remote sensing

MDPI

Article

Semantic Segmentation-Driven Integration of Point Clouds from Mobile Scanning Platforms in Urban Environments

Joanna Koszyk [1], Aleksandra Jasińska [2], Karolina Pargieła [2], Anna Malczewska [2], Kornelia Grzelka [2], Agnieszka Bieda [2,*] and Łukasz Ambroziński [1]

[1] Faculty of Mechanical Engineering and Robotics, AGH University of Krakow, al. Mickiewicza, 30, 30-059 Kraków, Poland; jkoszyk@agh.edu.pl (J.K.); ambrozin@agh.edu.pl (Ł.A.)
[2] Faculty of Geo-Data Science, Geodesy and Environmental Engineering, AGH University of Krakow, al. Mickiewicza, 30, 30-059 Kraków, Poland; ajasins@agh.edu.pl (A.J.); pargiela@agh.edu.pl (K.P.); zadlo@agh.edu.pl (A.M.); grzelka@agh.edu.pl (K.G.)
* Correspondence: bieda@agh.edu.pl

Abstract: Precise and complete 3D representations of architectural structures or industrial sites are essential for various applications, including structural monitoring or cadastre. However, acquiring these datasets can be time-consuming, particularly for large objects. Mobile scanning systems offer a solution for such cases. In the case of complex scenes, multiple scanning systems are required to obtain point clouds that can be merged into a comprehensive representation of the object. Merging individual point clouds obtained from different sensors or at different times can be difficult due to discrepancies caused by moving objects or changes in the scene over time, such as seasonal variations in vegetation. In this study, we present the integration of point clouds obtained from two mobile scanning platforms within a built-up area. We utilized a combination of a quadruped robot and an unmanned aerial vehicle (UAV). The PointNet++ network was employed to conduct a semantic segmentation task, enabling the detection of non-ground objects. The experimental tests used the Toronto 3D dataset and DALES for network training. Based on the performance, the model trained on DALES was chosen for further research. The proposed integration algorithm involved semantic segmentation of both point clouds, dividing them into square subregions, and performing subregion selection by checking the emptiness or when both subregions contained points. Parameters such as local density, centroids, coverage, and Euclidean distance were evaluated. Point cloud merging and augmentation enhanced with semantic segmentation and clustering resulted in the exclusion of points associated with these movable objects from the point clouds. The comparative analysis of the method and simple merging was performed based on file size, number of points, mean roughness, and noise estimation. The proposed method provided adequate results with the improvement of point cloud quality indicators.

Keywords: smart city; point clouds; UAV; mobile scanning; quadruped robot; semantic segmentation

check for updates

Citation: Koszyk, J.; Jasińska, A.; Pargieła, K.; Malczewska, A.; Grzelka, K.; Bieda, A.; Ambroziński, Ł. Semantic Segmentation-Driven Integration of Point Clouds from Mobile Scanning Platforms in Urban Environments. *Remote Sens.* **2024**, *16*, 3434. https://doi.org/10.3390/rs16183434

Academic Editor: Pablo Rodríguez-Gonzálvez

Received: 27 July 2024
Revised: 5 September 2024
Accepted: 11 September 2024
Published: 16 September 2024

Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

To enhance the quality of life for urban residents, transforming cities into smart cities is essential. This concept emphasizes city management through the use of technical tools, incorporating the latest technologies [1]. One of the significant aspects of smart cities development is an accurate and high-quality reconstruction of 3D spatial information of urban environments [2–7]. Technologies such as UAVs (unmanned aerial vehicles), aerial photogrammetry, oblique photogrammetry, LiDAR (Light Detection and Ranging), Mobile Mapping system, and SLAM (Simultaneous Localization and Mapping) are widely used but each has its limitations [8–12]. When nadir data are acquired by UAV, the recordings lack building elevations, while mobile systems miss the roofs [13]. To fill such gaps, data integration is necessary [14]. Another aspect is the geolocation and positioning of the

collected data. This can be done in real-time using RTK (Real-Time Kinematics), or by aligning it to GCPs (Ground Control Points) with known coordinates. Therefore, data integration from different sources is an essential part of preparing accurate and complete point cloud representations [15–19].

A previous publication [20] provides an overview of methods used for collecting 3D spatial data on urban areas and terrain topography that can be used in building a smart city. In addition, the authors emphasized that full 3D modeling of a city should consist of a combination of ground and aerial measurements. On the other hand, in the article [21] the authors presented an example of using data from different sensors. Improved positioning of UAV photogrammetry without RTK can be achieved by using point clouds obtained from mobile mapping. The authors investigated the usefulness of point clouds from mobile scanning in improving geolocation and enhancing the obtained point cloud by combining clouds from both sources. UAV photogrammetry without RTK often suffers from inaccuracies due to a lack of precise geolocation data. However, by integrating these data with point clouds generated from mobile mapping systems, positioning accuracy can be significantly improved. This combination benefits from the high precision of mobile mapping point clouds to enhance and correct UAV photogrammetric data, resulting in more accurate and comprehensive mapping. The integration of these two data sources thus resolves the shortcomings of each method individually and creates a more reliable point cloud dataset. In the context of point clouds, each point represents a part of an object. Accurate recognition of the object can enable the distinction between moving and stationary elements. However, removing moving objects from a point cloud can leave significant gaps. Moreover, obtaining a complete scene may require an additional laser scanning source due to the limitations of a single solution. A detailed review of object detection, segmentation, and cloud classification methods obtained using mobile techniques is presented in the literature [22]. The authors provide a comprehensive analysis segmented by the different types of scenes studied (forest, railroad, tunnel, and urban/street), as well as a review of existing and potential benchmarks that can be used.

In [23], the authors integrated indoor registration data collected with a low-cost sensor (LiDAR from Apple) with a photogrammetric point cloud acquired from a UAV to model buildings according to the LOD3 standard. In the article [24], the authors analyze various algorithms for 3D reconstruction of urban objects based on point clouds. In [3], the authors proposed a method to generate a 3D building model based on matching point clouds and meshes. The most common method of integration is to attach the points of one cloud to another, but this can lead to data duplication and increased file sizes, especially for external scans, where it is crucial to complete only certain parts of the point clouds [3]. A solution may be to use algorithms that allow the detection of specific elements, such as semantic segmentation, which assigns a specific class to each segment in the scene representation [25–28]. The application of deep convolutional networks to 2D objects with structured features gives excellent results but cannot be directly applied to 3D point clouds [29]. PointNet++ is a neural network adapted for processing complex scenes provided as an unstructured set of points [30]. The architecture is based on direct processing of 3D data using PointNet [31], with an extension of the hierarchical structure. The proposed deep neural network consists of a sampling layer, a clustering layer, and a PointNet layer, which allows for building local regions, leading to an increase in the performance of the neural network in semantic segmentation. Striving for a complete and accurate representation of structures is crucial across various fields such as architecture and construction [32], including of the increasingly popular smart cities [33,34]. In this article, we propose a new method for mutual complementation of point clouds obtained from ground-based (the quadruped robot Spot) and aerial (UAV) mobile methods. An additional challenge potentially affecting the integration of both clouds was acquiring them during different vegetative seasons. Significant for building structural representations was also the elimination of moving elements that were abundant in the studied scene (such as cars, trees, and pedestrians). Our method relies on a semantic segmentation algorithm

and point cloud subdivision into subregions. Points from both clouds are then selected to mutually complement any missing elements. Additionally, we implemented clustering to reduce unnecessary elements. The results demonstrate that our proposed method achieves its intended goals, providing a full representation of objects while excluding outliers. Consequently, the resulting point cloud can serve as a reference for creating and managing smart cities.

This paper primarily contributes the following:

- Development of a new algorithm for fusion and complementation of point clouds obtained from two different mobile methods: ground-based (the quadruped robot Spot) and aerial (UAV).
- The proposed method involves eliminating moving elements that appear on both point clouds, tested in the case of a large time difference between data acquisition.
- The proposed method allows for the densification of clouds not only in the gaps left after removing moving elements, but also in the case of rare registration of one of the scans.

