positioning covariance estimation. Moreover, the RTK GNSS will be combined with lidar to verify how much the proposed method can help the RTK GNSS–lidar integration.

Acknowledgments

About the Authors

Weisong Wen is a Ph.D. degree candidate in mechanical engineering at the Hong Kong Polytechnic University, China, and a visiting student researcher at the University of California, Berkeley. His research interests include Global Navigation Satellite System/inertial navigation system/lidar/high-definition-map-based localization for autonomous vehicles.

Guohao Zhang earned his B.S. degree in mechanical engineering and automation from the University of Science and Technology Beijing, China, in 2015. He earned his M.S. degree in mechanical engineering and currently is a research assistant at Hong Kong Polytechnic University, China. His research interests include Global Navigation Satellite System urban localization, vehicle-to-vehicle cooperative localization, and multisensor integrated navigation.

Li-Ta Hsu earned his B.S. degree and Ph.D. degree in aeronautics and astronautics from the National Cheng Kung University, Tainan City, Taiwan, in 2007 and 2013, respectively. He is currently an assistant professor in the Interdisciplinary Division of Aeronautical and Aviation Engineering, Hong Kong Polytechnic University, China. Previously, he served as a postdoctoral researcher at the Institute of Industrial Science, University of Tokyo. In 2012, he was a visiting scholar at University College London. His research interests include Global Navigation Satellite System positioning in challenging environments and localization for pedestrians, autonomous vehicles, and unmanned aerial vehicles.

References

[1] C. Urmson et al., "Autonomous driving in urban environments: Boss and the urban challenge," *J. Field Robot.*, vol. 25, no. 8, pp. 425–466, 2008. doi: 10.1002/rob.20255.

[2] J. Wei, J. M. Snider, J. Kim, J. M. Dolan, R. Rajkumar, and B. Litkouhi, "Towards a viable autonomous driving research platform," in *Proc. 2013 IEEE Intelligent Vehicles Symp. (IV)*, pp. 763–770. doi: 10.1109/IVS.2013.6629559.

[3] B. Douillard et al., "On the segmentation of 3D LIDAR point clouds," in *Proc. 2011 IEEE Int. Conf. Robotics and Automation (ICRA)*, pp. 2798–2805. doi: 10.1109/ICRA.2011.5979818.

[4] W. Li, Q. Guo, M. K. Jakubowski, and M. Kelly, "A new method for segmenting individual trees from the lidar point cloud," *Photogramm. Eng. Remote Sens.*, vol. 78, no. 1, pp. 75–84, 2012. doi: 10.14358/PERS.78.1.75.

[5] G. Wan, X. Yang, R. Cai, H. Li, H. Wang, and S. Song, "Robust and precise vehicle localization based on multi-sensor fusion in diverse city scenes," 2018, pp. 4670–4677. doi: 10.1109/ICRA.2018.8461224.

[6] S. Thrun, "Probabilistic robotics," *Commun. ACM*, vol. 45, no. 3, pp. 52–57, 2002. doi: 10.1145/504729.504754.

[7] R. W. Wolcott and R. M. Eustice, "Fast LIDAR localization using multiresolution Gaussian mixture maps," in *Proc. 2015 IEEE Int. Conf. Robotics and Automation (ICRA)*, 2015, pp. 2814–2821. doi: 10.1109/ICRA.2015.7139582.

[8] A. Fernández et al., "ATENEA: Advanced techniques for deeply integrated GNSS/INS/LiDAR navigation," in *Proc. 2010 5th ESA Workshop Satellite Navigation Technologies and European Workshop GNSS Signals and Signal Processing (NAVITEC)*, pp. 1–8. doi: 10.1109/NAVITEC.2010.5708013.

[9] A. Fernández, P. Silva, and I. Colomina, "Real-time navigation and mapping with mobile mapping systems using LiDAR/Camera/INS/GNSS advanced hybridization algorithms: description and test results," in *Proc. 27th Int. Tech. Meeting Satellite Division of the Institute of Navigation (ION GNSS)*, 2014, pp. 896–903.

[10] Y. Gao, S. Liu, M. M. Atia, and A. Noureldin, "INS/GPS/LiDAR integrated navigation system for urban and indoor environments using hybrid scan matching algorithm," *Sensors*, vol. 15, no. 9, pp. 23,286–23,302, 2015. doi: 10.3390/s150923286.

