The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Volume L-4/W1-2026
Free and Open Source Software for Geospatial (FOSS4G) 2026 – Academic Track "Bridging Geospatial Technology and Humanity",
30 August–5 September 2026, Hiroshima, Japan

A Pipeline for Low-Cost Wide-Area 3D Mapping Using LiDAR-Equipped Mobile Devices and Open Data

Ryosei Ueda[1], Daisuke Yoshida[1]

[1] Graduate School of Informatics, Osaka Metropolitan University, Osaka 558-8585, Japan − daisuke@omu.ac.jp

Keywords: 3D Point Cloud, Mobile LiDAR, Open Data, Point Cloud Registration, Georeferencing.

Abstract

The emergence of LiDAR-equipped mobile devices has enabled low-cost 3D point cloud acquisition. However, accurate 3D model reconstruction using such devices remains confined to individual, locally captured scans. This limitation makes wide-area coverage infeasible without additional correction. Existing methods for wide-area mapping require specialized equipment or high-accuracy base maps, which poses significant barriers in terms of cost and data infrastructure availability. This study proposes a low-cost pipeline for constructing wide-area 3D maps using only LiDAR-equipped mobile devices and open data. The proposed approach automatically registers multiple locally captured point clouds and assigns absolute coordinates to each by referencing open geospatial datasets. A key feature of the method is the decomposition of 3D spatial information into horizontal and vertical components. These components are processed independently to reduce errors and improve computational efficiency. Validation experiments in Tondabayashi City, Osaka, Japan confirmed a horizontal RMSE of 1.75 m or less and a vertical RMSE of 0.10 m or less. Both values meet the accuracy requirements for 1:2,500-scale disaster prevention base maps. The vertical accuracy in particular exceeds the stricter standard required for flood hazard mapping applications. These results demonstrate that combining mobile devices with open data enables non-experts to construct high-accuracy 3D maps suitable for disaster risk management applications.

1. Introduction

1.1 Background

Recent advances in 3D measurement technologies, including Light Detection and Ranging (LiDAR), have driven rapid growth in the adoption of 3D point cloud techniques for digitally representing real-world environments. These technologies are being applied across a wide range of domains, including civil engineering, construction, and urban planning, becoming increasingly central to the digital transformation of infrastructure asset management. Particularly in disaster prevention, 3D spatial data is gaining importance over conventional 2D hazard maps, as it enables more intuitive comprehension of damage conditions. Laefer et al. (2025) demonstrated that combining high-density point clouds acquired by terrestrial LiDAR with WebGL-based rendering technology enables dynamic 3D visualization of inundation scenarios at the street level. A comparative user study found that 92% of participants rated the 3D visualization as more effective than conventional 2D maps. 3D visualization of disaster damage has thus become a critical component in encouraging proactive evacuation behavior among residents and supporting the development of more effective disaster prevention plans.

3D spatial data is increasingly being released to the public by governments, research institutions, and local authorities, as exemplified by Project PLATEAU (MLIT, 2026) and the 3D Elevation Program (USGS, 2026). However, these datasets are inconsistent in coverage and level of detail across regions. In Project PLATEAU, for example, building geometries are highly detailed in some areas, while structures are presented as simple untextured volumes in a considerable number of regions. Furthermore, many areas still lack high-accuracy 3D data altogether. To enable high-accuracy analysis in such regions, where 3D data is either absent or of insufficient resolution, individual organizations must independently conduct their own data acquisition.

Traditionally, high-accuracy point cloud acquisition has required dedicated equipment such as Mobile Mapping Systems (MMS) or Terrestrial Laser Scanners (TLS), which impose substantial financial barriers. Recent technological advances, however, have introduced LiDAR-equipped mobile devices, such as the iPhone Pro series, enabling relatively affordable and accessible point cloud acquisition. Nevertheless, LiDAR-equipped mobile devices are subject to the incremental accumulation of small localization errors during scanning. These errors result in significant geometric distortion in the reconstructed 3D models over large areas. Consequently, high-accuracy 3D data acquisition using LiDAR-equipped mobile devices remains confined to individual, locally captured scans.

1.2 Related Work

Several studies have proposed approaches to this challenge: integrating point clouds into urban digital twins (Yamazaki et al., 2026), merging iPad LiDAR datasets with static GNSS equipment (Saptari et al., 2024), and reconstructing terrain models using iPhone LiDAR combined with total stations and GNSS receivers (Krausková et al., 2025). All of these approaches, however, rely on either pre-existing high-accuracy base maps or costly surveying instruments. A solution enabling anyone to construct 3D maps using only low-cost devices has yet to be realized.

1.3 Objectives

This study proposes a low-cost pipeline for constructing wide-area 3D maps by integrating multiple locally captured point clouds, using only LiDAR-equipped mobile devices and open data. The goal is to enable non-experts to construct practical 3D maps under low-cost conditions. The novelty of the proposed method lies in the integration of multiple open-source technologies into a custom processing pipeline that is robust to the noise and errors inherent in mobile scanning. In particular, the approach adopts a strategy of decomposing the registration and georeferencing of 3D space into horizontal and vertical components, optimizing each independently in a stepwise manner. To support disaster prevention applications such as disaster risk education and 3D hazard mapping, accuracy targets are defined as follows. Horizontal accuracy is set to an RMSE of 1.75 m or less, the accuracy required for base maps used in landslide hazard mapping at a scale of 1:2,500, while vertical accuracy is set to an RMSE of 0.30 m or less, a level sufficient for flood hazard mapping.

