Correcting NLOS by 3D LiDAR and Building Height to Improve GNSS Single Point Positioning

Weisong Wen, Guohao Zhang, Li-Ta Hsu*

Abstract**—**Autonomous driving introduces high demand in GNSS in all driving environments. Currently, GNSS performance is heavily challenged in urbanized cities. The positioning error can go up to even 100 meters, due to the notorious non-line-of-sight (NLOS) receptions that dominate the GNSS positioning errors in deep-building areas. The recent state-of-the-art ray-tracing-based 3D mapping-aided (3DMA) GNSS based on the particle filter approach [1] can correct most of the pseudorange measurements affected by NLOS receptions. However, the computational load of the ray-tracing simulation is immense in that the simulations are required in each particle. In addition, an accurate prior-known receiver position is required. To address these two issues, we present a novel method to detect the GNSS signal blockage caused by surrounding buildings and correct the NLOS pseudorange measurements based on the perceived environment features by the sensor installed on an experiment vehicle. This paper demonstrates the use of LiDAR scanner and a list of building heights to describe the perceived environment. To estimate the geometry and pose of the top edges of buildings (TEBs) relative to the GNSS receiver, a surface segmentation method is employed to detect the TEBs of surrounding buildings using 3D LiDAR point clouds. The top edges of the building are extracted and extended by making use of the building height list in Skyplot to identify the NLOS-affected ones from all the measurements. Innovatively, the NLOS delay in pseudorange can be modeled by three parameters: 1) lateral distance between the reflector to the antenna of the receiver, 2) the elevation angle, and 3) azimuth angle of the satellite. Weighted least squares (WLS) is used to cooperate the corrected NLOS and other pseudorange measurements. Vehicle experiments are conducted in two different urban canyons to verify the effectiveness of the proposed method in improving GNSS single point positioning (SPP) accuracy.

Index Terms**—GPS; GNSS; LiDAR; 3D point clouds; 3D maps; NLOS; Reliability; Urban area**

I. Introduction

POSITIONING in urban environments is becoming essential due to the increasing need for autonomous vehicles. To achieve L4 [2] autonomous driving in all scenarios, centimeter-level absolute positioning is required. The 3-dimensional (3D) light detection and ranging (LiDAR) is widely used in autonomous driving vehicles [3, 4]. LiDAR, camera, and inertial navigation system (INS) only provide relative positioning. Thus, these sensors are usually integrated with the global navigation satellite system (GNSS) [5-8], because the GNSS is one of the indispensable sources that can provide absolute positioning. A GNSS/INS/LiDAR/high definition (HD) map integrated system can provide satisfactory localization service in sub-urban [9, 10] areas. The suburban areas, in fact, can be regarded as GNSS-friendly area since the GNSS receiver can receive sufficient direct signals transmitted from multi-constellation GNSS [11]. In urbanized cities, such as Tokyo, Hong Kong, and New York, the signals from satellites can be reflected, blocked, and diffracted by surrounding buildings before they are received by the receiver. If the direct light-of-sight (LOS) is blocked and reflected signals from the same satellite are received, the notorious non-light-of-sight (NLOS) receptions occur. This NLOS is the dominant GNSS positioning error in the cities mentioned above [12]. As a result, the positioning error can go up to even 100 meters [13, 14].

According to a recent review paper [12], NLOS is currently the major difficulty in use of GNSS in the applications of the intelligent transportation system. Due to NLOS, the performance of GNSS positioning becomes highly related to environmental features, such as buildings. Utilizing the 3D building model to detect the NLOS is straightforward. NLOS can be detected with the aid of a building model and then be excluded from GNSS positioning [15, 16]. However, the NLOS exclusion will distort the geometric distribution of the satellites. In urban canyons, the distortion results in large positioning error in across-street direction because only the measurements from the satellites located at the along-street direction are not excluded. Moreover, identifying NLOS measurement based on the 3D building model relies heavily on the initial guess of the GNSS receiver [17].

Fig. 1 shows the numbers of GNSS satellite (GPS and BeiDou) received by a commercial GNSS receiver in an urban canyon in Hong Kong. We can see from Fig. 1 that the number of satellites is dramatically decreased after applying NLOS exclusion. The horizontal dilution of precision (HDOP) is increased (shown in Fig. 1) distinctly. Thus, it is not preferable to exclude all the NLOS measurements in such an area (i.e., a narrow urban canyon). A smart approach, GNSS shadow matching, is proposed to match the measured satellite visibility (classifying into LOS and NLOS) with the predicted satellite visibility of hypothesized positions [18]. This method makes use of the NLOS to improve the positioning accuracy in the

Weisong, Wen, Guohao, Zhang. Li-Ta, Hsu are with the Hong Kong Polytechnic University, Hong Kong (e-mail: 17902061r@connect.polyu.hk; guohao.zhang@connect.polyu.hk; lt.hsu@polyu.edu.hk).

cross-street direction [19]. A likelihood-based 3DMA GNSS method, which models the measurement uncertainty to mitigate the NLOS receptions, is also proposed to provide accurate positioning in the along-street direction [20]. Due to the complementariness of the shadow matching and the likelihood-based 3DMA GNSS, their integration has been studied recently [21]. Another stream of range-based 3DMA GNSS methods is to correct the NLOS affected measurement for GNSS positioning [1, 22-24]. These methods are proposed to simulate the signals' transmission routes using the ray-tracing method [25]. However, the drawbacks of these ray-tracing-based 3DMA GNSS methods are the stringent requirements on 1) the accuracy of 3D mapping database, 2) the initial guess of receiver positions, and 3) the computational power of the processors.

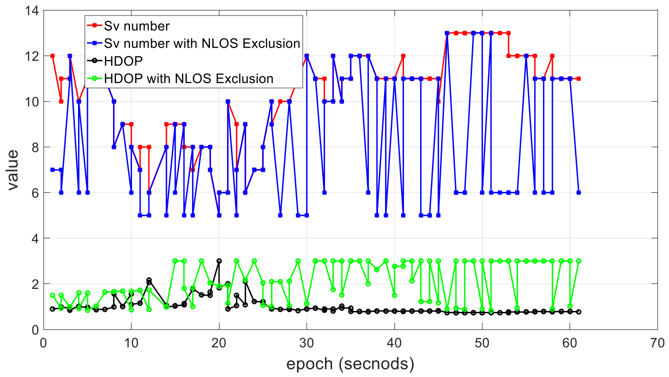

Fig. 1. Demonstration of numbers of satellite (GPS/BeiDou) measurement before (red) and after (blue) NLOS exclusion in an urban canyon in Hong Kong. The horizontal dilution of precision (HDOP) are also given before (black) and after (green) exclusion.

Instead of using only the 3D mapping data, other scanning sensors, including cameras and LiDAR, can also be employed to sense the surrounding environments of the receiver in real-time operation. To detect the visibility of satellites, omnidirectional and fisheye cameras [26-28] are used to detect the skylines of buildings in the urbanized area. NLOS receptions can be detected with the detected skylines, and some improvements are obtained. However, this method can suffer from strong light or night scenarios, as computer vision is employed to detect the skylines. The constructed map of the environment using 3D LiDAR is employed to classify the visibility of satellites. A study then modeled the GNSS noise covariance by NLOS detection based on a LiDAR-constructed map [29]. A research incorporating LiDAR map and 3D city model to exclude NLOS is conducted in the application of unmanned aerial vehicles [30]. However, these methods still tend to exclude the NLOS receptions from further GNSS positioning, which is not applicable in the deep urban areas.

In this paper, we propose to improve the GNSS single point positioning (SPP) by detecting and correcting the NLOS receptions based on the environment features perceived by real-time 3D point clouds generated by 3D LiDAR. The perceived environment features refer to the surrounding buildings of the receiver in this paper. Dimension and pose (including the position and orientation) of the building wall relative to GNSS receiver are calculated by the point cloud-based segmentation. Due to the limited field of view (FOV), tall buildings cannot be fully scanned. Thus, the height of the detected building wall is extended to the exact height provided by a building height list obtained from Google Earth. Then, the satellites and detected TEBs are projected into the Skyplot. To implement the projection, the globally referenced yaw angle of the vehicle is needed and is provided by an RTK GNSS/INS integrated system (SPAN-CPT). Based on the detected TEBs, NLOS measurement can be identified. Previously, our team proposed an NLOS error model based on two parameters: the distance between the GNSS receiver and NLOS reflector, and the elevation and azimuth angle of the satellite [13]. In this paper, innovatively, the distance between the GNSS receiver and reflectors can be obtained by LiDAR scanner. Thus, the correction of NLOS-affected pseudorange measurements can be calculated. Finally, GNSS SPP is calculated using both the corrected NLOS measurements and LOS visible measurements.

To the best of the authors' knowledge, this is the first attempt to aid GNSS SPP by employing the real-time 3D point clouds to detect and correct the NLOS measurements. This is important because GNSS is usually integrated with dead-reckoning (e.g., INS, odometer, visual odometry, LiDAR odometry, etc.) for various applications. Tightly-coupled integration is one of the most popular existing solutions for the integrated navigation system. Our proposed GNSS SPP with NLOS correction can easily fit into the tightly-coupled integration scheme. In other words, it can be easily implemented in many existing navigation systems, especially those used in autonomous driving.