[11] Y. Gu, L.-T. Hsu, and S. Kamijo, "GNSS/Onboard inertial sensor integration with the aid of 3-d building map for lane-level vehicle self-localization in urban canyon," *IEEE Trans. Vehi. Technol.*, vol. 65, no. 6, pp. 4274–4287, 2016. doi: 10.1109/TVT.2015.2497001.

[12] X. Zhang, L. Miao, and H. Shao, "Tracking architecture based on dual-filter with state feedback and its application in ultra-tight GPS/INS integration," *Sensors*, vol. 16, no. 5, pp. 627, 2016. doi: 10.3390/s16050627.

[13] G. P. Huang, A. I. Mourikis, and S. I. Roumeliotis, "Analysis and improvement of the consistency of extended Kalman filter based SLAM," in *Proc. IEEE Int. Conf. Robotics and Automation*, 2008, pp. 473–479. doi: 10.1109/ROBOT.2008.4543252.

[14] J. Zhu, N. Zheng, Z. Yuan, Q. Zhang, X. Zhang, and Y. He, "A SLAM algorithm based on the central difference Kalman filter," in *Proc. 2009 IEEE Intelligent Vehicles Symp.*, pp. 123–128. doi: 10.1109/IVS.2009.5164264.

[15] C. Cadena and J. Neira, "SLAM in O (logn) with the combined Kalman-information filter," *Robot. Auton. Syst.*, vol. 58, no. 11, pp. 1207–1219, 2010. doi: 10.1016/j.robot.2010.08.003.

[16] S. Thrun and Y. Liu, "Multi-robot SLAM with sparse extended information filers," in *Robotics Research. The Eleventh International Symposium (Springer Tracts in Advanced Robotics)*, vol. 15, P. Dario and R. Chatila, Eds. Berlin, Germany: Springer-Verlag, 2005, pp. 254–266.

[17] J.-S. Gutmann, E. Eade, P. Fong, and M. Munich, "A constant-time algorithm for vector field slam using an exactly sparse extended information filter," in *Robotics: Science and Systems VI*, Cambridge, MA: MIT Press, 2011.

[18] D. Törnqvist, T. B. Schön, R. Karlsson, and F. Gustafsson, "Particle filter SLAM with high dimensional vehicle model," *J. Intell. Robot. Syst.*, vol. 55, nos. 4–5, pp. 249–266, 2009. doi: 10.1007/s10846-008-9301-y.

[19] N. Fairfield, G. Kantor, and D. Wettergreen, "Towards particle filter SLAM with three dimensional evidence grids in a flooded subterranean environment," in *Proc. 2006 IEEE Int. Conf. Robotics and Automation*, pp. 3575–3580. doi: 10.1109/ROBOT.2006.1642248.

[20] R. Sim, P. Elinas, and J. J. Little, "A study of the Rao-Blackwellised particle filter for efficient and accurate vision-based SLAM," *Int. J. Comput. Vision*, vol. 74, no. 3, pp. 303–318, 2007. doi: 10.1007/s11263-006-0021-0.

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

[22] F. Dellaert and M. Kaess, "Square Root SAM: Simultaneous localization and mapping via square root information smoothing," *Int. J. Robot. Res.*, vol. 25, no. 12, pp. 1181–1203, 2006. doi: 10.1177/0278364906072768.

[23] F. Lu and E. Milios, "Globally consistent range scan alignment for environment mapping," *Auton. Robots*, vol. 4, no. 4, pp. 333–349, 1997. doi: 10.1023/A:1008854505733.

[24] E. Olson, J. Leonard, and S. Teller, "Fast iterative alignment of pose graphs with poor initial estimates," in *Proc. 2006 IEEE Int. Conf. Robotics and Automation*, 2006, pp. 2262–2269. doi: 10.1109/ROBOT.2006.1642040.

[25] T. Sakai, K. Koide, J. Miura, and S. Oishi, "Large-scale 3D outdoor mapping and on-line localization using 3D-2D matching," in *Proc. 2017 IEEE/SICE Int. Symp. System Integration (SII)*, pp. 829–834. doi: 10.1109/SII.2017.8279325.