2. Proposed Methodology

2.1 Registration

The conventional approach combining Fast Point Feature Histograms (FPFH) and Random Sample Consensus (RANSAC) (Rusu et al., 2009) was evaluated on the dataset used in this study. With default parameters, the algorithm failed to terminate within a feasible time. When parameters were tuned to limit computation, the algorithm consistently returned an identity matrix, indicating no valid transformation was found. These failures confirm that directly estimating a transformation matrix in 3D space is unreliable in outdoor environments, where geometric features are sparse and mobile LiDAR data suffers from noise and poor fine-scale reproducibility. To address these challenges, the proposed method divides the registration process into three phases, as illustrated in Figure 1. After preprocessing in Phase 1, Phase 2 performs optimization in the horizontal plane; Phase 3 performs optimization along the vertical axis.

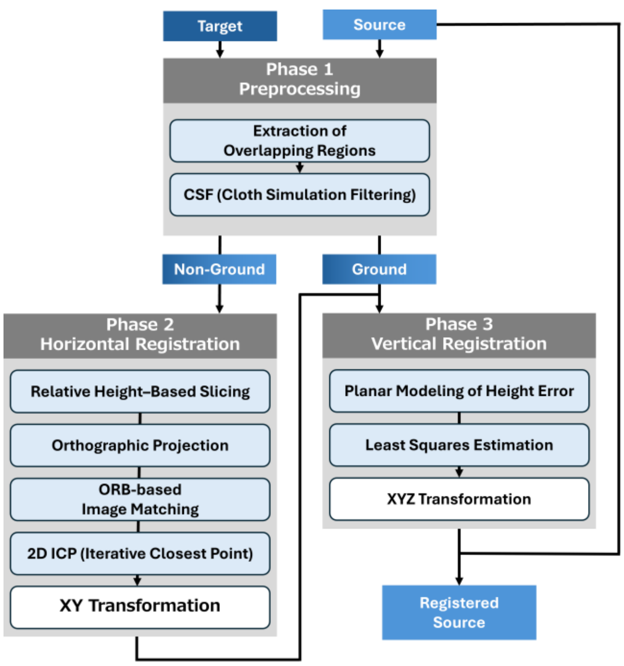

Figure 1. Workflow for registration in the proposed method.

First, in Phase 1, overlapping regions are extracted from both the source and target point clouds based on the approximate geographic information each carries. While LAS data acquired with mobile devices contains global positioning information referenced to the UTM coordinate system (WGS 84 / UTM zone 53N, EPSG:32653), it also includes GNSS positioning errors on the order of several meters. Nearest-neighbor search is therefore performed on 2D point clouds projected onto the XY plane. This approach eliminates the influence of vertical errors and enables accurate identification of horizontally overlapping regions. A KD-Tree is constructed from the 2D target point cloud using scipy.spatial.cKDTree from SciPy (Virtanen et al., 2020). Each point in the source cloud is then queried against this tree, and the distance to its nearest neighbor is computed. Points whose nearest-neighbor distance falls below a defined threshold are extracted using the select_by_index method from Open3D (Zhou et al., 2018). The same procedure is applied to the source point cloud by constructing a corresponding KD-Tree, thereby extracting only the mutually overlapping regions from both clouds (Figure 2). This process reduces the data volume for subsequent stages, which significantly decreases computational cost and mitigates the risk of local optima caused by false correspondences.

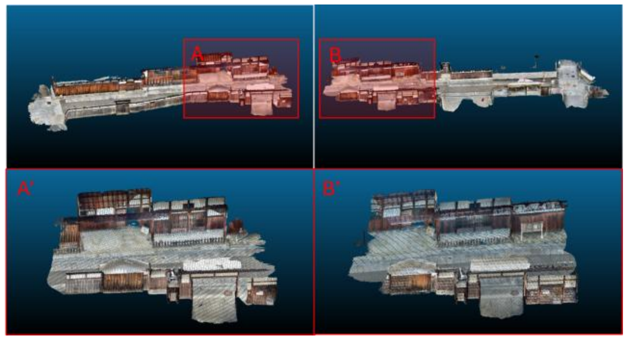

Figure 2. Extraction example of overlapping regions (Top: Overlapping regions A and B of adjacent point clouds, Bottom: Extracted corresponding regions A' and B').

As a preliminary step toward performing horizontal and vertical registration independently, the point cloud extracted from the overlapping region is separated into ground and non-ground point clouds. The Cloth Simulation Filter (CSF) (Zhang et al., 2016) is adopted as the separation algorithm, owing to its relatively few parameter requirements and high adaptability to diverse terrain conditions. CSF is a method based on a physical simulation in which the point cloud data are inverted vertically and a virtual cloth is allowed to fall naturally from above. The position at which the cloth comes to rest after colliding with the inverted terrain is approximated as the digital terrain model (DTM); points within a prescribed distance from the cloth mesh are classified as ground points, while all remaining points are classified as non-ground points. The CSF parameters applied in this study are as follows: Cloth Resolution = 1.0, Rigidness = 3, and Classification Threshold = 0.5. This process reduces the influence of non-uniform point cloud density inherent to mobile devices and suppresses excessive fitting to fine surface irregularities such as roadside ditches and minor undulations. As a result, the global terrain trend is successfully captured (Figure 3).

Figure 3. Separation example of point cloud using CSF (Left: Original point cloud, Top right: Non-ground points, Bottom right: Ground points).

https://doi.org/10.5194/isprs-archives-L-4-W1-2026-319-2026 |

Next, regarding Phase 2, a point cloud slice is extracted from the non-ground point cloud. Using a KD-Tree, the nearest ground point on the XY plane is identified for each non-ground point, and its Z coordinate is used as the reference for computing relative height. To eliminate near-ground vegetation and noise, as well as measurement errors at elevated positions, a slice of 0.2 m width is extracted from the non-ground point cloud within a relative height range of 0.7 m to 2.5 m (Figure 4). The slice is extracted using the select_by_index method of Open3D.

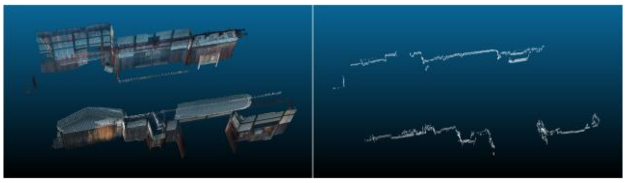

Figure 4. Extraction example of point cloud slice (Left: Non-ground points, Right: Extracted slice).