The remainder of this paper is structured as follows. An overview of the proposed method is given in Section II. Section III discusses a method to detect TEBs from LiDAR point clouds. Coordinate transformation from LiDAR to GNSS Skyplot coordinate system is also presented in this section. In Section IV, the criterion of NLOS detection is proposed and the NLOS correction model is introduced. In Section V, we evaluate the effectiveness of the proposed method by two vehicle experiments in two typical urban canyons in Hong Kong. Finally, conclusions are drawn in Section VI.

II. Overview of The Proposed Method

In this study, we focus on the NLOS receptions caused by surrounding buildings. Fig. 2 presents direct propagation routes, multipath and potential NLOS receptions of GNSS signals. The buildings, of which height is indicated by H, can block a signal transmitted from a satellite, for example, satellite 1 in Fig. 2. Meanwhile, this GNSS signal is reflected by the other nearby building and finally received by GNSS receiver equipped on top of the autonomous vehicle, which results in NLOS receptions. Actually, this kind of scenario is a regular case in Hong Kong. In this case, the number of satellites visible to the GNSS receiver is related to the height of buildings and the distance from the receiver to the building (α_i in Fig. 2).

As a significant sensor for positioning and perception of autonomous driving [31], 3D LiDAR is installed on the top, as shown in Fig. 2. In this paper, LiDAR is employed to detect the surrounding building surfaces and obtain the distance from the GNSS receiver to the building surface, and then the TEBs can be identified consequently. Then, NLOS detection and correction is implemented based on detected TEBs, which are projected into a Skyplot, and the distance from GNSS receiver to buildings. Finally, GNSS positioning is performed using both the corrected and healthy pseudorange measurements. Fig. 3 shows the flowchart of the proposed method. The proposed method can be executed as follows:

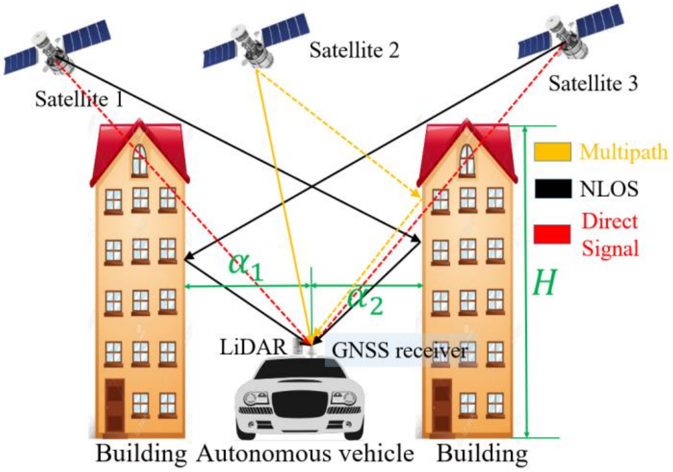

Fig. 2. Illustration of GNSS signal transmission routes in the urbanized area in Hong Kong. NLOS/multipath can be caused by surrounding buildings.

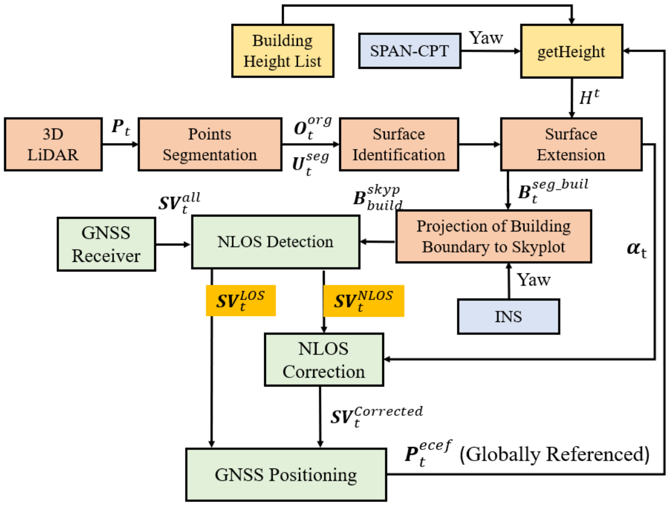

Fig. 3. Flowchart of the proposed method of GNSS SPP with NLOS correction. The inputs are the 3D LiDAR, yaw angle, building height list, and GNSS raw measurements.

Step I: Point cloud segmentation method is employed to detect the building surface. The pose of TEBs relative to the GNSS receiver are calculated. The distance between the GNSS receiver and the buildings can be obtained subsequently. Moreover, building height list from Google Earth is employed to extend the detected building height to the exact height.

Step II: The TEBs are projected into a GNSS Skyplot based on their estimated poses relative to the GNSS receiver, and yaw angle provided by the SPAN-CPT (RTK GNSS/INS integrated navigation system).

Step III: Considering satellites elevation angle, azimuth angle, SNR, and TEB information (elevation and azimuth angles in Skyplot), satellites blocked by buildings are detected. Consequently, NLOS correction is implemented with an NLOS error model.

Step IV: Implementing GNSS weighted least squares (WLS) based on the corrected pseudorange measurements and healthy pseudorange measurements.

The details of the algorithms are introduced in the following sections.

III. Building Surface Detection and Transformation

To detect the surface of the building and obtain the corresponding distances between the GNSS receiver and buildings, a point cloud segmentation method is employed to implement the building surface detection in this section.

A. *TEBs Detection and Transformation*

The surrounding environment is expressed as points set $\boldsymbol{P}_t = \{\boldsymbol{p}_1, \boldsymbol{p}_2, \dots, \boldsymbol{p}_n, t\}$ at a given time, t, where $\boldsymbol{p}_i = (x_i, y_i, z_i)$ represents a single point in the LiDAR coordinate system. To distinguish the building surface from the unordered points set and determine the distance from GNSS receiver to the building surface, three steps are needed: 1) Segmentation of point clouds, 2) Identification of buildings from segmented objects, and 3) Extension of top edges of buildings (TEBs).

The point clouds segmentation is summarized in detail in Algorithm 1. Inputs of Algorithm 1 are: points set (3D point clouds) and search radius r_{search}, which is the variable constraining the searching area in the *KD*-tree [32]. Outputs include the bounding box [33] sets ($\boldsymbol{U}_{\mathbf{t}}^{\mathbf{seg}}$) and organized point clusters ($\boldsymbol{O}_{\mathbf{t}}^{\mathbf{org}}$), which indicate different objects, such as buildings and vehicles. The definitions of applied variables and functions in Algorithm 1 are listed as follows:

- $\boldsymbol{P}_{\mathbf{t}}$: input 3D point clouds. $\boldsymbol{U}_{\mathbf{t}}^{\mathbf{seg}}$: segmented bounding box sets. $\boldsymbol{O}_{\mathbf{t}}^{\mathbf{org}}$: segmented point clusters.
- $\boldsymbol{P}_{\mathbf{t}}^{\mathbf{check}}$: a middle variable that contains checked points. $\boldsymbol{N}_i$: a neighboring points set given a searching radius. r_{search}: the radius of neighboring points searching area.

The *BoundingBox* mentioned in Algorithm 1 is a function to get the bounding box that represents the organized point cluster. Bounding box $\boldsymbol{U}_{\boldsymbol{i}}$ is specifically determined by a vector, $\boldsymbol{U}_i$, as follows:

$$\boldsymbol{U}_{\mathbf{i}} = \left[x_i^c, y_i^c, z_i^c, roll_i^c, pitch_i^c, yaw_i^c, d_i^{len}, d_i^{wid}, d_i^{hei}\right]^T \quad (1)$$

where x_i^c, y_i^c, and z_i^c denote the position of the bounding box in *x*, *y*, and *z* directions in LiDAR coordinate system, respectively. $roll_i^c$, $pitch_i^c$, and yaw_i^c denote the orientation of the bounding box in LiDAR coordinate system. d_i^{len} is the length, d_i^{wid} is the width, and d_i^{hei} is the height of the bounding box.

The principle of Algorithm 1 is also shown in the left side of Fig.4. The colored points represent raw 3D point clouds. After applying the Algorithm 1, two clusters are detected which are annotated by the two black 2D bounding box ($\boldsymbol{U}_\mathbf{t}^\mathbf{seg}$). However, we do not know which belongs to the building class.