2. Materials and Methods

2.1. Test Field

The research test field was part of AGH University of Krakow campus located in southern Poland. The measured area (20,391 m^2) is marked in Figure 1. This location was selected for its distinctive characteristics. The study area features various multi-story educational buildings, mainly from the latter half of the 20th century, along with a modern building from 2022 with large, glazed areas. The area between the buildings includes roads, sidewalks, car parks, and green spaces with many mature trees. It also serves as a primary pathway for student movement through the campus. The integration of mobile data from different sources in this environment is challenging due to the ongoing movement of cars and pedestrians during measurements.

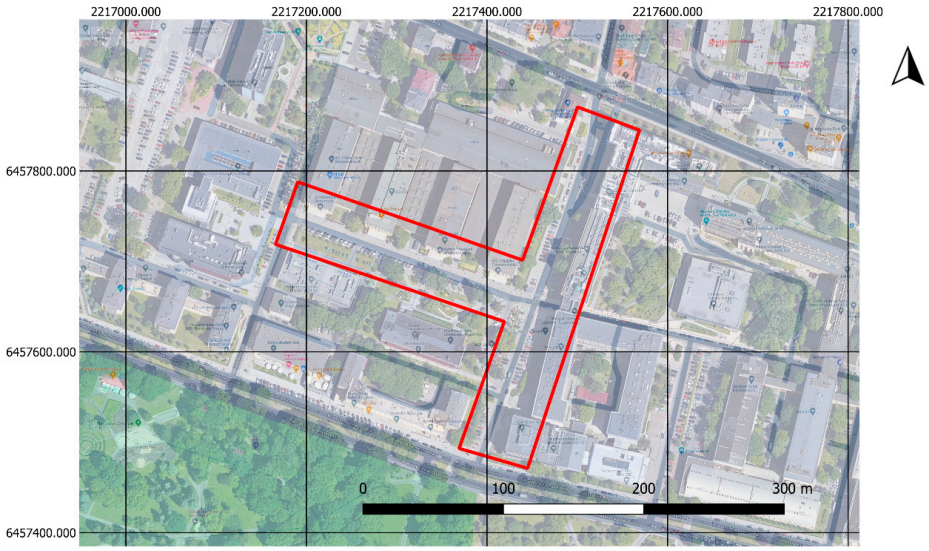


Figure 1. Area of investigation (red box). Coordinates refer to WGS84 (EPSG: 4326). Background image: Google Earth, earth.google.com/web/.

2.2. Equipment

The initial phase involved utilizing the Boston Dynamics Spot robot, which was equipped with the Leica BLK ARC scanner, as depicted in Figure 2. This quadruped robot's design enables it to navigate stairs and traverse rough terrain, making it adept at operating in difficult-to-access areas. Equipped with a laser scanner mounted at the front, both systems were integrated using Spot CORE I/O to perform a comprehensive scan of the environment. Table 1 provides detailed specifications of the scanner. The manufacturer

indicates that noise levels and accuracy may fluctuate based on the working environment. The obtained point clouds had a local coordinate system.



(**a**) (**b**)

Figure 2. Leica BLK ARC laser scanner (**a**), Boston Dynamics Spot equipped with Leica BLK ARC (**b**).

Table 1. Parameters of Leica BLK ARC laser scanner (Source: [35]).

Parameter Name	Parameter Value
Weight	690 g
Height	183.6 mm
Diameter	80 mm
Wavelength	830 nm
Field of view	360° (horizontal)/270° (vertical)
Range	Min. 0.5-up to 25 m
Point measurement rate	420,000 pts/s
Range noise	+/−3 mm (dynamic) +/−2 mm (static)
Accuracy indoors	+/−10 mm

In the next step, data were collected using the DJI Matrice 350 RTK drone equipped with the DJI Zenmuse L1 lidar (Figure 3). Due to the RTK module mounted on the drone, the acquired point clouds were obtained in the ETRF2000-PL / CS2000/21 (EPSG: 2178) coordinate system. The flights were carried out in a single line, flying along the central strip of the analyzed area at 50 m altitude. The mission was autonomously planned to use the DJI Pilot 2 application. In the next step, the point clouds were aligned in the DJI Terra software. The parameters of the scanner used are presented in Table 2.



Figure 3. DJI Matrice 350 RTK equipped with DJI Zenmuse L1.

Table 2. Parameters of DJI Zenmuse L1 laser scanner (Source: [36]).

Parameter Name	Parameter Value
Weight	ca. 900 g
Height	169 mm
Field of view	70.4° × 4.5°
Range	up to 450 m
Point measurement rate	240,000 pts/s single reflection 480,000 pts/s external reflection
Accuracy	+/−10 cm (horizontal) +/−5 cm (vertical)
Distance accuracy	+/−3 cm/100 m

2.3. Workflow

2.3.1. Semantic Segmentation

Due to the point clouds obtained from UAV and SPOT having different coordinate systems, the work began with an initial alignment of the clouds to a common reference system. The ETRF2000-PL/CS2000/21 (EPSG: 2178) from UAV RTK was taken as the reference. The alignment was carried out using the ICP (Iterative Closest Point) algorithm first introduced by Besl and McKay in 1992 [37]. Due to the differences between the two clouds, the algorithm was unable to accurately align them. Therefore, the resulting alignment was further manually refined. The goal was not to achieve perfect alignment, as the main objective was to test whether the algorithm we implemented could produce correct results even with only preliminarily aligned clouds.

As the point clouds from aerial and mobile acquisitions are significantly different, two deep neural network models were trained. We applied a semantic segmentation example from the MATLAB documentation to training datasets and our data [38]: ALS: Dayton Annotated LiDAR Earth Scan (DALES) [39]; and MLS: Toronto 3D [40]. The DALES data were labeled with 8 classes including ground, buildings, cars, and vegetation. In the original paper, the authors conducted a comprehensive study regarding the performance of state-of-the-art deep neural networks suitable for semantic segmentation. The study of the DALES dataset provided an overall accuracy of 95.7% for PointNet++. The Toronto 3D dataset was obtained in an urban environment. The labels slightly differed from those of the DALES dataset, but, among others, distinguished roads, buildings, vegetation, and cars. Similarly, as in the first dataset, the authors of the paper explored the performance of a few methods trained on their data. The overall accuracy for PointNet++ reached 91.21%.

The expected result of our study was to achieve better results for the UAV scan from the model trained on DALES and for the mobile platform from the model trained on Toronto 3D since it is MLS data. The semantic segmentation of our data from the models trained on the two datasets is shown in Figure 4.

Due to the unfavorable results for the model built using the Toronto 3D dataset, the semantic segmentation based on the model trained on DALES was selected for further analysis.

The street scene point clouds obtained with aerial and on-ground laser scanning are different from the data from the training dataset, which affected the classification. The most accurate results are depicted in Figure 5.

Although the model achieved a high overall performance of 93.6% on the testing dataset, the accuracy of each class varied from 50 to 99%. The ground label reached the highest accuracy, and because of that, in this paper we consider labels “ground” and “other”. The results from semantic segmentation were binarized. The separation of the ground from the rest of the elements in the scene is shown in Figure 6.