The extracted point cloud slice is orthographically projected onto a 2D plane and converted into a binary image through the following steps. First, the overall minimum coordinates are subtracted from the XY coordinates to offset the point cloud to the origin. The coordinates are then scaled according to a spatial resolution of 0.045 m/pixel, converting real-world coordinates into pixel coordinates. Finally, a NumPy (Harris et al., 2020) array is generated by mapping the presence or absence of points to pixel values (Figure 5).

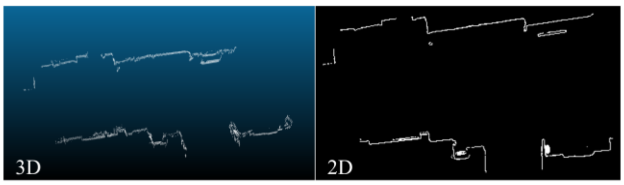

Figure 5. Orthographic projection example of point cloud slice onto 2D plane (Left: 3D point cloud, Right: 2D binary image).

Using the wall slice images of the source and target generated by the aforementioned procedure, initial registration in the horizontal direction is performed. Feature-based matching methods such as Scale-Invariant Feature Transform (SIFT) (Lowe, 2004) and Speeded-Up Robust Features (SURF) (Bay et al., 2006) are widely employed for image matching. However, the images handled in this study are binary images projected from point clouds and thus contain limited texture information. In contrast, geometric corners derived from building wall surfaces appear prominently in these images. For this reason, Oriented FAST and Rotated BRIEF (ORB) (Rublee et al., 2011), which specializes in corner detection, is adopted. Since ORB defines the principal orientation of keypoints using the intensity centroid, it enables stable feature extraction even under rotational variations. The similarity between extracted keypoints is evaluated using the Hamming distance. Mismatches are subsequently removed based on descriptor similarity, and the optimal rigid transformation matrix is estimated using the RANSAC algorithm (Figure 6). OpenCV (Bradski, 2000) is used to implement the series of image processing and feature matching operations described above.

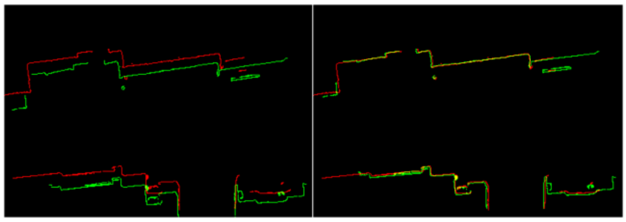

Figure 6. 2D image registration example using ORB feature matching (Left: Before registration, Right: After registration; Red: Target, Green: Source).

During initial registration, the point cloud is converted into a discrete pixel-based image for processing. Therefore, the initial registration inevitably contains sub-pixel errors. To address this, Iterative Closest Point (ICP) (Besl et al., 1992) is applied to the previously extracted point cloud slice, using the transformation estimated in the preceding step as an initial value. This enables fine-grained correction of horizontal displacement. However, applying ICP directly in 3D space carries the risk that vertical errors may interfere with horizontal convergence. To mitigate this, a preprocessing step is introduced in which the Z-coordinates of the slice are set to zero, generating a pseudo-2D point cloud. This restricts the optimization target to horizontal translation and rotation about the vertical axis only. The ICP implementation uses the registration_icp method of Open3D, with TransformationEstimationPointToPoint adopted as the objective function. The correspondence distance threshold is set to an extremely small value of 0.75 times the image resolution to suppress false correspondences to local noise. Additionally, the convergence criteria are tightened via ICPConvergenceCriteria to ensure that iterations continue even when improvements are marginal. Through this process, erroneous convergence to local noise is eliminated, and accurate horizontal registration is achieved (Figure 7).

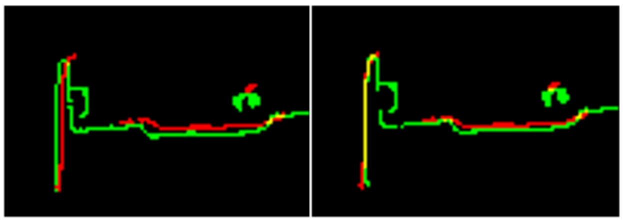

Figure 7. 2D image registration example using 2D ICP (Left: Before registration, Right: After registration; Red: Target, Green: Source).

In Phase 3, the ground point cloud is used to correct positional offset and tilt along the Z-axis. Using scipy.spatial.cKDTree, the nearest target ground point $t_i \in T$ on the XY-plane is identified for each point s_i in the horizontally registered source ground point cloud S. Correspondence pairs with a horizontal distance of less than 0.5 m are then extracted. The elevation difference Δz_i between each pair reflects not only translational offset but also tilt-induced errors. Therefore, the elevation error is modeled using the following first-order planar equation:

$$\Delta z_i = z_{t_i} - z_{s_i} \approx a\, x_{s_i} + b\, y_{s_i} + c \tag{1}$$

where Δz_i = elevation difference for correspondence pair i
z_{t_i}, z_{s_i} = Z coordinates of target point t_i and source point s_i, respectively
x_{s_i}, y_{s_i} = X and Y coordinates of source point s_i
a, b = tilt coefficients along the X-axis and Y-axis
c = vertical translational component

The optimal parameters $\hat{a}, \hat{b}, \hat{c}$ are estimated using numpy.linalg.lstsq. The Z-coordinates of the entire source point cloud are then updated using the estimated parameters, achieving high-accuracy registration in both horizontal and vertical directions. This multi-stage approach, which constrains the search space at each step, enables robust registration against mobile device noise while keeping computational cost low.