Algorithm 1: Segmentation for points set $\boldsymbol{P}_t$

```
Input: points set $\boldsymbol{P}_\mathbf{t} = \{\boldsymbol{p}_1, \boldsymbol{p}_2, \dots, \boldsymbol{p}_n, t\}$, search radius $r_{search}$
Output: Bounding box sets $\boldsymbol{U}_\mathbf{t}^\mathbf{seg} = \{\boldsymbol{U}_1, \boldsymbol{U}_2, \dots, U_i, \dots \boldsymbol{U}_m, t\}$, Organized point clusters $\boldsymbol{O}_\mathbf{t}^\mathbf{org} = \{\boldsymbol{O}_1, \boldsymbol{O}_2, \dots, \boldsymbol{O}_i, \dots \boldsymbol{O}_m, t\}$
create a KD-tree representation for the input points set $\boldsymbol{P}_t$
setup an empty list to save point sets $\boldsymbol{P}_\mathbf{t}^\mathbf{check}$
for all points $p_i$ in $\boldsymbol{P}_\mathbf{t}$ do
  add $p_i$ to the points set $\boldsymbol{P}_t^{check}$
  for all $p_i$ in $\boldsymbol{P}_\mathbf{t}^\mathbf{check}$ do
    search for the points set $\boldsymbol{N}_i$ of point neighbor of $\boldsymbol{p}_i$ in a sphere with radius $r$<$r_{search}$
    for every point $\boldsymbol{N}_i^i$ in points set $\boldsymbol{N}_i$ do
      if $\boldsymbol{N}_i^i$ have not been processed
      add $\boldsymbol{N}_i^i$ to points sets $\boldsymbol{P}_t^{check}$
      end if
    end for the points set $\boldsymbol{N}_i$
  if all the points in $\boldsymbol{P}_\mathbf{t}^\mathbf{check}$ have been processed
    add $\boldsymbol{P}_\mathbf{t}^\mathbf{check}$ to $\boldsymbol{O}_\mathbf{t}^\mathbf{org}$ as an organized points set
    add BoundingBox($\boldsymbol{P}_\mathbf{t}^\mathbf{check}$) to $\boldsymbol{U}_\mathbf{t}^\mathbf{seg}$ as a bounding box
    reset $\boldsymbol{P}_\mathbf{t}^\mathbf{check}$ to empty
  end if
  end for $\boldsymbol{P}_\mathbf{t}^\mathbf{check}$
end for $\boldsymbol{P}_\mathbf{t}$
```

To effectively identify the bounding box ($\boldsymbol{U}_\mathbf{t}^\mathbf{seg}$) representing the building surface which can result in GNSS signal reflections and subsequent NLOS receptions, a surface identification method is needed; this is summarized in detail in Algorithm 2. The objective of Algorithm 2 is to: 1) identify the buildings, shown in the middle side of Fig. 4 and 2) extend its heights to exact one (right side of Fig. 4). The inputs of this algorithm are $\boldsymbol{U}_t^{seg}$ and $\boldsymbol{O}_t^{org}$ obtained from Algorithm 1, and some experimentally determined thresholds.

The definitions of applied variables and functions in Algorithm 2 are listed as follows:

- num_{thres}: The number of points that the cluster belongs to the building class should contain.
- len_{thres}: minimum length of a 2D bounding box which belongs to the building class.
- hei_{thres}: minimum height of a 2D bounding box which belongs to the building class.
- building height list, H_{build}, receiver position $\boldsymbol{P}_r^{ecef}$, and yaw angle, Yaw_r.

The output is the bounding box set $\boldsymbol{B}_t^{seg\_buil}$ specifically representing the building surface. The function *Num* mentioned in Algorithm 2 is used to calculate the number of points in each cluster, $\boldsymbol{O}_i$. The function $getHeight$ is used to search the height information from a saved building height list, which contains the height information. To determine the actual height of the identified building surface, $\boldsymbol{P}_r^{ecef}$, $\boldsymbol{U}_i$, and Yaw_r are also needed. $\boldsymbol{P}_r^{ecef}$ indicates the GNSS position given by previous-epoch positioning result. Relative position between GNSS receiver and detected building can be obtained from $\boldsymbol{U}_i$. For each bounding box, $\boldsymbol{B}_i$, the distance, α_i, from the receiver to the detected building surface can be calculated as follows:

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

Algorithm 2: Building surface identification from bounding box sets and height extension of TEBs

```
Input: Bounding Box sets $\boldsymbol{U}_\mathbf{t}^\mathbf{seg} = \{\boldsymbol{U}_1, \boldsymbol{U}_2, \dots, U_i, \dots \boldsymbol{U}_m, t\}$, Organized point clusters $\boldsymbol{O}_\mathbf{t}^\mathbf{org} = \{\boldsymbol{O}_1, \boldsymbol{O}_2, \dots, \boldsymbol{O}_i, \dots \boldsymbol{O}_m, t\}$, point number threshold $num_{thres}$, length threshold $len_{thres}$ and height threshold $hei_{thres}$, building height list $\boldsymbol{H}_{build}$, receiver position $\boldsymbol{P}_\mathbf{r}^\mathbf{ecef}$, yaw angle $Yaw_r$
Output: Bounding Box set represents building surfaces $\boldsymbol{B}_\mathbf{t}^\mathbf{seg\_buil} = \{\boldsymbol{B}_1, \boldsymbol{B}_2, \dots, \boldsymbol{B}_i, \dots \boldsymbol{B}_m, t\}$
setup an empty clusters list $\boldsymbol{B}_\mathbf{t}^\mathbf{seg\_buil}$ to save bounding box
for all bounding box $\boldsymbol{U}_i$ in $\boldsymbol{U}_\mathbf{t}^\mathbf{seg}$ do
  if $Num(\boldsymbol{O}_i) > num_{thres}$
    $\boldsymbol{U}_\mathbf{i} \leftarrow [x_i^c, y_i^c, z_i^c, roll_i^c, pitch_i^c, yaw_i^c, d_i^{len}, d_i^{wid}, d_i^{hei}]$
    if $d_i^{len} > len_{thres}$ and $d_i^{hei} > hei_{thres}$
      $d_i^{hei} \leftarrow getHeight(\boldsymbol{H}_\mathbf{build}, \boldsymbol{P}_\mathbf{r}^\mathbf{ecef}, \boldsymbol{U}_\mathbf{i}, Yaw_r)$
      $\boldsymbol{B}_i \leftarrow \boldsymbol{U}_\mathbf{i}$
    end if
  end if
end for $\boldsymbol{U}_\mathbf{t}^\mathbf{seg}$
```

Thus, the bounding box with extended height representing the building surface can be identified with Algorithm 2. The height of the bounding box representing building surface can be extended to the real one. The bounding box is extended from rectangle ABCD to rectangle CDEF, as can be seen in the right side of Fig. 4. Then, the parameters of TEBs for the bounding box, B_i, corresponding to building surface are denoted by line segment $\overline{EF}$, denoted as $\boldsymbol{B}_{build}^{3d}$, the matrix of the building boundary [18]. To represent the building boundary, two points, E and F, are required. The B_{build}^{3d} is structured as follows:

$$\boldsymbol{B}_\mathbf{build}^\mathbf{3d} = \begin{bmatrix} x_{3dE} & y_{3dE} & z_{3dE} \\ x_{3dF} & y_{3dF} & z_{3dF} \end{bmatrix} \qquad (3)$$

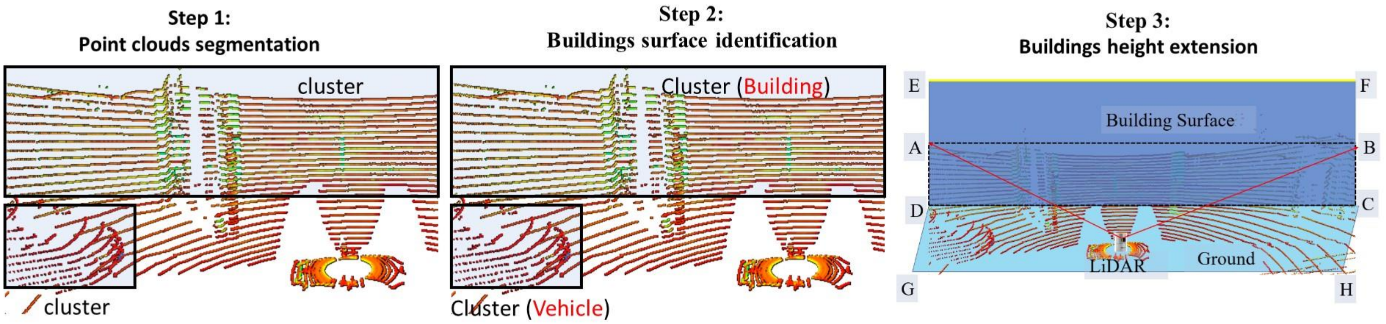

Fig. 4 Illustration of point sets segmentation and TEBs identification, extension. Box ABCD represents the initially detected building surface. Box CDEF represents the extended building surface. Box CDGH represents the ground. The color points denote the point clouds from 3D LiDAR.

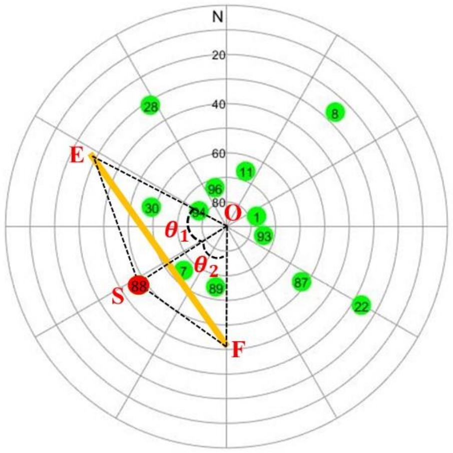

Fig. 5 Skyplot of GNSS satellites and detected TEBs. Green and red circles and the numbers indicate satellites and corresponding PRNs. Line segment $\overline{EF}$ indicates the TEBs.