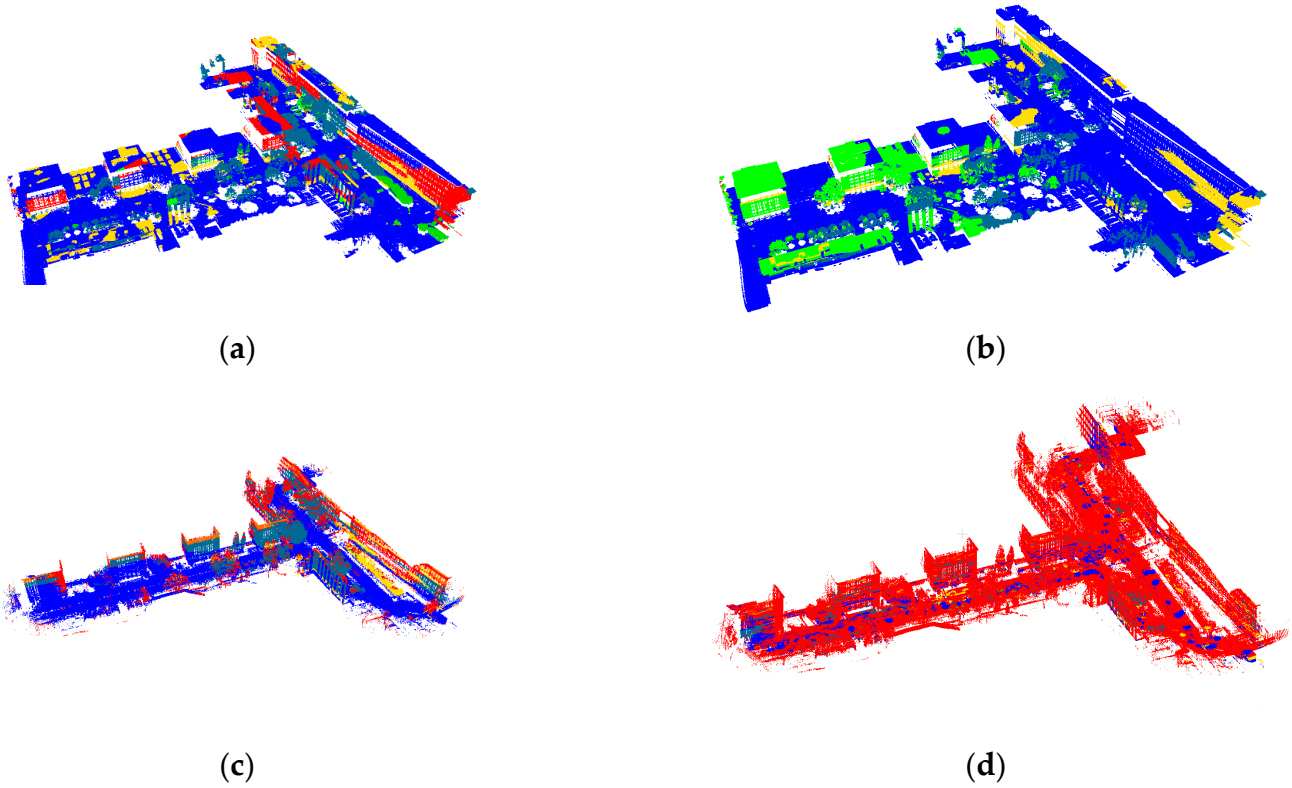


Figure 4. Comparison of PointNet++ performance. UAV data are classified based on models trained on (**a**) DALES and (**b**) Toronto 3D. Mobile robot data classified based on models trained on (**c**) DALES and (**d**) Toronto 3D. Different colors represent labels assigned to points.

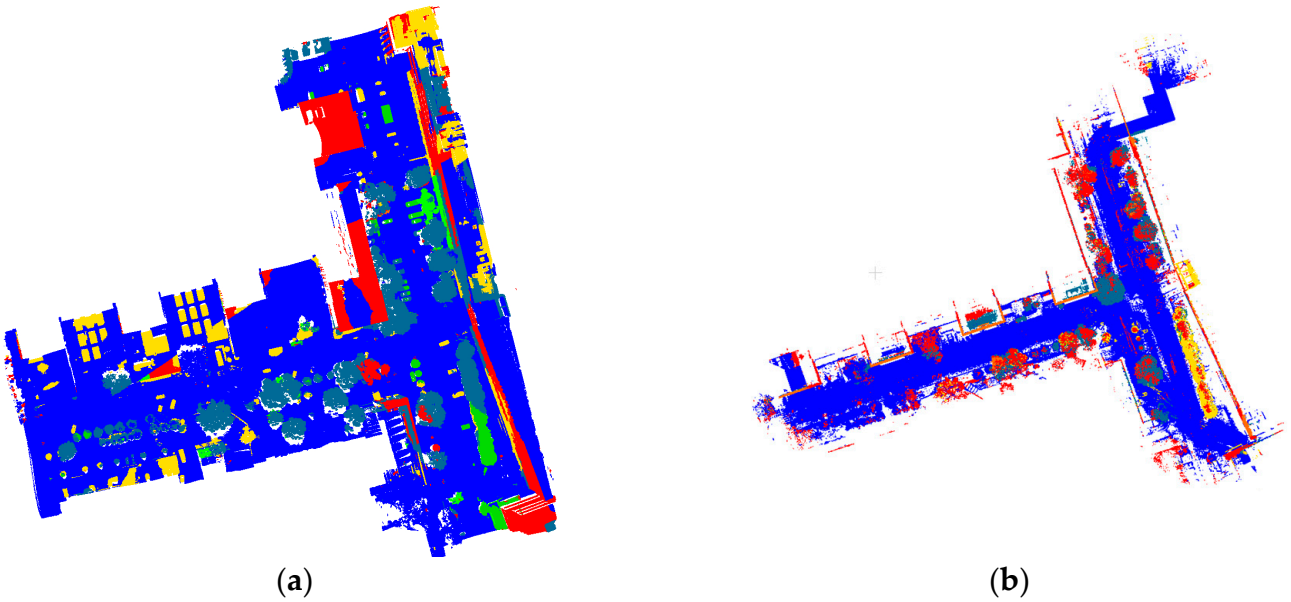
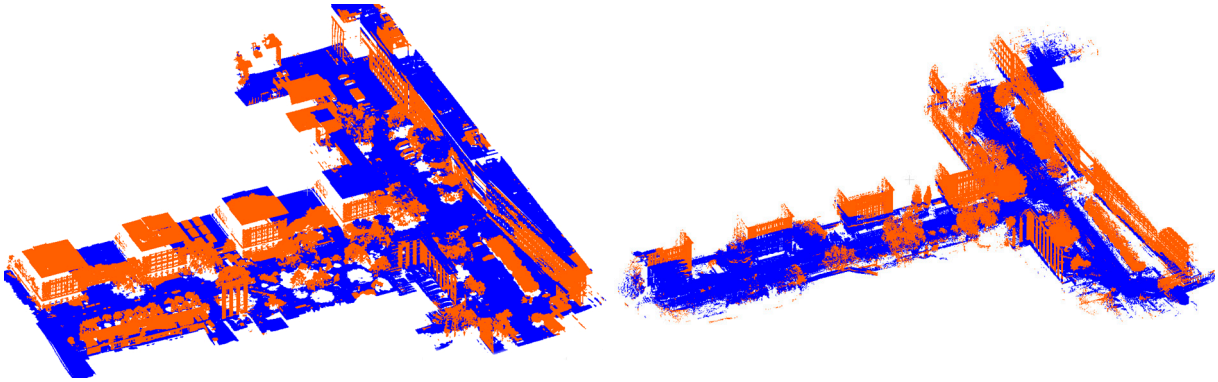


Figure 5. Semantic segmentation: (**a**) UAV point cloud, (**b**) mobile platform point cloud. Different colors represent labels assigned to points.



(**a**) (**b**)

Figure 6. Ground classification after binarization: (**a**) UAV point cloud, (**b**) mobile platform point cloud. Blue color represents the ground label. and orange color represents the non-ground label.

2.3.2. Integration

To integrate two point clouds from different sources, we propose a method where each point cloud is divided into small subregions that are compared to each other to select a preferable subregion. The final point cloud is composed of selected subregions. This section presents our approach. The method is structured into three main stages:

1. Data preparation: Semantic segmentation and outlier removal.
2. Subregion separation: Division of point clouds to corresponding subregions.
3. Subregion selection: Subregions are selected based on label dominance in the subregion, local density, and centroid localization.

At first, data are prepared by performing semantic segmentation on both point clouds. This operation is followed by binarization of labels, which provides a classification of each point in the dataset with the distinction between ground and non-ground elements. In the next step of data preparation, the point clouds are filtered by removing the outliers based on the average distance of points to their neighbors with the number of neighbors set to 20.