2.2 Georeferencing

Absolute coordinates are assigned to the 3D point cloud that has been aligned and integrated through the registration process described in the previous section. High-accuracy georeferencing of 3D point clouds conventionally requires ground control points measured by total stations or equivalent surveying instruments. However, reliance on such equipment poses a significant barrier to achieving the primary objective of this study: enabling non-experts to construct practical 3D maps under low-cost conditions. Therefore, this method adopts the Fundamental Geospatial Data (Basic Items) (GSI, 2026a) and the Fundamental Geospatial Data (Digital Elevation Model) (GSI, 2026b), both published by the Geospatial Information Authority of Japan (GSI). These datasets are publicly available on the Internet and can be obtained free of charge, thereby minimizing the economic cost of implementation. Furthermore, since anyone can access the identical data, the reproducibility of the method is ensured. As illustrated in Figure 8, the georeferencing procedure separates 3D space into horizontal and vertical components. Position determination is performed in a stepwise manner. Note that the XML-format data obtained from the Fundamental Geospatial Data undergoes preprocessing prior to input into this procedure. Specifically, road edge data is converted to Shapefile format, and the Digital Elevation Model (DEM) is converted to GeoTIFF (.tif) format before use.

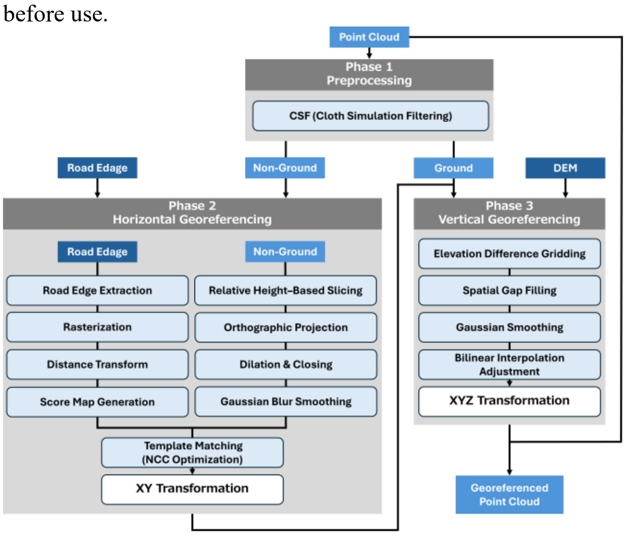

Figure 8. Workflow for georeferencing in the proposed method.

In Phase 1, the CSF algorithm is applied to the integrated point cloud to separate it into ground points and non-ground points.

In Phase 2, horizontal position determination is performed. A wall slice image is generated from the integrated point cloud using a similar method to that described in the previous section. In order to eliminate the influence of terrain undulation and to remove matching obstacles such as roadside vegetation and overhead roofs, only the non-ground points located at a relative height between 1.0 m and 1.5 m above the ground are extracted as a slice. The extracted point cloud is orthographically projected onto the XY plane at a spatial resolution of 0.10 m/pixel to generate a binary image.

Images generated from mobile LiDAR point clouds consist of discrete pixel distributions and tend to contain discontinuities and partial data gaps. To improve the stability of the subsequent template matching process, morphological operations (dilation and closing) and Gaussian blur are applied to generate a continuous and smooth wall slice image that is robust to spatial density variations in the point cloud.

Next, a reference image for template matching is generated from the road edge vector data provided by GSI. Using the R-tree spatial index of GeoPandas (Van den Bossche et al., 2024), the road edge data intersecting the bounding box of the integrated point cloud is retrieved and clipped. The extracted vector data is then rasterized as a binary image with the same resolution and coordinate system as the wall slice image. However, simple binary image matching between line segments presents a critical problem: even a slight positional offset results in zero overlap area, making it impossible to obtain a search gradient toward the optimal solution. To address this issue, a distance transform is applied to the road edge image using OpenCV. For each pixel in the image, the Euclidean distance d to the nearest road edge segment is computed. This distance is then converted into a similarity score V using the following equation:

$$V = \frac{1}{1 + \alpha \cdot d} \tag{2}$$

where V = similarity score
d = Euclidean distance to the nearest road edge
α = decay coefficient

In this implementation, α is set to 0.10. This transformation generates a score image in which the maximum value is assigned directly to the road edge, with values smoothly decaying as the distance increases (Figure 9). The spatial gradient of this score image functions as an attractive force that pulls the solution toward the correct position during optimization. This significantly improves convergence while preventing entrapment in local optima.

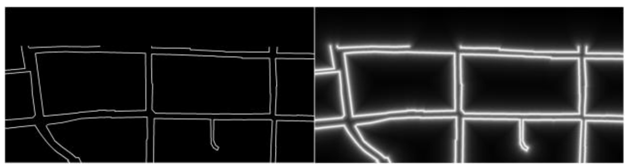

Figure 9. Generation example of a road edge score image (Left: Before transformation, Right: After transformation).

The two generated images are used to search for the optimal translation and rotation. Specifically, the wall slice image is rotated at each candidate angle, and template matching is performed against the road edge score image. The orientation of the point cloud captured by the device is approximately accurate. Therefore, the search range for the rotation is set to ±5.0 degrees with a step size of 0.25 degrees to reduce computational cost and suppress false matches. Normalized Cross-Correlation (NCC) is adopted as the similarity evaluation function, as it is robust against differences in pixel value scale and spatial variations in point cloud density. The pixel position that maximizes the NCC score is identified. A 2D transformation matrix is then constructed from the resulting optimal parameters. This transformation matrix is applied to the XY coordinates of the integrated point cloud to update the horizontal position (Figure 10).