B. Coordinate Transformation

To implement the algorithm of NLOS detection and subsequent correction, satellites' visibility must be determined based on the extended TEBs. Thus, the relative poses of the GNSS receiver to satellites and to building surfaces need to be transformed into the same representation, the Skyplot. In each epoch, information from satellites, including azimuth, elevation angles, and SNR, can be obtained from the GNSS receiver. Part of satellite information can be represented as $\boldsymbol{SV}_{\mathbf{t}}^{\mathbf{all}} = \{\boldsymbol{SV}_{\boldsymbol{1}}, \boldsymbol{SV}_{\boldsymbol{2}}, \dots, \boldsymbol{SV}_{\boldsymbol{i}}, \dots \boldsymbol{SV}_{\boldsymbol{n}}\}$. n represents the number of satellites received. $\boldsymbol{SV}_{\boldsymbol{i}}$ represents the information for satellite i, and $\boldsymbol{SV}_{\boldsymbol{i}} = \{az_i, el_i, SNR_i, \rho_i\}$. az_i denotes the satellite azimuth angle. el_i represents satellite elevation angle. SNR_i indicates satellite signal-to-noise ratio (SNR) and ρ_i denote the pseudorange measurement.

Satellite positions can be easily indicated in the Skyplot which is a 2-dimension coordinate based on corresponding elevation and azimuth angles. A transformation matrix should be employed for TEBs transformation from a 3D coordinate to a 2D coordinate. The transformation is conducted as per the following formula.

$$\boldsymbol{B}_{\mathrm{build}}^{\mathrm{skyp}} = \boldsymbol{B}_{build}^{3d} \boldsymbol{G}_{T}^{build} \tag{4}$$

where $\boldsymbol{B}_{build}^{3d}$ denotes the matrix of building boundary presented in the previous sub-section. $\boldsymbol{G}_{T}^{build}$ is a 3x2 transform matrix. $\boldsymbol{B}_{build}^{skyp}$ denotes the boundary matrix (2x2) in Skyplot structured as follows:

$$\boldsymbol{B}_{\mathrm{build}}^{\mathrm{skyp}} = \begin{bmatrix} x_{skyE} & y_{skyE} \\ x_{skyF} & y_{skyF} \end{bmatrix} \tag{5}$$

After the transformation, satellites and building surface boundaries can be presented in the same coordinate frame, the Skyplot, as shown in Fig. 5. Bounding box set $\boldsymbol{B}_{\mathrm{t}}^{\mathrm{seg\_buil}} = \{\boldsymbol{B}_1, \boldsymbol{B}_2, \dots, \boldsymbol{B}_i, \dots \boldsymbol{B}_m, t\}$ can be transformed into $\boldsymbol{B}_{\mathbf{t}}^{\mathbf{skyp}} = \{\boldsymbol{B}_1^{skyp}, \boldsymbol{B}_2^{skyp}, \dots, \boldsymbol{B}_i^{skyp}, \dots \boldsymbol{B}_m^{skyp}, t\}$, where $\boldsymbol{B}_{\mathrm{i}}^{\mathrm{skyp}}$ indicates the i_{th} boundary in the Skyplot. Moreover, the distance list representing the distances from GNSS the receiver to the detected surfaces can also be obtained as $\boldsymbol{\alpha}_{\mathbf{t}}^{\mathbf{seg\_buil}} = \{\alpha_1, \alpha_2, \dots, \alpha_i, \dots \alpha_m, t\}$, where α_i is associated with $\boldsymbol{B}_{\mathbf{i}}^{\mathbf{skyp}}$. Line segment $\overline{EF}$ represents the building surface boundary corresponding to line segment $\overline{EF}$ shown in Fig. 4. Then, the azimuth and the elevation angles for point E and F can be calculated in the Skyplot, respectively.

IV. Improved GNSS Positioning with NLOS Correction

In this section, an NLOS error model is presented first. Then, the NLOS detection criterion is proposed based on the detected TEBs, satellite elevation angle, azimuth angle, and SNR. NLOS error correction is then implemented. Finally, the GNSS positioning is conducted by applying the WLS method using the LOS and corrected NLOS pseudorange measurements.

A. NLOS Correction Based on Detected Building Boundary

In terms of the measurements from the GNSS receiver, each pseudorange measurement, ρ_n, is written as follows [34].

$$\rho_n = R_n + c(\delta t^{\mathrm{r}} - \delta t_n^{\mathrm{sv}}) + I_n + T_n + \varepsilon_n \tag{6}$$

where R_n is the geometric range between the satellite and the GNSS receiver. δt_n^{sv} denotes the satellite clock bias. δt^{r} indicates the receiver clock bias. I_n represents the ionospheric delay distance; T_n indicates the tropospheric delay distance. ε_n represents the errors caused by the multipath effects, NLOS receptions, receiver noise, antenna delay. In this paper, we focus on mitigating the NLOS errors caused by surrounding buildings.

The NLOS error model proposed in [13] is expressed in Fig. 6. The expected signal transmission route is expressed as a dashed blue line in Fig. 6. α represents the distance from the receiver to the building. θ_{ele} represents the elevation angle of the GNSS signal. We assume that:

(1) The building is vertical to the ground.

(2) GNSS signal reflection satisfies the law of reflection.

Thus, we can get $\theta_a = \theta_b$. Moreover, the direction of real signal transmission is parallel to the direction of expected signal transmission. Finally, we have $\theta_a = \theta_b = \theta_0 = \theta_{ele}$ .The route distance difference, γ, between the reflected signal and the expected signal is indicated as follows:

$$\gamma = \gamma_1 + \gamma_2 \tag{7}$$

$$\gamma_1 = \alpha sec\theta_{ele} \tag{8}$$

$$\gamma_2 = \gamma_1 \cos\,(2\theta_{ele}) \tag{9}$$

Thus, the NLOS error can be calculated based on the azimuth angle, elevation angle, and the distance from the receiver to the building causing the reflection. In general, two steps are needed to proceed with the NLOS correction: NLOS detection and NLOS error calculation. The process of NLOS correction is summarized in detail in Algorithm 3.

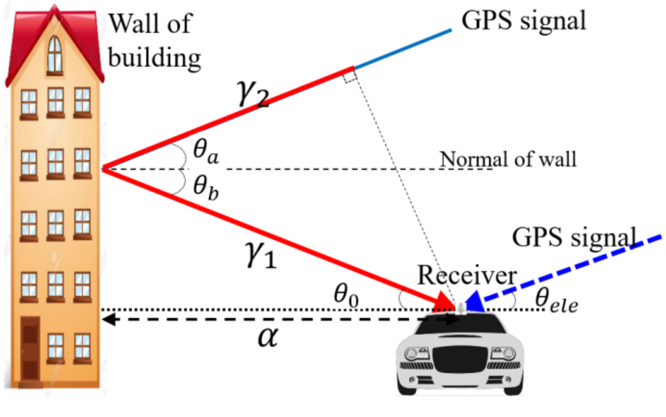

Fig. 6 NLOS correction model. The signal is reflected by the building and subsequently received by receiver [13].

Algorithm 3: NLOS detection and correction

Input: Satellites information set $\boldsymbol{SV_t^{all}} = \{\boldsymbol{SV_1}, \boldsymbol{SV_2}, \dots, \boldsymbol{SV_i}, \dots \boldsymbol{SV_n}\}$, building boundary matrix $\boldsymbol{B_t^{skyp}} = \{\boldsymbol{B_1^{skyp}}, \boldsymbol{B_2^{skyp}}, \dots, \boldsymbol{B_i^{skyp}}, \dots \boldsymbol{B_m^{skyp}}, t\}$, distance list $\boldsymbol{\alpha_t^{seg\_buil}}$, area threshold $S_{threshold}$, SNR threshold $SNR_{threshold}$, threshold of boundary uncertainty θ_{thres}