In the consecutive stage, subregion separation is performed. First, the algorithm parameters necessary to define subregions are calculated based on the dimensions of input point clouds. The first subregion coordinates are chosen as minimal coordinates of the point clouds. The number of subregions was set to approximately provide a rectangle with 0.5 m sides on both X and Y axes. Accordingly, the number of subregions along X was set to 552 and along Y was set to 470. The last subregion was constrained with the maximal coordinates of point clouds.

The next stage focuses on subregion selection. For each considered subregion area, UAV and mobile robot subregions are delineated with corresponding labels from semantic segmentation. The first part of the subregion selection phase is to check whether subregions contain any points or are empty. If one of the subregions does not have any points, the final point cloud is completed with points from the other source. In case none of the subregions have points, the next subregion area is considered.

In case both subregions have points, local densities, centroids, and the Euclidean distance between subregions are calculated for proper subregion selection. Next, the algorithm considers scenarios when the subregions overlap and the Euclidean distance between them is lower than the DJI Zenmuse L1 laser scanner vertical measurement accuracy, which is equal to 0.05 m; the subregion from the UAV scan is selected due to the disadvantages of the mobile robot-based solution, such as the lack of the buildings' ceilings. The advantage of UAV subregions results from the fact that the laser scanner suitable for a quadruped robot proliferates points corresponding to the scanning time, and the resulting point cloud has artefacts that arise from the operator walking next to the robot. In different cases, when the subregions do not overlap and are further than 5.0 m, the subregions are considered as different elements in the point cloud, and both are added to the final point cloud.

In other scenarios that have not been already covered by the part of the algorithm described above, ground dominance is calculated in each subregion based on the number of ground labels assigned earlier with semantic segmentation. If at least 50% of points are labeled as ground, the ground dominance is set to True. If only one subregion has ground dominance, only that region will be a part of the final point cloud. In ambiguous cases where both or neither of subregions have ground dominance, the local density is compared and the subregion with a much higher local density is selected because the other subregion is regarded as being not thoroughly scanned in that particular area.

For all the remaining cases, where the local density is similar, the algorithm takes into account the Z coordinates of subregion centroids. The area with a lower Z value is selected because a higher Z value might indicate that some object was positioned in the subregion during the scan. This approach of selecting a lower Z centroid value was implemented to reduce points from cars, people, or other mobile objects.

The procedure is described in Algorithm 1. The source code was written in Python.

Algorithm 1: UAV and mobile scan integration

```
Input: uav_point_cloud, spot_point_cloud, n, m
Output: integrated_point_cloud
uav_point_cloud, uav_labels = semantic_segmentation(uav_point_cloud)
spot_point_cloud, spot_labels = semantic_segmentation(spot_point_cloud)
uav_point_cloud = remove_statistical_outlier(uav_point_cloud)
spot_point_cloud = remove_statistical_outlier(spot_point_cloud)
pcd_start_x, pcd_stop_x, step_x, pcd_start_y, pcd_stop_y, step_y =
calculate_algorithm_limits(uav_point_cloud, spot_point_cloud)
for i in n:
    for j in m:
        Select spot_subregion, uav_subregion, uav_label_subregion, spot_label_subregion
            if (spot_subregion.size == 0) and (uav_subregion.size == 0):
                continue
            elif spot_subregion.size == 0:
                complete_pcd = concatenate((complete_pcd, uav_subregion))
                continue
            elif uav_subregion.size == 0:
                complete_pcd = concatenate((complete_pcd, spot_subregion))
                continue
            spot_local_density, uav_local_density =
calculate_local_densities(spot_subregion, uav_subregion)
            euclidean_dist = calculate_euclidean_distance(uav_subregion, spot_subregion)
            if euclidean_dist < 0.05:
                complete_pcd = concatenate((complete_pcd, uav_subregion))
            else:
                spot_ground_dominance, uav_ground_dominance =
calculate_label_dominance(spot_subregion_labels, uav_subregion_labels)
            if euclidean_dist > 5.0:
                complete_pcd = concatenate((complete_pcd, uav_subregion))
                complete_pcd = concatenate((complete_pcd, spot_subregion))
            elif uav_ground_dominance and not spot_ground_dominance:
                complete_pcd = concatenate((complete_pcd, uav_subregion))
            elif not uav_ground_dominance and spot_ground_dominance:
                complete_pcd = concatenate((complete_pcd, spot_subregion))
            else:
                if abs(uav_local_density-spot_local_density) > 15:
                    if uav_local_density > spot_local_density:
                        complete_pcd = concatenate((complete_pcd, uav_subregion))
                    else:
                        complete_pcd = concatenate((complete_pcd, spot_subregion))
                else:
                    if subregion_spot_centroid.Z- subregion_uav_centroid.Z > 0:
                        complete_pcd = concatenate((complete_pcd, uav_subregion))
                    else:
                        complete_pcd = concatenate((complete_pcd, spot_subregion))
```

The entire workflow of the conducted research is presented in the diagram in Figure 7.

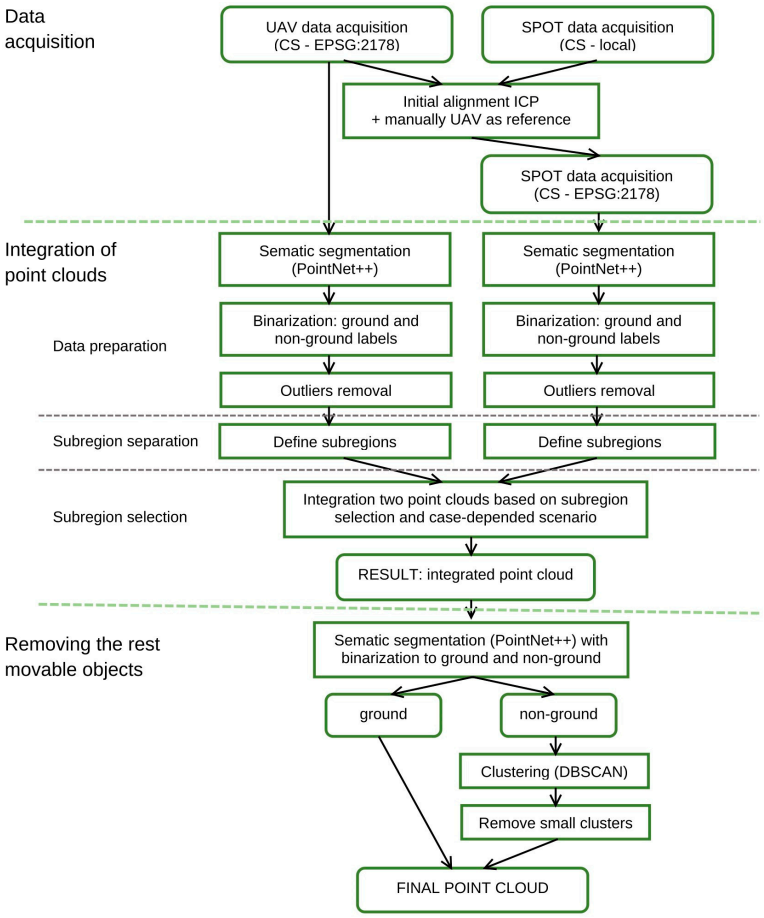


Figure 7. The diagram of research workflow.

3. Results

The algorithm's outcome provided the point cloud that derived assets from the UAV and the quadruped robot scanning. The result of the algorithm is shown in Figure 8.