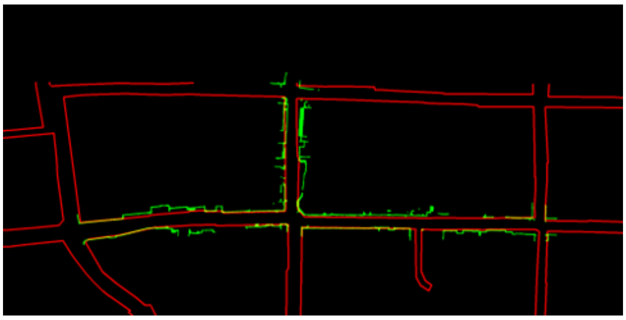

Figure 10. 2D top-down view of horizontal georeferencing result (Red: Road edges, Green: Point cloud).

In Phase 3, vertical position determination along the Z-axis is performed on the horizontally aligned integrated point cloud. The 5 m mesh Digital Elevation Model (DEM) provided by the GSI is used as reference data. Point clouds acquired by mobile LiDAR have a high spatial resolution on the order of centimeters. In contrast, the 5 m mesh DEM is a coarse grid dataset with 5 m intervals. Therefore, simply replacing the Z coordinates with the DEM elevation values would introduce grid-like or staircase artifacts and eliminate fine terrain variations. To address this issue, the proposed method adopts a smoothing correction approach that preserves the original point cloud geometry while fitting only the global elevation trend to the DEM.

First, the elevation difference information serving as the correction reference is computed. Ground points are extracted from the horizontally aligned integrated point cloud using CSF. The DEM elevation value corresponding to the XY coordinates of each ground point is retrieved using rasterio (Gillies et al., 2013). The elevation difference between the DEM values and the Z coordinates of the ground points is subsequently computed. The computed elevation differences are projected onto a grid with a spatial resolution of 2.0 m. The mean difference value within each cell is then calculated. Since the grid contains empty cells where no point cloud data exists, gap-filling is performed using the NearestNDInterpolator from SciPy to construct a spatially continuous grid.

Next, a large-kernel Gaussian blur is applied to the constructed difference grid to remove stepwise discontinuities caused by the 5 m mesh resolution, yielding an elevation correction map that smoothly captures only the global slope and undulation trends of the terrain.

Finally, Z correction values for all points in the integrated point cloud are computed from the smoothed correction map through bilinear interpolation using the RegularGridInterpolator from SciPy. By adding these correction values to the original Z coordinates of the point cloud, the absolute elevation is aligned without compromising the original fine-scale terrain geometry.

3. Experimental Results and Evaluation

3.1 Experimental Setup and Dataset

To evaluate the accuracy of the proposed method, the area surrounding the former Sugiyama Residence in the Jinaimachi district, Tondabayashi City, Osaka Prefecture, Japan (Figure 11) was selected as the survey area. This area features continuous walls and fences throughout the district. Such an environment is well-suited for extracting geometric features from non-ground point clouds in the proposed method. Data acquisition was carried out using an iPad Pro 11-inch (4th generation) and the 3D scanning application Scaniverse. According to Luetzenburg et al. (2021), the iPhone LiDAR scanner computes depth by using a single-photon avalanche diode (SPAD) to detect a total of 576 pulse signals emitted from a near-infrared VCSEL. The maximum measurement range is 5 m, and an absolute accuracy of approximately ±1 cm can be achieved for objects larger than 10 cm on each side. It has also been reported that there is no difference in sensor performance between iPhone and iPad devices. Data acquisition was conducted along five scanning routes (Scan 1–5) within the survey area, as illustrated in Figure 12. Ground truth data was separately acquired using a high-performance 3D scanner (Matterport Pro 3) and a high-accuracy GNSS surveying instrument (Drogger RZX.D).

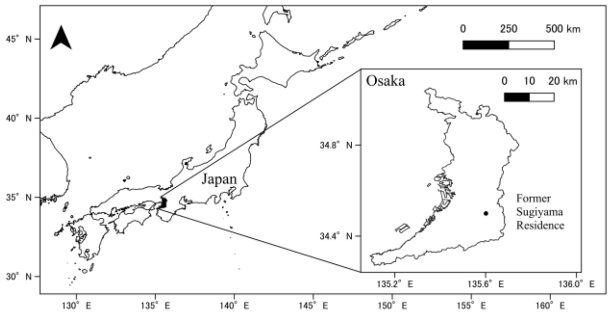

Figure 11. The location of the former Sugiyama Residence (indicated by the black point in the detailed map).

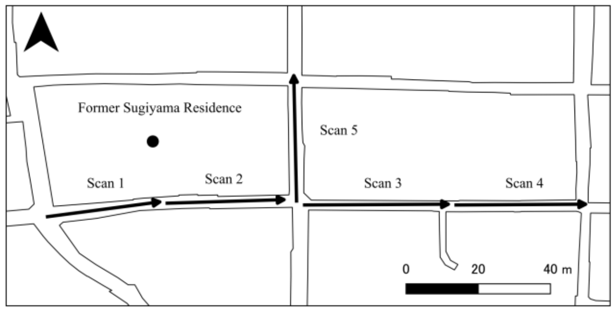

Figure 12. Scanning path in the survey area.

3.2 Evaluation Metrics and Target Values

The 3D map constructed by the proposed method was evaluated in two stages: relative accuracy, which reflects the quality of registration, and absolute accuracy, which reflects the quality of georeferencing. For relative accuracy, 10 feature points were selected in the overlapping regions between adjacent point clouds to ensure a uniform spatial distribution, and the RMSE was computed (Figure 13). To prevent bias toward a specific plane, feature points were extracted from both ground and non-ground points. These points were manually extracted from geometrically unambiguous locations, such as the corners where white plastered walls meet wooden structural members, and the corners of wooden entrance doors. Given the hardware limitation that the mobile LiDAR sensor exhibits reduced measurement accuracy for objects smaller than 0.10 m on each side (Luetzenburg et al., 2021), the target value was set to an RMSE of 0.10 m or less. For absolute accuracy, 30 feature points were selected from across the entire target area to avoid spatial bias, and the RMSE was computed by comparing the ground truth data with the 3D map constructed by the proposed method (Figure 14). As with relative accuracy, these points were extracted from both ground and non-ground points to prevent data concentration on a single plane. The target value for horizontal accuracy was set to an RMSE of 1.75 m or less, corresponding to the positional accuracy requirement for 1:2,500-scale mapping as defined by the Guidelines for Work Procedures as Specified in Article 34 of the Surveying Act (GSI, 2025b). This was based on the fact that the Guidelines for Creating Landslide Hazard Maps (MLIT, 2020) specify a base map scale of approximately 1:2,500 as the standard. The target value for vertical accuracy was set to a more stringent RMSE of 0.30 m or less, in anticipation of potential application to flood hazard mapping. The Manual for the Preparation of Flood Inundation Maps (4th Edition) (MLIT, 2017) recommends the use of high-accuracy terrain data. Based on this recommendation, this target aimed to ensure an elevation accuracy equivalent to that of DEM1A (1 m mesh), the highest-accuracy digital elevation model provided by GSI.