Output: corrected satellites information set after NLOS identification: $\boldsymbol{SV_t^{cor}} = \{\boldsymbol{SV_1^{cor}}, \boldsymbol{SV_2^{cor}}, \dots, \boldsymbol{SV_i^{cor}}, \dots \boldsymbol{SV_j^{cor}}\}$. LOS satellite assembles $\boldsymbol{SV_t^{los}} = \{\boldsymbol{SV_1^{los}}, \boldsymbol{SV_2^{los}}, \dots, \boldsymbol{SV_i^{los}}, \dots \boldsymbol{SV_k^{los}}\}$.

```
for all boundary $\boldsymbol{B_i^{skyp}}$ in $\boldsymbol{B_t^{skyp}}$ do
  for all satellites $\boldsymbol{SV_i}$ in $\boldsymbol{SV_t^{all}}$ do
    estimate $\theta_1$, $\theta_2$ as shown in Fig. 5
    Get triangle area $S_{\Delta SEO}$ of triangle SEO from $\boldsymbol{B_i^{skyp}}$
    Get triangle area $S_{\Delta SFO}$ of triangle SFO from $\boldsymbol{B_i^{skyp}}$
    Get triangle area $S_{\Delta SEF}$ of triangle SEF from $\boldsymbol{B_i^{skyp}}$
    Get triangle area $S_{\Delta EOF}$ of triangle EOF from $\boldsymbol{B_i^{skyp}}$
    $\Delta S = S_{\Delta SEO} + S_{\Delta SFO} + S_{\Delta SEF} - S_{\Delta EOF}$
    if $(SNR_i > SNR_{threshold}) or\ (\theta_1 < \theta_{thres})\ or\ (\theta_2 < \theta_{thres})$
      break
    if $\Delta S > S_{threshold}$ and $((\theta_1 + \theta_2) < \angle EOF < 180^\circ$
      $\boldsymbol{SV_i^{los}} \leftarrow \boldsymbol{SV_i}$ // LOS
    else // NLOS
      $\boldsymbol{SV_i} \leftarrow \{az_i, el_i, SNR_i, \rho_i\}$
      $\boldsymbol{SV_i}(\rho_i) \leftarrow \boldsymbol{SV_i}(\rho_i) - (\gamma_1 + \gamma_2)$ // $\alpha_i$ from $\boldsymbol{\alpha_t^{seg\_buil}}$
      $\boldsymbol{SV_i^{cor}} \leftarrow \boldsymbol{SV_i}$
    end if
  end for satellites set $\boldsymbol{SV_t^{all}}$
end for boundary set $\boldsymbol{B_t^{skyp}}$
```

The inputs of Algorithm 3 include satellites information, SV_t^{all} , building surface boundaries information , $\boldsymbol{B_t^{skyp}}$, distance list, $\boldsymbol{\alpha_t^{seg\_buil}}$ and some experimentally determined thresholds. The definitions of applied variables and functions in Algorithm 2 are listed as follows:

- $S_{threshold}$: Used to determine whether the satellite is inside the triangle. For example, if satellite 88 is inside triangle EOF.
- $SNR_{threshold}$: if the SNR for a certain satellite is more than this threshold, we treat it as LOS.
- θ_{thres}: threshold of boundary uncertainty.

The outputs are the corrected satellites information set $\boldsymbol{SV_t^{cor}}$ and LOS satellite assembles $\boldsymbol{SV_t^{los}}$. Firstly, geometry angle θ_1 (∠EOS) and θ_2 (∠FOS) shown in Fig. 5 are estimated. Then areas of triangles $S_{\Delta SEO}$, $S_{\Delta SFO}$, $S_{\Delta SEF}$, and $S_{\Delta EOF}$ are calculated and ΔS can be estimated subsequently. Secondly, GNSS measurement whose SNR is larger than $SNR_{threshold}$ will not be excluded, as signals with strong SNR is not considered to be reflected by buildings.

Satellites whose positions are quite near the extended edge beam ($\theta_1 < \theta_{thres}$ or $\theta_2 < \theta_{thres}$) also should not be excluded, such as satellite 7 in Fig. 5; thus, the angle threshold θ_{thres} is set. To avoid a faulty exclusion, a heuristically determined threshold $S_{threshold}$ is set. Satellites whose positions are quite near the TEBs of building surface should not be identified as NLOS, which can be determined by comparing ΔS and $S_{threshold}$, such as from satellite 7 in Fig. 5. Finally, the pseudorange measurements from NLOS receptions can be corrected using the NLOS error model in formula (7).

In this case, these NLOS satellites can be detected, and corresponding pseudorange measurements are corrected.

B. GNSS Positioning Based on Corrected and Healthy Pseudorange Measurements

Measurements with low elevation angle are more likely to be a contaminated GNSS signal in urban canyon, such as the multipath or NLOS, due to the reflection, blockage, and diffraction. Thus, proper threshold must be set to exclude the unhealthy measurements. For satellite SV_i, if ele_i is less than ele_{thres}, it should be excluded from GNSS positioning. Pseudorange measurements in $\boldsymbol{SV}_t^{cor}$ and $\boldsymbol{SV}_t^{los}$ will be employed for GNSS positioning calculation.

The clock bias between the GNSS receiver and satellites is usually represented by the pseudorange measurements. The equation linking the receiver position and satellite can be structured as per the following formula using least squares (LS) method:

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

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

To better represent the quality of each measurement based on the information measured by the receiver, the weightings of each satellite are needed. The weightings for each satellite are calculated using the formulation in [35] by integrating the SNR and satellite elevation. Finally, the GNSS receiver position can be estimated using WLS method as:

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

The weighting is given as follows [35]:

$$\boldsymbol{W}^{(i)}(ele_i, SNR_i) =$$

$$\frac{1}{sin^2 ele_i}\left(10^{-\frac{(SNR_i - T)}{a}}\left(\left(\frac{A}{10^{-\frac{(F-T)}{a}}} - 1\right)\frac{(SNR_i - T)}{F-T} + 1\right)\right) \tag{12}$$

where $\mathbf{W}^{(i)}(ele_i, SNR_i)$ denotes the weighting for satellite $\boldsymbol{SV}_i$.The parameter T indicates the threshold of SNR and is equal to $SNR_{threshold}$. Parameter a, A and F in (12) are experimentally determined. Then, the weighting matrix $\boldsymbol{W}$ is a diagonal matrix constituted by the weightings $\boldsymbol{W}^{(k)}(ele_i, SNR_i)$.

V. Experimental Results and Discussions

To evaluate the performance of the proposed method, two experiments conducted in two separate scenarios, are presented in this section. Firstly, the experiment setup is introduced in subsection A. Experimental validations in two typical urban canyons are presented in subsection C and D, respectively. The relationship between the satellite elevation angle and NLOS error is presented in subsection D before the discussion is given in subsection E.

A. Experiment Setup

Experiments are conducted in two typical urban canyons (urban canyon 1 and urban canyon 2) of Hong Kong, and the experimental scenes are shown in Fig. 7. The Skymask in the right-hand side demonstrates the degree of urbanization.

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 3D point clouds scanned from the surroundings. Both the u-blox M8T receiver and the 3D LiDAR are installed on the top of an experiment vehicle, which can be seen in left-hand side of Fig. 7. The data were collected at a frequency of 1 Hz for GNSS and 10 Hz for the 3D LiDAR.

In addition, the NovAtel SPAN-CPT, GNSS RTK/INS (fiber optic gyroscopes) integrated navigation system is used to provide the ground truth of positioning. All the data are collected and synchronized using the Robot Operation System (ROS) [36]. Moreover, the coordinate systems of all the sensors are calibrated before the experiments.

The parameters used in this paper, which are experimentally determined, are shown in TABLE I. Three GNSS positioning methods are compared:

(1) WLS: GNSS positioning with the WLS.
(2) WLS-NE: WLS with NLOS exclusion.
(3) WLS-NC: WLS with NLOS correction.

TABLE I
PARAMETER VALUES USED IN THIS PAPER

Parameters	$S_{threshold}$	$SNR_{threshold}$	ele_{thres}	θ_{thres}
Value	10	45 dB-Hz	20°	5°
Parameters	a	A	F	
Value	30	32	10	

B. Evaluation of the Proposed Method in Urban Canyon 1

Fig. 8 and TABLE II show the positioning results comparison of the conventional WLS, WLS-NE and the proposed method.

As can be seen from Fig. 8, the total satellites numbers fluctuate between 5 and 13, with a mean satellite number of 10 during the experiment. With the aid of the proposed NLOS correction method, the positioning performance is improved at most of the epochs, which is indicated by the blue curve in the bottom panel of Fig. 8. 30.29 meters of mean positioning error and 19.86 meters of standard deviation were obtained using the WLS method without any NLOS exclusion or correction. After the NLOS exclusion (all the NLOS are excluded), the mean error goes up to 35.25 meters. The main reason for this increase is due to the distortion of satellites' geometric distribution. In other words, the HDOP increases accordingly. According to the experiment, approximately 2–6 satellites are classified as NLOS due to the blockage from surrounding buildings. Therefore, the availability of GNSS positioning is decreased to about 92.5% due to the lack of satellites (at least five satellites are needed for GPS/BeiDou-based positioning calculation). The positioning error decreases to 22.86 meters using the proposed NLOS correction method. Moreover, the availability of GNSS positioning is also guaranteed. This result shows that the proposed NLOS correction model can obtain improved GNSS positioning performance.