[26] G. Grisetti, R. Kummerle, C. Stachniss, and W. Burgard, "A tutorial on graph-based SLAM," *IEEE Intell. Transp. Syst. Mag.*, vol. 2, no. 4, pp. 31–43, 2010. doi: 10.1109/MITS.2010.939925.

[27] L.-T. Hsu, "Analysis and modeling GPS NLOS effect in highly urbanized area," *GPS Solut.*, vol. 22, no. 1, p. 7, 2018. doi: 10.1007/s10291-017-0667-9.

[28] J. Breßler, P. Reisdorf, M. Obst, and G. Wanielik, "GNSS positioning in non-line-of-sight context—A survey," in *Proc. 2016 IEEE 19th Int. Conf. Intelligent Transportation Systems (ITSC)*, pp. 1147–1154. doi: 10.1109/ITSC.2016.7795701.

[29] M. Adjrad and P. D. Groves, "Intelligent urban positioning: Integration of shadow matching with 3D-mapping-aided GNSS ranging," *J. Navig.*, vol. 71, no. 1, pp. 1–20, 2018. doi: 10.1017/S0373463317000509.

[30] P. D. Groves, "Shadow matching: A new GNSS positioning technique for urban canyons," *J. Navig.*, vol. 64, no. 3, pp. 417–430, 2011. doi: 10.1017/S0373463311000087.

[31] L.-T. Hsu, Y. Gu, Y. Huang, and S. Kamijo, "Urban pedestrian navigation using smartphone-based dead reckoning and 3-D map-aided GNSS," *IEEE Sens. J.*, vol. 16, no. 5, pp. 1281–1293, 2016. doi: 10.1109/JSEN.2015.2496621.

[32] L.-T. Hsu, Y. Gu, and S. Kamijo, "3D building model-based pedestrian positioning method using GPS/GLONASS/QZSS and its reliability calculation," *GPS Solut.*, vol. 20, no. 3, pp. 413–428, 2016. doi: 10.1007/s10291-015-0451-7.

[33] S. Miura, L.-T. Hsu, F. Chen, and S. Kamijo, "GPS error correction with pseudorange evaluation using three-dimensional maps," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 6, pp. 3104–3115, 2015. doi: 10.1109/TITS.2015.2432122.

[34] P. François, B. David, and M. Florian, "Non-Line-Of-Sight GNSS signal detection using an on-board 3D model of buildings," in *Proc. 2011 11th Int. Conf. ITS Telecommunications (ITST)*, 2011, pp. 280–286. doi: 10.1109/ITST.2011.6060069.

[35] P. D. Groves, Z. Jiang, L. Wang, and M. K. Ziebart, "Intelligent urban positioning using multi-constellation GNSS with 3D mapping and NLOS signal detection," presented at the 25th Int. Tech. Meeting Satellite Division of the Institute of Navigation, Nashville, TX, Sept. 17–21, 2012.

[36] S. Peyraud et al., "About non-line-of-sight satellite detection and exclusion in a 3D map-aided localization algorithm," *Sensors*, vol. 13, no. 1, pp. 829–847, 2013. doi: 10.3390/s130100829.

[37] F. Peyret, D. Bétaille, P. Carolina, R. Toledo-Moreo, A. F. Gómez-Skarmeta, and M. Ortiz, "GNSS autonomous localization: NLOS satellite detection based on 3-D maps," *IEEE Robot. Autom. Mag.*, vol. 21, no. 1, pp. 57–63, 2014. doi: 10.1109/MRA.2015.2295944.

[38] D. Bétaille, F. Peyret, and M. Voyer, "Applying standard digital map data in map-aided, lane-level GNSS location," *J. Navig.*, vol. 68, no. 5, pp. 827–847, 2015. doi: 10.1017/S0373463315000152.

[39] M. Obst, S. Bauer, P. Reisdorf, and G. Wanielik, "Multipath detection with 3D digital maps for robust multi-constellation GNSS/INS vehicle localization in urban areas," in *Proc. 2012 IEEE Intelligent Vehicles Symp. (IV)*, pp. 184–190. doi: 10.1109/IVS.2012.6232285.

[40] M. Hentschel, O. Wulf, and B. Wagner, "A GPS and laser-based localization for urban and non-urban outdoor environments," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, 2008, pp. 149–154. doi: 10.1109/IROS.2008.4650585.