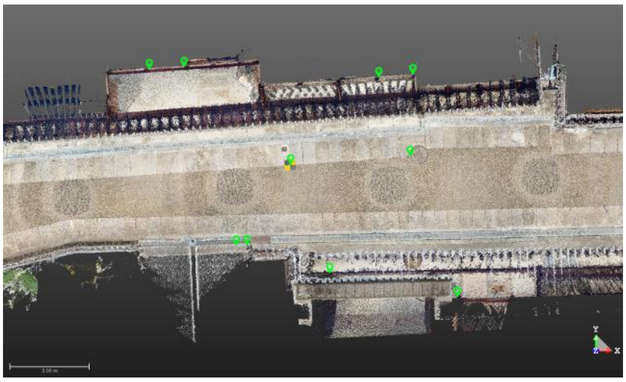

Figure 13. Example of selected feature points for relative accuracy.

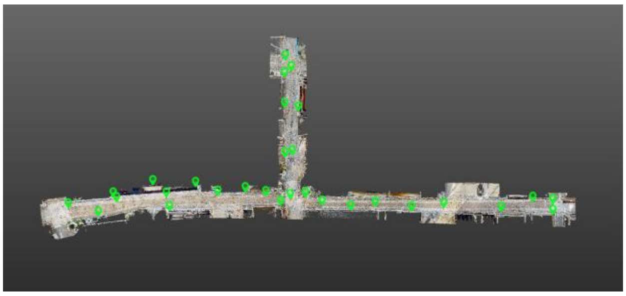

Figure 14. Selected feature points for absolute accuracy.

3.3 Results

The proposed registration method was applied to five captured point cloud pairs (Scans 1−5), and their relative accuracy was evaluated. As shown in Figure 15, the RMSE target was achieved for all point cloud pairs. Notably, Scans 2−3, 3−4, 2−5, and 3−5 demonstrated particularly high accuracy, with RMSE values within 0.05 m. Furthermore, the mean RMSE across all pairs was 0.048 m, confirming that stable registration accuracy was maintained throughout.

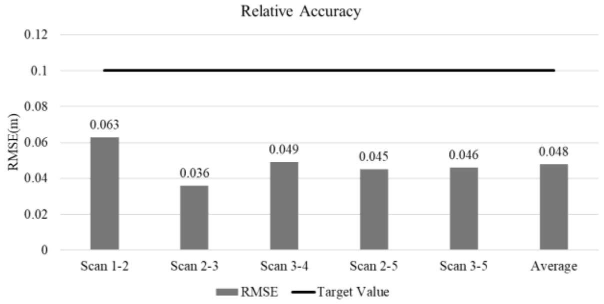

Figure 15. Evaluation of relative accuracy across five point cloud pairs.

The absolute accuracy of the georeferenced integrated point cloud was then evaluated. As shown in Figure 16, the RMSE of the point cloud constructed by the proposed method met the target values in both the horizontal and vertical directions. In particular, the vertical component achieved an RMSE within 0.10 m, yielding higher accuracy than that observed in the horizontal direction.

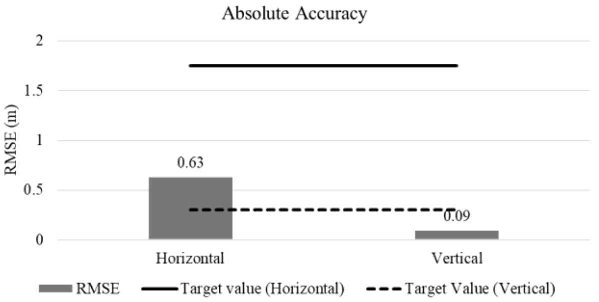

Figure 16. Evaluation of absolute accuracy in horizontal and vertical direction.

4. Discussion

4.1 Interpretation of results

Regarding relative accuracy, the primary factor contributing to this outcome was the spatial arrangement of structures in the target environment. In the overlapping regions of Scans 2−5, the distance between opposing walls was relatively narrow, measuring less than 4.6 m. This enabled the capture of surrounding point clouds with minimal device movement, which likely contributed to the high registration accuracy. In contrast, the overlapping region between Scans 1 and 2 had a wider inter-wall distance of approximately 7.0 m. The increased self-localization error of the mobile device during walking likely made it difficult to accurately reconstruct the spatial relationships between structures.

Regarding absolute accuracy, a high level of accuracy was achieved in the vertical direction. This accuracy level satisfies the criteria set forth in the Manual for Creating 3D Digital Topographic Map Data to Promote i-Construction (GSI, 2025a), which defines the standard for utilizing original data in and after the design phase. In the horizontal direction, however, the accuracy was limited to the 1:2,500-scale mapping level. The superior vertical accuracy can be attributed to the characteristic that in areas with little terrain undulation, vertical errors are less susceptible to the effects of increasing travel distance compared to horizontal errors. Figure 17 illustrates the relationship between travel distance and positional errors. After manually aligning the starting points of the captured 3D point cloud and the reference data, the accumulation of positional errors was evaluated. The results indicate that while the horizontal positional error increases with travel distance, the vertical positional error remains relatively stable.