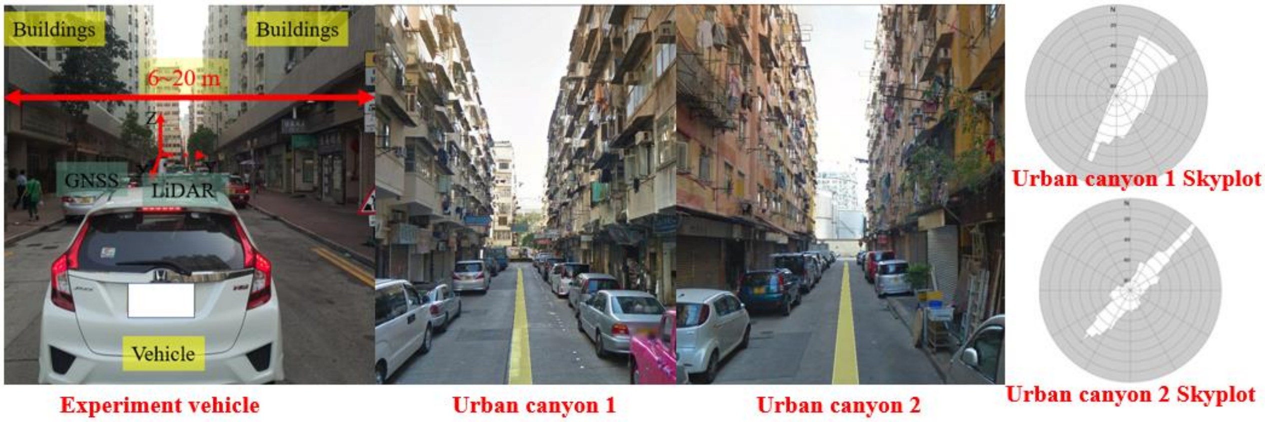

Fig. 7. The sensors setup of the vehicle and tested environment: GNSS and LiDAR sensors are installed on the top of the vehicle shown in the left side of the figure. The two tested urban scenarios are shown in the middle of the figures. The Skyplot of the two experiments is shown in the right side of the figure.

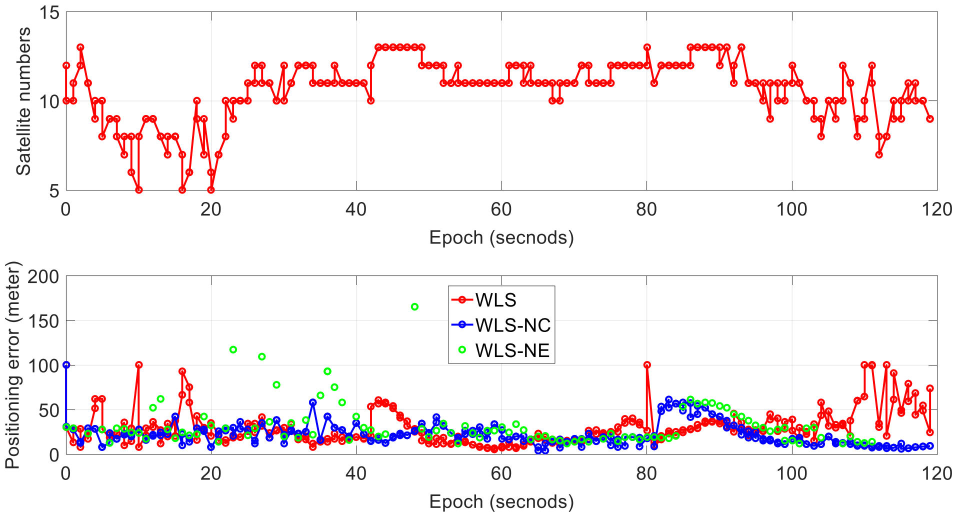

Fig. 8. Positioning error of the GNSS before and after adding the NLOS correction, and NLOS exclusion in the urban canyon 1. The top panel indicates the satellites numbers. The bottom panel shows the positioning error: red curve indicates the positioning error using WLS, blue curve denotes the positioning based on proposed NLOS correction. The green curve shows the result using WLS-NE.

TABLE II
POSITIONING PERFORMANCE OF THE TWO METHODS IN URBAN CANYON 1 (IN THE UNIT OF METER)

All data	**WLS**	**WLS-NE**	**WLS-NC**
Mean error	30.29	35.25	22.86
Std	19.86	57.49	13.17
Availability	100%	92.5%	100%

C. Evaluation of the Proposed Method in Urban Canyon 2

Fig. 9 and TABLE III show the positioning results comparison of the discussed three methods.

As can be seen from Fig. 9, the total satellites numbers fluctuate between 8 and 15 with a mean satellite number of 11 during the experiment. With the aid of the proposed NLOS correction method, the positioning performance is improved through almost all the experiment. 42.15 meters of mean positioning error and 21.29 meters of standard deviation are obtained using the WLS method without any NLOS exclusion or correction. After the NLOS exclusion (all the NLOS are excluded), the mean error goes up to 47.74 meters. According to the experiment, approximately 3–7 satellites are classified as NLOS due to the blockage from surrounding buildings. Therefore, the availability of GNSS positioning is decreased to 88.69%. The mean positioning error is decreased to 26.7 meters using the proposed NLOS correction method. Moreover, the availability of GNSS positioning is also guaranteed. Interestingly, we can find that the variation trends of positioning error using two separate solutions are quite similar although the experiment.

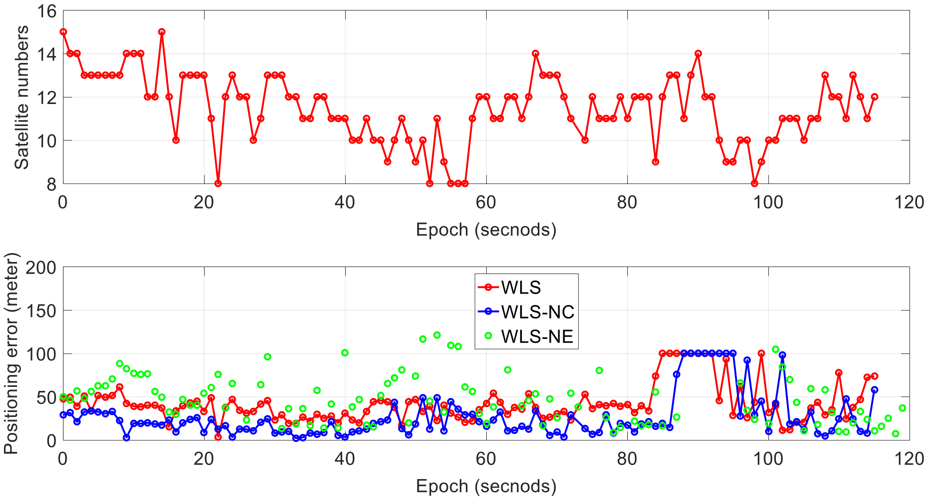

Fig. 9. Positioning error of the GNSS before and after adding the NLOS correction in the urban canyon 2. The top panel indicates the satellites numbers. The bottom panel shows the positioning error: red curve indicates the positioning error using WLS, blue curve denotes the positioning based on proposed NLOS correction. The green curve shows the result using WLS-NE.

TABLE III
POSITIONING PERFORMANCE OF THE TWO METHODS IN URBAN CANYON 2 SCENARIO (IN THE UNIT OF METER)

All data	**WLS**	**WLS-NE**	**WLS-NC**
Mean error	42.15	47.74	26.70
Std	21.29	29.34	24.32
Availability	100%	88.69%	100%

D. Analysis of Satellite Elevation Angles versus NLOS Errors

Our previous work in [13] shows that the potential NLOS error is positively correlated to the satellite elevation angle. In other words, the NLOS satellite with higher elevation can cause larger GNSS signal transmission delay. To show the relationship of GNSS positioning error and satellite elevation angle, we apply the manual correction in the collected data. Only satellites whose elevation angles are in a certain elevation angle range are corrected. The objective is to analysis the percentages of NLOS errors contributed by each elevation angle range of satellites.

TABLE IV shows the results of three separate NLOS correction tests of urban canyon 1. Three satellite elevation angle ranges are given. If satellites 2, 5 and 24 are corrected whose elevation angles are between 20° and 36°, 5.5 meters of improvement are obtained with a mean positioning error of 24.79 meters. More than half of the results possess errors less than 15 meters. If satellites 29 and 88 are corrected whose elevation angles are between 36° and 54°, 1.65 meters of improvement are obtained with a mean positioning error of 28.64 meters. Only 17.01% of the results possess errors less than 15 meters. There is almost no improvement if only satellite 13 is corrected whose elevation angle is between 54° and 72°.

TABLE V shows the results of three separate NLOS correction tests of urban canyon 2. Firstly, if the satellites 8, 17, 22 and 28, whose elevation angles are between 18°~36°, are corrected with the proposed method, the mean positioning error is decreased from 42.15 meters to 29.93 meters compared with the WLS method. 12.22 meters of improvement is obtained. Interestingly, the corresponding standard deviation also increases slightly. Approximately 79.64% of the positioning results have an error which is less than 30 meters. Secondly, only one satellite, satellite 88, possess an elevation angle which is between 36°~54° and is NLOS. A slight improvement is introduced after the correction with a mean positioning error of 41.95 meters and a standard deviation of 21.80 meters respectively. 0.2 meters of improvement is obtained. Moreover, the percentage of positioning results which is more than 40 meters is similar to the results from WLS. Thirdly, two satellites, satellites 30 and 99 with an elevation which is between 54°~72°. A slight improvement (0.14 meters) is obtained with the proposed NLOS corrections. The corresponding percentages is similar to the result from the NLOS correction of elevation range (36°~54°). In summary, the NLOS satellites with lower elevation (18°~36°) introduce larger positioning errors, compared with the NLOS satellites with higher elevation (36°~72°).