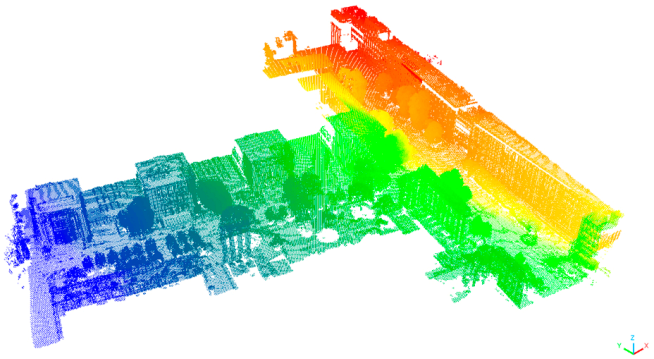


Figure 8. Integrated point cloud.

3.1. Visual Comparison

The integration proposed in this study provided test field representation containing lacking information from both point clouds. The comparison between scans collected with the UAV, the mobile robot, and the point cloud created with our integration algorithm is shown in Figure 9. Ceilings are presented in (a), (b), and (c). The UAV scan (Figure 9a) contains the upper side of the building, while the scan from the ground (Figure 9b) lacks this information. Moreover, the sides of the building were not obtained with the UAV, but the quadruped robot recorded them. Both these features were included in the integrated

point cloud (Figure 9c). The front of the buildings is presented in (d), (e), and (f). Aerial acquisition (Figure 9d) did not collect the upper and lower sides of the front due to occlusion from the ceiling. While only buildings' front centers are visible on UAV scan, the data from the robot (Figure 9e) not only contain the complete front walls but also a part of the interior of one of the buildings. These assets from the mobile platform point cloud were included in the integrated point cloud (Figure 9f).

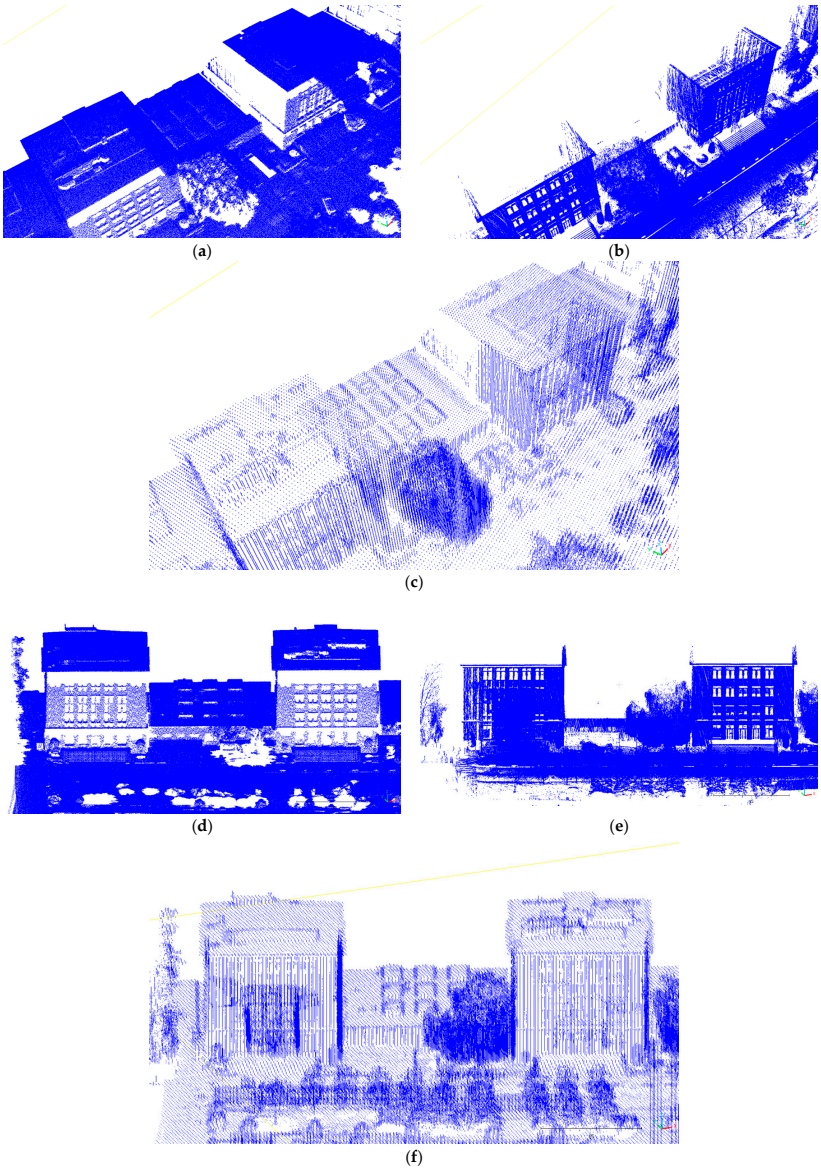


Figure 9. Comparison between scans obtained from different devices and the point cloud created with the proposed algorithm. Ceilings: (**a**) UAV, (**b**) quadruped robot, and (**c**) integrated point cloud. Building fronts: (**d**) UAV, (**e**) quadruped robot, and (**f**) integrated point cloud.

The algorithm not only managed to complement the lacking parts but also was able to reduce movable objects. The cars' positions in the parking lots differed between aerial (Figure 10a) and on-ground scans (Figure 10b). The algorithm reduced some of the vehicles (Figure 10c), but some parts remained due to cars coinciding in the same area on both scans.

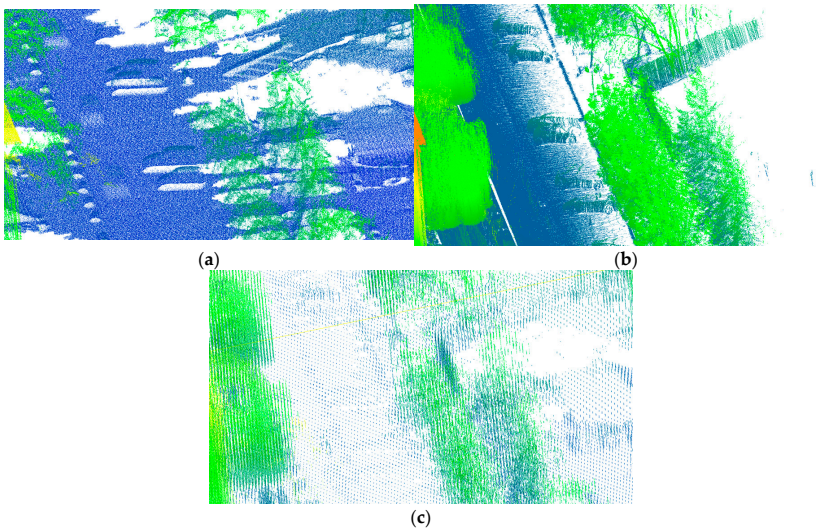


Figure 10. Comparison between scans obtained from different devices and the point cloud created with the proposed algorithm. Cars: (**a**) UAV, (**b**) quadruped robot, and (**c**) integrated point cloud.

Although the algorithm performs integration, it does not manage to reduce all the movable parts from the scene. In Figure 11a–c the retained cars are visible. The integrated point cloud (Figure 11c) contains the same elements as the UAV point cloud (Figure 11a) because the mobile robot did not collect data for that area (Figure 11b) and the point cloud was completed with UAV data. Additionally, the trees were not reduced from the scene in the algorithm outcome (Figure 11f) because they were present on both scans (Figure 11d,e).