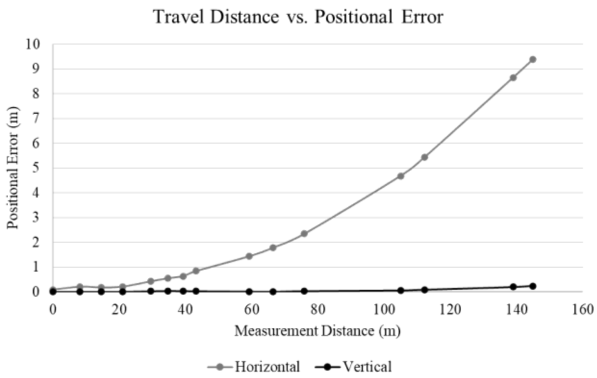

Figure 17. Travel distance vs. positional error: horizontal and vertical comparison.

4.2 Utility of the Proposed Method

The utility of the proposed method lies in its ability to construct 3D maps adaptable for disaster prevention applications using only LiDAR-equipped mobile devices and open data. This enables small municipalities and local residents with limited budgets and technical expertise to independently develop 3D geospatial data, thereby contributing to regional digital transformation.

Furthermore, the use of open data as reference data ensures reproducibility while minimizing economic costs. The applicability of this approach is not limited to Japan; it can be generalized to any region where equivalent road edge data and DEMs are available.

In addition, the entire pipeline is implemented using open-source Python libraries and produces output in standard formats including LAS/LAZ, enabling seamless integration with FOSS4G software such as QGIS. Such a reproducible workflow, as a practical example from the Asian region leveraging Japanese governmental open data, embodies the principles and objectives of FOSS4G in promoting the dissemination of open geospatial data.

4.3 Limitations

However, this method has several limitations that must be addressed before practical deployment. First, the validation environment is restricted; the experiments were conducted on a small-scale dataset, and accuracy has not been verified in steep terrain or wider-area environments. Second, processing parameters were not comprehensively examined, as the primary objective of this study was to propose the method and conduct fundamental validation. Third, the method is dependent on environmental structure, as it relies on walls and other structural features as registration cues, making it difficult to apply in environments with wide road widths and sparse structures. Fourth, cumulative drift and loop closure errors in larger-scale mapping remain unaddressed. When point clouds are captured over extensive areas, drift accumulates proportionally to travel distance, making accurate registration difficult with rigid transformations alone. Furthermore, the proposed method currently lacks a mechanism to correct positional gaps when the scan path forms a loop and the end point fails to coincide with the starting point. The present experiments, conducted across five scans in a single district, represent an initial validation of the concept; scaling to larger extents is a primary direction for future work.

5. Conclusions

This study proposed a pipeline for constructing low-cost wide-area 3D maps using only LiDAR-equipped mobile devices and open data. The novelty of the proposed method lies in the integration of multiple open-source technologies to build a processing pipeline that is robust against the local measurement noise inherent in mobile LiDAR capturing. While cumulative drift over wider areas remains a subject of future work, the method effectively reduces errors in adjacent point cloud registration and subsequent georeferencing through its sequential horizontal and vertical decomposition strategy. Experiments conducted in the Jinaimachi district, Tondabayashi City, Osaka Prefecture, Japan, confirmed a horizontal positional accuracy of 1.75 m RMSE, meeting the 1:2,500-scale mapping requirement, and a vertical positional accuracy of 0.10 m RMSE, surpassing the target set for flood hazard mapping applications. This demonstrates that the proposed method achieves sufficient accuracy for practical 3D disaster hazard map development. These results suggest that even non-experts can construct practical 3D maps using low-cost devices and open data.

Building on the limitations identified in this study, future work will address the following four directions. First, large-scale validation across diverse environments and parameter optimization will be pursued. Generalizability will be assessed using datasets with varying terrain and spatial extent. Additionally, methods for automatic parameter adjustment robust to environmental variation will be investigated. Second, non-rigid transformations will be introduced to address cumulative drift in wider-area capturing and circular routes, enabling more flexible distortion correction. Third, applicability to environments with sparse structures will be extended by incorporating deep learning techniques such as semantic segmentation to exploit road surfaces, sidewalks, and curb regions as additional registration cues. Fourth, a cloud-based web system will be developed to integrate data from multiple devices. This will establish a participatory platform that enables non-experts to easily construct wide-area 3D maps.

Acknowledgements

We express our sincere gratitude to Tondabayashi City, Osaka Prefecture, Japan, for providing access to experimental sites.

This research was supported by the Japan Science and Technology Agency (JST), Co-Creation Field Formation Support Program, grant number JPMJPF2115.

Declaration of generative AI and AI-assisted technologies in the manuscript preparation process

During the preparation of this work the authors used Gemini 3.1 Pro (Google) and Claude Sonnet 4.6 (Anthropic) in order to assist with text refinement and English translation. After using these tools, the authors reviewed and edited the content as needed and take full responsibility for the content of the published article.

References

Bay, H., Tuytelaars, T., Van Gool, L., 2006. SURF: Speeded Up Robust Features. In: Leonardis, A., Bischof, H., Pinz, A. (eds) *Computer Vision – ECCV 2006. Lecture Notes in Computer Science*, vol 3951. Springer, Berlin, Heidelberg. doi.org/10.1007/11744023_32.

Besl, P.J., McKay, N.D., 1992. A method for registration of 3-D shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2), 239-256. doi.org/10.1109/34.121791.

Bradski, G., 2000. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 25(11), 120-125.

Geospatial Information Authority of Japan, 2025a. i-Construction suishin no tame no 3-jigen suchi chikeizu deta sakusei manyuaru [Manual for Creating 3D Digital Topographic Map Data to Promote i-Construction]. Ministry of Land, Infrastructure, Transport and Tourism. www.gsi.go.jp/common/000259826.pdf (17 May 2026) (in Japanese).