TABLE IV
POSITIONING PERFORMANCE OF WLSP-NC WITH MANUAL SATELLITE CORRECTION (IN THE UNIT OF METER) IN URBAN CANYON 1

All data	**Elevation (20°~36°)**	**Elevation (36°~54°)**	**Elevation (54°~72°)**
Mean error	24.79	28.64	30.1
Std	17.18	15.8	16.24
Percentage (<15 meters)	51.62%	17.01%	16.69%
Percentage (<30 meters)	84.66%	43.80%	40.43%
Percentage (>40 meters)	10.46%	25.70%	29.57%
Improvement	5.5	1.65	0.19
NLOS Satellites PRN	2,5,24	29,88	13

TABLE V
POSITIONING PERFORMANCE OF WLSP-NC WITH MANUAL SATELLITE CORRECTION (IN THE UNIT OF METER) IN URBAN CANYON 2

All data	Elevation (20°~36°)	Elevation (36°~54°)	Elevation (54°~72°)
Mean error	29.93	41.95	42.01
Std	24.62	21.80	21.81
Percentage (<15 meters)	51.32%	7.96%	8.03%
Percentage (<30 meters)	79.64%	43.36%	39.29%
Percentage (>40 meters)	15.04%	28.32%	30.36%
Improvement	12.22	0.2	0.14
NLOS Satellites PRN	8,17,22,28	88	30,99

E. Discussion

1. When comparing the results from the urban canyon 1 and urban canyon, we can find that the improvement in urban canyon 2 is more distinct (from 42.15 meters to 26.70, 36.7% of improvement in total) than that in urban canyon 1 (from 30.29 meters to 22.86, 24.5% of improvement in total).
2. The remaining GNSS positioning errors still 22.86 meters in middle urban and 26.70 meters, respectively. The major reason behind is that the potential GNSS multipath contributes to the remaining error. On the other hand, the performance of the NLOS correction relies on the performance of distance α_i (from GNSS receiver to the reflector) estimation, as the signal from the NLOS satellite can be reflected by different buildings. As shown in Fig. 11, the NLOS can be caused by both building A and building B (reflector), thus causing different α_i. The proposed NLOS correction method can mis-identify the reflector in some ways. In this case, this mis-identification can result in positioning error. Thus, the reflector detection will be studied in the future work to improve the performance of the proposed NLOS correction method.
3. According to our previous research [37], the dynamic objects (such as double-decker bus) on the road can also cause NLOS receptions. The effects from the dynamic objects are not modeled which can also contribute to the remaining GNSS positioning error.
4. **Performance sensitivity of the proposed method against building heights errors**:

As the proposed method employs the building height list to extend the detected TEBs to the exact height. We implement an offline simulation to analyze the performance sensitivity of the proposed method against building heights errors. We collect 6 hours of satellite ephemeris data and do the LOS/NLOS classification based on 3D building models in Hong Kong. The building models are manually added with heights noise which is subject to Gaussian distribution ($\boldsymbol{N}(\mu, \delta^2)$). The offline processing setup is as follows:

(1) **Applied satellites**: GPS/BeiDou/GLONASS/Galileo.

(2) **Ground truth for LOS/NLOS classification**: the LOS/NLOS classification based on original 3D building models are treated as ground truth.

(3) **Accuracy of LOS/NLOS classification**: detected NLOS satellites number denoted by $N_{NLOS}^{Detected}$, ground truth number of NLOS satellites denoted by N_{NLOS}^{Label}. Accuracy (P_s) is calculated by $P_s = N_{NLOS}^{Detected}/N_{NLOS}^{Label} * 100\%$.

(4) The position of the assumed GNSS receiver is accurately set.

The detail result is shown in Fig. 10 over the 6 hours of ephemeris. 4 kinds ($\delta = 2, 4, 6, 8$) of building heights noise model (mean μ equals to zero) are applied. The percentages are shown in TABLE VI. About 99.07 % of the LOS/NLOS classification is obtained. With the increased noise standard deviation, the accuracy decreases gradually. However, even when the building heights noise standard deviation reach 8 meters, accuracy still can reach 95.47%.

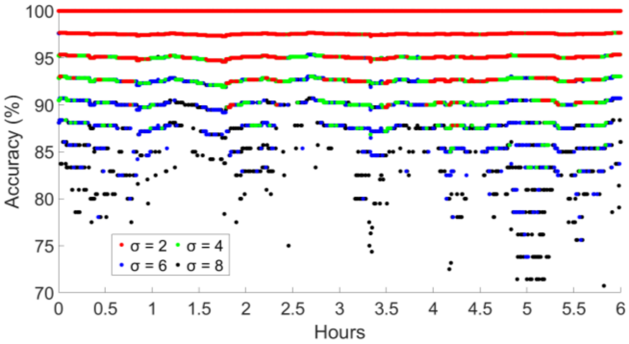

Fig. 10 Relationship between the NLOS detection accuracy and simulated buildings heights error. The simulated building heights errors are subject to Gaussian distribution $\boldsymbol{N}(\mu, \delta^2)$. The color points denote the accuracy in different epochs. The solid lines represent fitted curves based on the result (in 6 hours) under different error noise models.

TABLE VI
PERFORMANCE SENSITIVITY OF PROPOSED METHOD AGAINST BUILDING HEIGHTS ERRORS

Yaw Bias	Mean NLOS Detection Accuracy	Std
$\boldsymbol{N}(0, (2)^2)$	99.07%	1.72%
$\boldsymbol{N}(0, (4)^2)$	97.81%	2.74%
$\boldsymbol{N}(0, (6)^2)$	96.69%	3.56%
$\boldsymbol{N}(0, (8)^2)$	95.47%	4.46%

We can conclude from the result that: 1) the building height can error can have a slight negative impact against the LOS/NLOS classification, thus can deteriorate the performance of the proposed method. 2) proper way to identify the height of buildings can increase the robustness of the proposed method.

5. **Performance sensitivity of the proposed method against yaw (heading) angle errors**:

In this paper, the yaw angle is derived by the highly accurate RTK GNSS/INS integrated navigation system. To analyze the performance sensitivity of the proposed method against yaw angle errors, we propose to manually add error noise to the yaw angle. The added error noise is subject to Gaussian distribution ($\boldsymbol{N}(\mu, \delta^2)$). After posing different noise magnitude ($\delta = 2, 4, 6, 8$) to the yaw angle, the NLOS detection accuracy is decreased accordingly. Be noted that the NLOS detection accuracy calculation is the same as that in subsection E-6.

If the yaw angle with noise error model $\boldsymbol{N}(\mathbf{0}, (\mathbf{1}^\circ)^2)$ is applied, the performance of the proposed method remains the same. However, the mean positioning error increased from 26.70 to 29.56 meters after increase the δ to $\mathbf{2}^\circ$. The NLOS detection accuracy is also reduced to 98.2%. If the δ is set as $\mathbf{4}^\circ$ and $\mathbf{6}^\circ$, the NLOS detection accuracies are decreased to 95.43% and 93.01%, respectively. Meanwhile, the mean positioning errors and standard deviations are slightly increased. We can conclude that: the yaw angle error can have a negative impact on the performance of the proposed method.

TABLE VII
PERFORMANCE SENSITIVITY OF PROPOSED METHOD AGAINST YAW ERRORS IN URBAN CANYON 2 DATASET

Yaw Bias	NLOS Detection Accuracy	Mean positioning error (m)	Std
$\boldsymbol{N}(\mathbf{0}, (\mathbf{1}^\circ)^2)$	100%	26.70	24.32
$\boldsymbol{N}(\mathbf{0}, (\mathbf{2}^\circ)^2)$	98.2%	29.56	24.75
$\boldsymbol{N}(\mathbf{0}, (\mathbf{4}^\circ)^2)$	95.43%	30.41	24.93
$\boldsymbol{N}(\mathbf{0}, (\mathbf{6}^\circ)^2)$	93.01%	31.39	25.51
$\boldsymbol{N}(\mathbf{0}, (\mathbf{8}^\circ)^2)$	92%	31.10	27.39

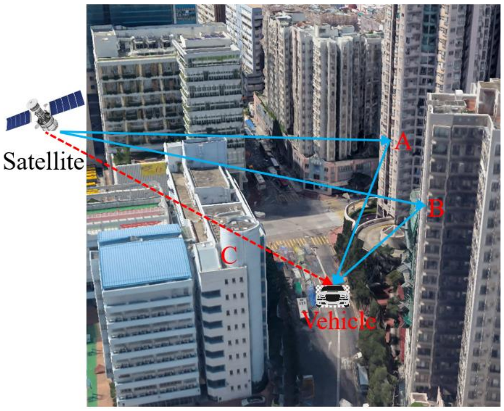

Fig. 11. Illustration of multiple NLOS signal transmission routes. The direct route from GNSS satellite to the vehicle is blocked by building C. However, the signal can be reflected by building A or building B. As a result, the vehicle can receive the GNSS signal as well.