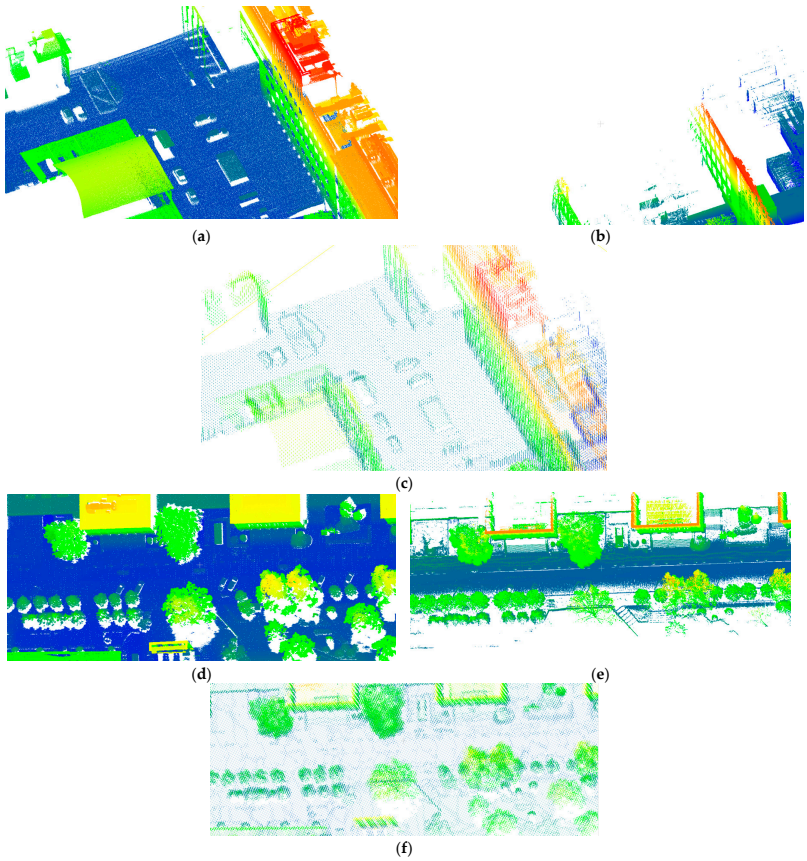
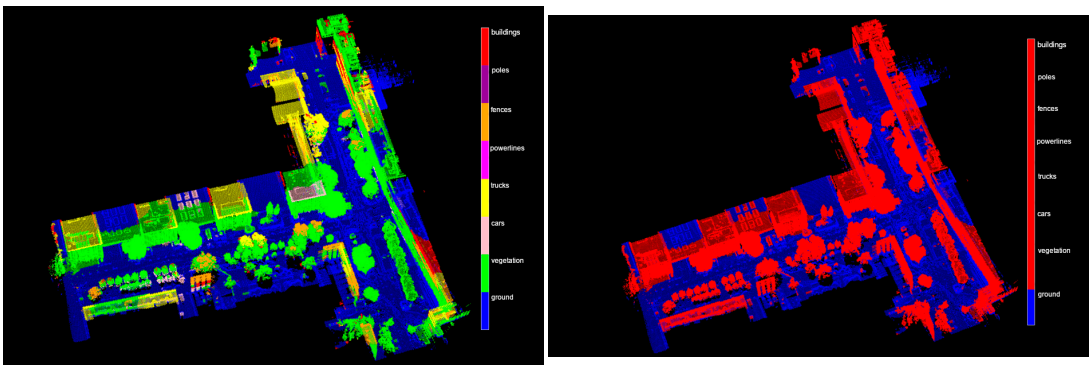


Figure 11. Comparison between scans obtained from different devices and the point cloud created with the proposed algorithm. Cars: (**a**) UAV, (**b**) quadruped robot, and (**c**) integrated point cloud. Trees: (**d**) UAV, (**e**) quadruped robot, and (**f**) integrated point cloud.

3.2. Moving Object Removal

To reduce movable parts, further steps were required. First, semantic segmentation was performed on the integrated point cloud with prior down-sampling. As noted previously, the semantic segmentation result was binarized to distinguish the ground label from other labels. Semantic segmentation before and after binarization is shown in Figure 12.



(**a**) (**b**)

Figure 12. Semantic segmentation of integrated point cloud (**a**) with 8 classes and (**b**) binarized.

In the next step, the points with the ground label were removed. This operation allowed isolation of the points with an ambiguous label that could have been incorrectly classified. Points with the non-ground label are shown in Figure 13.

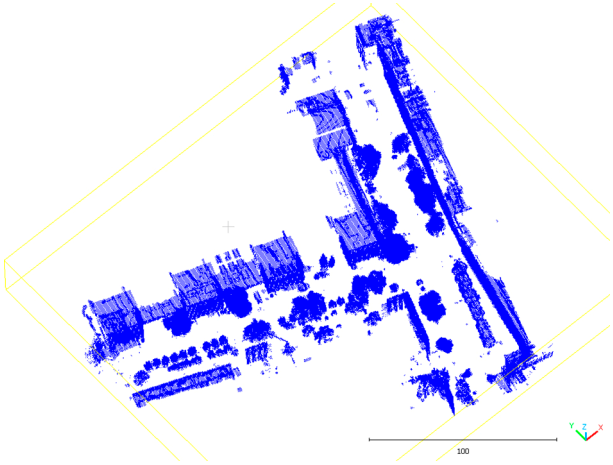


Figure 13. Point cloud without points with the ground label.

Since some of the movable elements were not reduced with the proposed algorithm, we performed clustering on non-ground objects to identify small objects. For that purpose, a clustering algorithm, DBSCAN [41], was employed with a maximum distance between neighbors set to 3.0 and a minimum number of samples equal to 300. The point cloud was down-sampled to decrease calculation time. The algorithm should group neighboring points into one cluster. The clusters representing larger objects such as buildings consisting of more points than small elements like trees or the remaining parts of cars. The DBSCAN approach enabled outlier removal. The clustering result is shown in Figure 14.

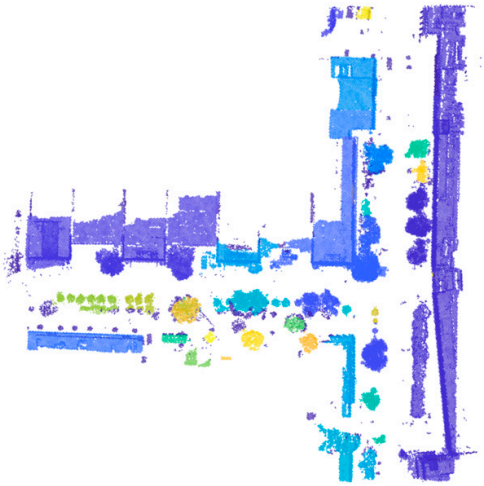
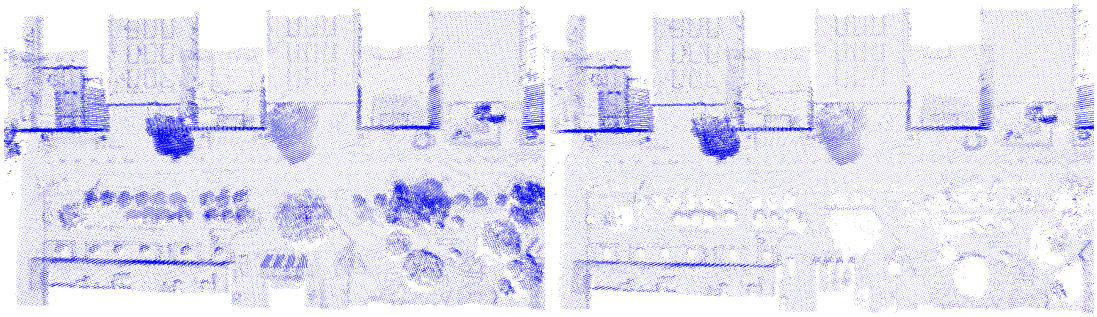


Figure 14. Point cloud with ground removed after clustering with DBSCAN. Each cluster is indicated with a different color. Small elements such as small trees are grouped into separated clusters.

As expected, elements such as trees and cars were grouped into small clusters, which allowed them to be separated from buildings. Setting the threshold to 20,000 allowed stationary objects like buildings to be preserved and small elements, e.g., trees, cars, and other artefacts, to be removed. The last stage included adding the ground to the remaining points. The main result obtained from the performed processing was a point cloud integrated with UAV and SPOT. In the final point cloud, areas of solid objects where the lack of points was identified were densified, while objects that were considered moving were removed. The final result is illustrated in Figure 15. The scene fragment focuses on the area where small trees were removed. Contrary to the point cloud acquired by performing visible integration in Figure 8, the scene after clustering and outlier removal does not contain movable objects that might change in the future.