[41] D. Maier and A. Kleiner, "Improved GPS sensor model for mobile robots in urban terrain," in *Proc. 2010 IEEE Int. Conf. Robotics and Automation (ICRA)*, pp. 4385–4390. doi: 10.1109/ROBOT.2010.5509895.

[42] A. P. Shetty, "GPS-LiDAR sensor fusion aided by 3D city models for UAVs," M.S. thesis, Dept. Aerospace Eng., Univ. Illinois at Urbana-Champaign, 2017.

[43] W. Wen, G. Zhang, and L.-T. Hsu, Exclusion of GNSS NLOS receptions caused by dynamic objects in heavy traffic urban scenarios using real-time 3D point cloud: An approach without 3D maps. 2018. [Online]. Available: arXiv:1804.1091.

[44] J. Marshall, "Creating and viewing skyplots," *GPS Solut.*, vol. 6, nos. 1–2, pp. 118–120, 2002. doi: 10.1007/s10291-002-0017-3.

[45] J. Zhang and S. Singh, "Low-drift and real-time lidar odometry and mapping." *Auton. Robot*, vol. 41, no. 2, pp. 401–416. doi: 10.1007/s10514-016-9548-2.

[46] D. Chetverikov, D. Stepanov, and P. Krsek, "Robust Euclidean alignment of 3D point sets: The trimmed iterative closest point algorithm," *Image Vision Comput.*, vol. 23, no. 3, pp. 299–309, 2005. doi: 10.1016/j.imavis.2004.05.007.

[47] M. Magnusson, A. Lilienthal, and T. Duckett, Scan registration for autonomous mining vehicles using 3D-NDT, *J. Field Robot.*, vol. 24, no. 10, pp. 803–827, 2007. doi: 10.1002/rob.20204.

[48] M. Magnusson, "The three-dimensional normal-distributions transform: an efficient representation for registration, surface analysis, and loop detection," Ph.D. dissertation, Center for Appl. Autonomous Sensor Syst., Örebro Universitet, 2009.

[49] M. Magnusson, H. Andreasson, A. Nuchter, and A. J. Lilienthal, "Appearance-based loop detection from 3D laser data using the normal distributions transform," in *Proc. IEEE Int. Conf. Robotics and Automation (ICRA'09)*, 2009, pp. 23–28. doi: 10.1109/ROBOT.2009.5152712.

[50] W. Wen, G. Zhang, and L.-T. Hsu, "Exclusion of GNSS NLOS receptions caused by dynamic objects in heavy traffic urban scenarios using real-time 3D point cloud: An approach without 3D maps," in *Proc. 2018 IEEE/ION Position, Location and Navigation Symp. (PLANS)*, pp. 158–165. doi: 10.1109/PLANS.2018.8373577.

[51] C. Dempsey, "Global navigation satellites system," GIS Lounge, Nov. 13, 2011. [Online]. Available: https://www.gislounge.com/global-navigation-satellite-systems.

[52] E. Realini and M. Reguzzoni, "goGPS: open source software for enhancing the accuracy of low-cost receivers by single-frequency relative kinematic positioning," *Meas. Sci. Technol.*, vol. 24, no. 11, p. 115010, 2013. doi: 10.1088/0957-0233/24/11/115010.

[53] E. Kaplan and C. Hegarty, *Understanding GPS: Principles and Applications*. Norwood, MA: Artech House, 2005.

[54] R. Kümmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard, "G2o: A general framework for graph optimization," in *Proc. 2011 IEEE Int. Conf. Robotics and Automation (ICRA)*, pp. 3607–3613. doi: 10.1109/ICRA.2011.5979949.

[55] W. Wen, L.-T. Hsu, and G. Zhang, "Performance analysis of NDT-based graph SLAM for autonomous vehicle in diverse typical driving scenarios of Hong Kong," *Sensors*, vol. 18, no. 11, pp. 928, 2018. doi: 10.3390/s18113928.

[56] M. Quigley et al., "ROS: An open-source robot operating system," in *Proc. ICRA Workshop Open Source Software*, 2009, vol. 3, no. 3.2, p. 5.

[57] P. D. Groves, *Principles of GNSS, Inertial, and Multisensor Integrated Navigation Systems*. Norwood, MA: Artech house, 2013.

ITS