VI. Conclusions

In this paper, we propose an NLOS correction and improved GNSS positioning method aided by 3D LiDAR. Innovatively, the top boundary of building is detected using the 3D LiDAR-based point cloud segmentation method and NLOS satellites are detected based on the detected TEBs. The NLOS are corrected using an NLOS error model, instead of direct exclusion. The GNSS positioning is conducted based on corrected and healthy LOS satellites. The evaluated results show that the proposed method can obtain improved GNSS positioning accuracy comparing with the standalone WLS.

The paper proposes to cope with the effects of static buildings on GNSS positioning using 3D LiDAR. In the future work, we propose to integrate a sky-pointing camera together with LiDAR to correct the NLOS receptions, therefore, to improve the GNSS positioning. As camera is able to capture the sky view in a real-time manner. In the case, the camera will play the role to describe Skyplot with obstacles and LiDAR is used to provide the distances between the vehicle and the obstacles. Moreover, the yaw angle in this paper can be provided by the LiDAR-based positioning [38].

Acknowledgment

The authors acknowledge the support of Hong Kong PolyU internal grant on the project G-YBWB, “Research on GPS Error Modelling Using 3D Point Cloud-Based Map for Autonomous Driving Vehicle”.

References

[1] L.-T. Hsu, Y. Gu, and S. Kamijo, "3D building model-based pedestrian positioning method using GPS/GLONASS/QZSS and its reliability calculation," (in English), *GPS Solutions*, vol. 20, no. 3, pp. 413–428, 2016. [Online]. Available: http://dx.doi.org/10.1007/s10291-015-0451-7.

[2] C. Rödel, S. Stadler, A. Meschtscherjakov, and M. Tscheligi, "Towards autonomous cars: the effect of autonomy levels on acceptance and user experience," in *Proceedings of the 6th International Conference on Automotive User Interfaces and Interactive Vehicular Applications*, 2014: ACM, pp. 1-8.

[3] J. Levinson *et al.*, "Towards fully autonomous driving: Systems and algorithms," in *Intelligent Vehicles Symposium (IV), 2011 IEEE*, 2011: IEEE, pp. 163-168.

[4] J. Wei, J. M. Snider, J. Kim, J. M. Dolan, R. Rajkumar, and B. Litkouhi, "Towards a viable autonomous driving research platform," in *Intelligent Vehicles Symposium (IV), 2013 IEEE*, 2013: IEEE, pp. 763-770.

[5] 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. 23286-23302, 2015.

[6] 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 T Veh Technol*, vol. 65, no. 6, pp. 4274-4287, 2016.

[7] J. Levinson and S. Thrun, "Robust vehicle localization in urban environments using probabilistic maps," in *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, 2010: IEEE, pp. 4372-4378.

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

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

[10] 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 *Proceedings of the 27th International Technical Meeting of the Satellite Division of the Institute of Navigation, ION GNSS 2014*, 2014, pp. 896-903.

[11] L.-T. Hsu, H. Tokura, N. Kubo, Y. Gu, and S. Kamijo, "Multiple faulty GNSS measurement exclusion based on consistency check in urban canyons," *Ieee Sens J,* vol. 17, no. 6, pp. 1909-1917, 2017.

[12] J. Breßler, P. Reisdorf, M. Obst, and G. Wanielik, "GNSS positioning in non-line-of-sight context—A survey," in *Intelligent Transportation Systems (ITSC), 2016 IEEE 19th International Conference on*, 2016: IEEE, pp. 1147-1154.

[13] L.-T. Hsu, "Analysis and modeling GPS NLOS effect in highly urbanized area," *GPS Solutions,* journal article vol. 22, no. 1, p. 7, November 04 2018, doi: 10.1007/s10291-017-0667-9.

[14] S. H. Kong, "Statistical Analysis of Urban GPS Multipaths and Pseudo-Range Measurement Errors," *IEEE Transactions on Aerospace & Electronic Systems,* vol. 47, no. 2, pp. 1101-1113, 2011.

[15] C. Pinana-Diaz, R. Toledo-Moreo, D. Betaille, and A. F. Gomez-Skarmeta, "GPS multipath detection and exclusion with elevation-enhanced maps," in *Intelligent transportation systems (ITSC), 2011 14th International IEEE Conference on*, 2011: IEEE, pp. 19-24.

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

[17] M. Adjrad and P. D. Groves, "Enhancing least squares GNSS positioning with 3D mapping without accurate prior knowledge," *Navigation,* vol. 64, no. 1, pp. 75-91, 2017.

[18] P. D. Groves, "Shadow Matching: A New GNSS Positioning Technique for Urban Canyons," *The Journal of Navigation,* vol. 64, no. 03, pp. 417-430, 2011, doi: doi:10.1017/S0373463311000087.

[19] L. Wang, P. D. Groves, and M. K. Ziebart, "GNSS Shadow Matching: Improving Urban Positioning Accuracy Using a 3D City Model with Optimized Visibility Scoring Scheme," *NAVIGATION, Journal of The Institute of Navigation,* vol. 60, no. 3, pp. 195-207, 2013.

[20] P. D. Groves and M. Adjrad, "Likelihood-based GNSS positioning using LOS/NLOS predictions from 3D mapping and pseudoranges," *GPS Solutions,* journal article vol. 21, no. 4, pp. 1805-1816, October 01 2017, doi: 10.1007/s10291-017-0654-1.

[21] M. Adjrad and P. D. Groves, "Intelligent Urban Positioning: Integration of Shadow Matching with 3D-Mapping-Aided GNSS Ranging," *Journal of Navigation*, vol. 71, no. 1, pp. 1-20, 2018, doi: 10.1017/S0373463317000509.

[22] S. Miura, L.-T. Hsu, F. Chen, and S. Kamijo, "GPS error correction with pseudorange evaluation using three-dimensional maps," *Ieee T Intell Transp,* vol. 16, no. 6, pp. 3104-3115, 2015.

[23] 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 *Intelligent Vehicles Symposium (IV), 2012 IEEE*, 2012: IEEE, pp. 184-190.

[24] T. Suzuki and N. Kubo, "Correcting GNSS multipath errors using a 3D surface model and particle filter," *Proc. ION GNSS+ 2013*, 2013.

[25] Y.-W. Lee, Y.-C. Suh, and R. Shibasaki, "A simulation system for GNSS multipath mitigation using spatial statistical methods," *Comput Geosci-Uk,* vol. 34, no. 11, pp. 1597-1609, 2008.

[26] J. I. Meguro, T. Murata, J. I. Takiguchi, Y. Amano, and T. Hashizume, "GPS multipath mitigation for urban area using omnidirectional infrared camera," *Ieee T Intell Transp,* vol. 10, no. 1, pp. 22-30, 2009.

[27] T. Suzuki, M. Kitamura, Y. Amano, and T. Hashizume, "High-accuracy GPS and GLONASS positioning by multipath mitigation using omnidirectional infrared camera," in *IEEE International Conference on Robotics and Automation*, 2011, pp. 311-316.

[28] J. S. Sánchez, A. Gerhmann, P. Thevenon, P. Brocard, A. B. Afia, and O. Julien, "Use of a FishEye camera for GNSS NLOS exclusion and characterization in urban environments," in *ION ITM 2016, International Technical Meeting*, 2016: ION.

[29] D. Maier and A. Kleiner, "Improved GPS sensor model for mobile robots in urban terrain," in *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, 2010: IEEE, pp. 4385-4390.

[30] A. Shetty and G. X. Gao, "Covariance Estimation for GPS-LiDAR Sensor Fusion for UAVs," in *ON GNSS+ 2017, Portland, OR, USA*, 2017.

[31] J. Levinson, M. Montemerlo, and S. Thrun, "Map-Based Precision Vehicle Localization in Urban Environments," in *Robotics: Science and Systems*, 2007, vol. 4: Citeseer, p. 1.

[32] M. Greenspan and M. Yurick, "Approximate kd tree search for efficient ICP," in *Fourth International Conference on 3-D Digital Imaging and Modeling, 2003. 3DIM 2003. Proceedings.*, 2003: IEEE, pp. 442-448.

[33] G. Barequet and S. Har-Peled, "Efficiently approximating the minimum-volume bounding box of a point set in three dimensions," *J Algorithm,* vol. 38, no. 1, pp. 91-109, 2001.

[34] E. Kaplan and C. Hegarty, *Understanding GPS: principles and applications*. Artech house, 2005.

[35] A. M. Herrera, H. F. Suhandri, E. Realini, M. Reguzzoni, and M. C. de Lacy, "goGPS: open-source MATLAB software," *Gps Solut,* vol. 20, no. 3, pp. 595-603, 2016.

[36] M. Quigley *et al.*, "ROS: an open-source Robot Operating System," in *ICRA workshop on open source software*, 2009, vol. 3, no. 3.2: Kobe, Japan, p. 5.

[37] 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 *Position, Location and Navigation Symposium (PLANS), 2018 IEEE/ION*, 2018: IEEE, pp. 158-165.

[38] 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-Basel,* vol. 18, no. 11, p. 3928, 2018.