(**a**) (**b**)

Figure 15. Final point cloud (**a**) before outlier removal and (**b**) after outlier removal.

3.3. Results Validation

To evaluate the obtained results, a comparison between point clouds integrated with our method and point clouds obtained with state-of-the-art methods was performed. Both the outcome of the integration proposed in Section 2.3.2 and the point cloud after DBSCAN-based point reduction for outlier removal were considered in the comparison. Our method was balanced against a simple merging of UAV and mobile robot data, and the overview of findings is presented in Table 3. The file size of the dataset in ply format, number of points, mean roughness, and noise estimation are included in the comparison. The results showed that simple merging results in a point cloud that is increased both in size and the number of points. Our method reduces the points, making the point cloud more memory-efficient for possible further processing purposes. While the simple merging result might contain

duplicated points arising from appending UAV point cloud to mobile robot data without filtering, our method selects which subregions to append. Our method resulted in point clouds before and after clustering that were reduced in size by 28% and 69% relative to the simple merging method. Additional indicators other than the point cloud size considered in the comparative analysis are mean roughness and noise estimation. Due to the large volume of point clouds, the analysis was performed on a representative fragment of the street, which was the same for each method. The evaluation of roughness was run with a local neighbor radius set to 1 m to capture sharp changes in the surface over a larger area that can include movable objects. This spread of neighbors can indicate the presence of changeable objects. While roads, pavements, and buildings' walls are smooth surfaces, objects such as cars, pedestrians, and trees have geometries that can be distinguished from the surroundings due to the presence of irregularities. To assess the entire point cloud, mean roughness is calculated. The results show that the mean roughness of point clouds obtained with our method is lower than that created with the simple merging method. Moreover, the reduction in additional movable parts with clustering decreases the roughness score. Another indicator considered in the analysis was noise estimation calculated as the third eigenvalue. The worst score was achieved with simple merging and the best with integration proposed in this study with clustering.

Table 3. Comparison of our method to simple merging.

Method	File Size	Number of Points	Mean Roughness	Noise Estimation
Simple merging	1.00 GB	89,514,904	0.085992	0.027336
Our method before clustering	724 MB	63,305,027	0.084617	0.025439
Our method with clustering	311 MB	4,807,391	0.082617	0.023868

Additionally, the abovementioned artefacts pointed out through visual analysis of point clouds before integration are present after applying simple merging.

To further explore the merit of our approach, we compared it to two state-of-the-art methods. The geometric-aware method proposed by Li et al. [42], despite its ability to manage accurate point cloud integration from multiple sources, is complex in comparison to our method and is unable to correct defects that arise in the algorithm's input. The algorithm cannot handle the removal of dynamic objects like parked cars, while our method is capable of erasing movable elements, leaving only the permanent infrastructure. The second considered approach from the literature was the method proposed by Aijazi et al. [43]. It focused on the imperfections that appeared in scans. Although the algorithm can reduce the number of artefacts and occlusion based on scans gathered at different times, it depends on data quality. Additionally, different scanning conditions were not explored and it might be challenging to apply this method to integrate UAV data and a mobile scanner. The authors did not refer to varying perspectives and area coverage resulting from different scanning platforms.

4. Discussion

This study introduces a point cloud integration method from different platforms. The results and comparative analysis show the superiority of the proposed algorithm over simple merging. The comparison of point cloud sizes proved the ability of the proposed method to filter the input. Additionally, the mean roughness and noise estimation analysis proved the prevalence of our method over the well-known simple merging approach. Moreover, the presented indicators showed that supplementary small object removal improves the point cloud by making it smoother and less noisy. Furthermore, the comparison with two state-of-the-art methods highlighted the advantages of our method.

However, the algorithm has its limitations. First of all, semantic segmentation performance might be flawed in certain setups. Since PointNet++ was trained on the dataset where the similarities in training data result from the same method of acquisition with the

same scanning device, it might be challenging to provide accurate results for data gathered on another platform or with the use of significantly different laser scanners. Future research could focus on improving the model and making it suitable for different data. To improve the PointNet++ performance and make it more robust to various outdoor point clouds, data augmentation techniques could be implemented on the training dataset. Methods such as affine transformation, drop, jittering, and GT sampling could leverage the ability to adapt to data from another source [44]. Another solution would be gathering own data with the Leica BLK ARC laser scanner and labeling it. The network trained only on Leica BLK ARC point clouds could provide accurate results for the point cloud considered in this paper. However, both obtaining a dataset that is large enough to perform the training and labeling it would be time-consuming. Additionally, gathering diverse outdoor data on the mobile robot to avoid overfitting could be difficult due to car traffic. Another refinement would be combining a large dataset from the existing open-source datasets. This approach could provide a model that is suitable for both aerial and ground data, although the downside of this improvement would be difficulties with the unification of labels for all included datasets.

Other drawbacks of the method are the gaps that appear after clustering. The movable parts that are reduced from the scene are not replaced automatically, which causes environment representation to be incomplete. Since both UAV and mobile robot data do not contain information about areas underneath some of the removed elements, gathering additional data would be necessary to fill the gaps.

Future research could examine the proposed method's performance in areas under construction. In this paper, the test field scans were acquired in the span of a few months and the area was not being rebuilt at the time. This resulted in no significant changes in the scene. Although the aim of the study was achieved and the point clouds were integrated into a complete representation of the test field, the algorithm's robustness to more prominent modifications was not investigated. Additionally, the method could be modified to handle special scenarios such as covering a crater. To leverage the performance of the algorithm, further steps could be considered. Considering historical data would provide sufficient information to ensure that all the alterations are applied to the resulting point cloud. This would require introducing the certainty factor for each point where the value would be assigned based on acquisition time. However, for this approach more than two historical scans would be necessary.

5. Conclusions

Dividing the point cloud into subregions allowed for the penetration of the whole collected area. The algorithm reaps the benefits from aerial scanning and on-ground acquisition by completing the point cloud that lacks elements. Additionally, the algorithm can remove movable objects like cars if they are not present in the other scan. The point cloud resulting from the solution proposed in the paper is further subjected to clustering to reduce other nonmatching elements that can be considered outliers after removing ground-labeled points. The proposed semantic segmentation-driven integration of point clouds has proven its effectiveness in creating a complete environment representation. By utilizing prepared point clouds, sophisticated three-dimensional models can be created to enhance smart city development and management.

The obtained point cloud can allow the production of 3D models that will enable various types of spatial analysis, leading to better management of urban development. The measurements made, together with an algorithm for classifying individual objects on the scan, can be used for more effective monitoring of environmental hazards directly related to the concept of building a smart city. Both the combination of different data acquisition technologies and the created algorithm will allow work that will enable improved services, and energy, cost, and time savings.

Author Contributions: Conceptualization, J.K., K.G., A.J., A.M., K.P. and Ł.A.; methodology, J.K. and Ł.A.; software, J.K. and K.P.; validation, J.K., K.G., A.J., A.M., K.P. and Ł.A.; formal analysis, J.K., K.G., A.J., A.M., K.P., A.B. and Ł.A.; investigation, J.K. and K.P.; resources, J.K., K.G., A.J., A.M., K.P. and Ł.A.; data curation, J.K. and A.J.; writing—original draft preparation, J.K., K.G., A.J., A.M., K.P., and Ł.A.; writing—review and editing, J.K., A.J. and K.P.; visualization, J.K., A.J. and K.P.; supervision, Ł.A.; funding acquisition, A.B. All authors have read and agreed to the published version of the manuscript.

Funding: Authors JK and ŁA would like to acknowledge the research subvention of AGH University of Krakow No. 16.16.130.942. Authors K.P., A.J., A.M., K.G. and A.B. would like to acknowledge the research subvention of AGH University of Krakow No. 16.16.150.545 and note that research project partly supported by program „Excellence initiative–research university" for the AGH University of Krakow (action 4, application number: 6325).

Data Availability Statement: Data are available under the link: http://www.dx.doi.org/10.6084/m9.figshare.26362369.