Geospatial Information Authority of Japan, 2025b. Sokuryoho dai 34 jo de sadameru sagyo kitei no junsoku [Guidelines for Work Procedures as Specified in Article 34 of the Surveying Act]. Ministry of Land, Infrastructure, Transport and Tourism. www.gsi.go.jp/common/000258734.pdf (17 May 2026) (in Japanese).

Geospatial Information Authority of Japan, 2026a. Kiban chizu joho (kihon komoku) [Fundamental Geospatial Data (Basic Items)], Mesh 513554. Ministry of Land, Infrastructure, Transport and Tourism. service.gsi.go.jp/kiban/app/map/?search=base (17 May 2026) (in Japanese).

Geospatial Information Authority of Japan, 2026b. Kiban chizu joho (suchi hyoko moderu) [Fundamental Geospatial Data (Digital Elevation Model)], Mesh 513554. Ministry of Land, Infrastructure, Transport and Tourism. service.gsi.go.jp/kiban/app/map/?search=dem (17 May 2026) (in Japanese).

Gillies, S., et al., 2013. Rasterio: geospatial raster I/O for Python programmers. Mapbox. github.com/rasterio/rasterio (18 May 2026).

Harris, C.R., Millman, K.J., van der Walt, S.J., et al., 2020. Array programming with NumPy. *Nature*, 585, 357-362. doi.org/10.1038/s41586-020-2649-2.

Krausková, D., Mikita, T., Hrůza, P., Kudrnová, B., 2025. Accuracy Assessment of iPhone LiDAR for Mapping Streambeds and Small Water Structures in Forested Terrain. *Sensors*, 25(19), 6141. doi.org/10.3390/s25196141.

Laefer, D. F., O'Keeffe, E., Chandna, K., Hertz, K., Zhu, J., Lejano, R., Vo, A. V., Bertolotto, M., Ofterdinger, U., 2025. Low-Cost, LiDAR-Based, Dynamic, Flood Risk Communication Viewer. *Remote Sensing*, 17(4), 592. doi.org/10.3390/rs17040592.

Lowe, D.G., 2004. Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision*, 60, 91-110. doi.org/10.1023/B:VISI.0000029664.99615.94.

Luetzenburg, G., Kroon, A., Bjørk, A.A., 2021. Evaluation of the Apple iPhone 12 Pro LiDAR for an Application in Geosciences. *Sci Rep*, 11, 22221. doi.org/10.1038/s41598-021-01763-9.

Ministry of Land, Infrastructure, Transport and Tourism, 2017. Kouzui shinsui soutei kuikizu sakusei manyuaru (dai 4 han) [Manual for the Preparation of Flood Inundation Maps (4th Edition)]. www.mlit.go.jp/river/shishin_guideline/pdf/manual_kouzuishinsui_1710.pdf (17 May 2026) (in Japanese).

Ministry of Land, Infrastructure, Transport and Tourism, 2020. Dosha saigai hazadomappu sakusei gaidorain [Guidelines for Creating Landslide Hazard Maps]. www.mlit.go.jp/river/shishin_guideline/sabo/r0210map_sakusei.pdf (17 May 2026) (in Japanese).

Ministry of Land, Infrastructure, Transport and Tourism, 2026. Project PLATEAU. www.mlit.go.jp/plateau/ (17 May 2026) (in Japanese).

Rublee, E., Rabaud, V., Konolige, K., Bradski, G., 2011. ORB: An efficient alternative to SIFT or SURF. *Proceedings of the 2011 International Conference on Computer Vision (ICCV)*, 2564-2571. doi.org/10.1109/ICCV.2011.6126544.

Rusu, R.B., Blodow, N., Beetz, M., 2009. Fast Point Feature Histograms (FPFH) for 3D Registration. *Proceedings of the 2009 IEEE International Conference on Robotics and Automation (ICRA)*, 1848-1853.

Saptari, A.Y., Widyastuti, R., Suhari, K.T., Suseno, A.D., Hernandi, A., Haris, R.A., Nurmaulia, S., 2024. The Use of iPad LiDAR to Build a Digital Elevation Model in Defining Customary Zones (Case Study of Panglipuran, Bali). *Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci.*, XLVIII-2/W8-2024, 411-418. doi.org/10.5194/isprs-archives-XLVIII-2-W8-2024-411-2024.

U.S. Geological Survey, 2026. 3D Elevation Program. www.usgs.gov/3d-elevation-program (21 May 2026).

Van den Bossche, J., Jordahl, K., Fleischmann, M., Richards, M., McBride, J., Wasserman, J., et al., 2024. geopandas/geopandas: v1.0.1. Zenodo. doi.org/10.5281/zenodo.12625316.

Virtanen, P., Gommers, R., Oliphant, T.E., et al., 2020. SciPy 1.0: fundamental algorithms for scientific computing in Python. *Nature Methods*, 17, 261-272. doi.org/10.1038/s41592-019-0686-2.

Yamazaki, T., Watanabe, K., Kase, T., Hasegawa, K., Saida, K., Miyoshi, T., 2026. A 3D Mobile Crowdsensing Framework for Sustainable Urban Digital Twins. *IEEE Consumer Electronics Magazine*, 15(3), 76-88. doi.org/10.1109/MCE.2025.3598925.

Zhang, W., Qi, J., Wan, P., Wang, H., Xie, D., Wang, X., Yan, G., 2016. An Easy-to-Use Airborne LiDAR Data Filtering Method Based on Cloth Simulation. *Remote Sensing*, 8(6), 501. doi.org/10.3390/rs8060501.

Zhou, Q.-Y., Park, J., Koltun, V., 2018. Open3D: A Modern Library for 3D Data Processing. *arXiv:1801.09847*.

This contribution has been peer-reviewed.
https://doi.org/10.5194/isprs-archives-L-4-W1-2026-319-2026