Acknowledgments: We would like to acknowledge "mierzymy.pl Marek Pudło" company for providing the Leica laser scanner for tests.

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

References

1. Nevistić, Z.; Bacic, Z. The Concept, Realizations and Role of Geosciences in the Development of Smart Cities. *Teh. Vjesn.—Tech. Gaz.* **2022**, *29*, 330–336. [CrossRef]
2. Wang, X.; Jiang, L.; Wang, F.; You, H.; Xiang, Y. Disparity Refinement for Stereo Matching of High-Resolution Remote Sensing Images Based on GIS Data. *Remote Sens.* **2024**, *16*, 487. [CrossRef]
3. Liu, W.; Zang, Y.; Xiong, Z.; Bian, X.; Wen, C.; Lu, X.; Wang, C.; Marcato, J.; Gonçalves, W.N.; Li, J. 3D Building Model Generation from MLS Point Cloud and 3D Mesh Using Multi-Source Data Fusion. *Int. J. Appl. Earth Obs. Geoinf.* **2023**, *116*, 103171. [CrossRef]
4. Ismail, M.H.; Shaker, A.; Li, S. Developing complete urban digital twins in busy environments: A framework for facilitating 3D model generation from multi-source point cloud data. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2023**, *48*, 7–14. [CrossRef]
5. Hu, Y.; Liu, Z.; Fu, T.; Pun, M.-O. Dense 3D Model Reconstruction for Digital City Using Computationally Efficient Multi-View Stereo Networks. In Proceedings of the IGARSS 2022—2022 IEEE International Geoscience and Remote Sensing Symposium, Kuala Lumpur, Malaysia, 17–22 July 2022; IEEE: Piscataway, NJ, USA; pp. 959–962. [CrossRef]
6. Ariff, S.A.M.; Azri, S.; Ujang, U.; Nasir, A.A.M.; Ahmad Fuad, N.; Karim, H. Exploratory study of 3d point cloud triangulation for smart city modelling and visualization. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2020**, *44*, 71–79. [CrossRef]
7. Anjomshoaa, A.; Duarte, F.; Rennings, D.; Matarazzo, T.J.; deSouza, P.; Ratti, C. City Scanner: Building and Scheduling a Mobile Sensing Platform for Smart City Services. *IEEE Internet Things J.* **2018**, *5*, 4567–4579. [CrossRef]
8. Koszyk, J.; Łabędź, P.; Grzelka, K.; Jasińska, A.; Pargieła, K.; Malczewska, A.; Strząbała, K.; Michalczak, M.; Ambroziński, Ł. Evaluation of lidar odometry and mapping based on reference laser scanning. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2023**, *48*, 79–84. [CrossRef]
9. Pargieła, K. Optimising UAV Data Acquisition and Processing for Photogrammetry: A Review. *GaEE* **2023**, *17*, 29–59. [CrossRef]
10. Liu, Y. Application of Remote Sensing Technology in Smart City Construction and Planning. *J. Phys. Conf. Ser.* **2023**, *2608*, 012052. [CrossRef]
11. Wu, W.; Wang, W. LiDAR Inertial Odometry Based on Indexed Point and Delayed Removal Strategy in Highly Dynamic Environments. *Sensors* **2023**, *23*, 5188. [CrossRef]
12. Yang, C.; Zhang, F.; Gao, Y.; Mao, Z.; Li, L.; Huang, X. Moving Car Recognition and Removal for 3D Urban Modelling Using Oblique Images. *Remote Sens.* **2021**, *13*, 3458. [CrossRef]
13. Bodis-Szomoru, A.; Riemenschneider, H.; Van Gool, L. Efficient Volumetric Fusion of Airborne and Street-Side Data for Urban Reconstruction. In Proceedings of the 2016 23rd International Conference on Pattern Recognition (ICPR), Cancun, Mexico, 4–8 December 2016; IEEE: Piscataway, NJ, USA; pp. 3204–3209. [CrossRef]
14. Kedzierski, M.; Fryskowska, A. Methods of Laser Scanning Point Clouds Integration in Precise 3D Building Modelling. *Measurement* **2015**, *74*, 221–232. [CrossRef]
15. Blaszczak-Bak, W.; Masiero, A.; Bąk, P.; Kuderko, K. Integrating Data from Terrestrial Laser Scanning and Unmanned Aerial Vehicle with LiDAR for BIM Developing. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2024**, *48*, 25–30. [CrossRef]
16. Lee, E.; Kwon, Y.; Kim, C.; Choi, W.; Sohn, H.-G. Multi-Source Point Cloud Registration for Urban Areas Using a Coarse-to-Fine Approach. *GIScience Remote Sens.* **2024**, *61*, 2341557. [CrossRef]
17. Li, Z.; Jin, F.; Wang, J.; Zhang, Z.; Zhu, L.; Sun, W.; Chen, X. Adaptive Fusion of Different Platform Point Cloud with Improved Particle Swarm Optimization and Supervoxels. *Int. J. Appl. Earth Obs. Geoinf.* **2024**, *130*, 103934. [CrossRef]

18. Abdelazeem, M.; Elamin, A.; Afifi, A.; El-Rabbany, A. Multi-Sensor Point Cloud Data Fusion for Precise 3D Mapping. *Egypt. J. Remote Sens. Space Sci.* **2021**, *24*, 835–844. [CrossRef]
19. Che Ku Abdullah, C.K.A.F.; Baharuddin, N.Z.S.; Ariff, M.F.M.; Majid, Z.; Lau, C.L.; Yusoff, A.R.; Idris, K.M.; Aspuri, A. Integration of point clouds dataset from different sensors. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2017**, *42*, 9–15. [CrossRef]
20. Altuntas, C. Three-dimensional digitization of environments and buildings for smart city applications. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2021**, *46*, 65–71. [CrossRef]
21. Wang, R.; Peethambaran, J.; Chen, D. LiDAR Point Clouds to 3-D Urban Models$:$ A Review. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2018**, *11*, 606–627. [CrossRef]
22. Harshit; Chaurasia, P.; Zlatanova, S.; Jain, K. Low-Cost Data, High-Quality Models: A Semi-Automated Approach to LOD3 Creation. *IJGI* **2024**, *13*, 119. [CrossRef]
23. Lee, E.; Park, S.; Jang, H.; Choi, W.; Sohn, H.-G. Enhancement of Low-Cost UAV-Based Photogrammetric Point Cloud Using MMS Point Cloud and Oblique Images for 3D Urban Reconstruction. *Measurement* **2024**, *226*, 114158. [CrossRef]
24. Dai, Y.; Kim, D.; Lee, K. An Advanced Approach to Object Detection and Tracking in Robotics and Autonomous Vehicles Using YOLOv8 and LiDAR Data Fusion. *Electronics* **2024**, *13*, 2250. [CrossRef]
25. Wang, J.; Li, H.; Xu, Z.; Xie, X. Semantic Segmentation of Urban Airborne LiDAR Point Clouds Based on Fusion Attention Mechanism and Multi-Scale Features. *Remote Sens.* **2023**, *15*, 5248. [CrossRef]
26. Wicaksono, S.B.; Wibisono, A.; Jatmiko, W.; Gamal, A.; Wisesa, H.A. Semantic Segmentation on LiDAR Point Cloud in Urban Area Using Deep Learning. In Proceedings of the 2019 International Workshop on Big Data and Information Security (IWBIS), Bali, Indonesia, 11 October 2019; IEEE: Piscataway, NJ, USA; pp. 63–66. [CrossRef]
27. Soilán, M.; Riveiro, B.; Martínez-Sánchez, J.; Arias, P. Segmentation and Classification of Road Markings Using MLS Data. *ISPRS J. Photogramm. Remote Sens.* **2017**, *123*, 94–103. [CrossRef]
28. Hu, Q.; Yang, B.; Xie, L.; Rosa, S.; Guo, Y.; Wang, Z.; Trigoni, N.; Markham, A. RandLA-Net: Efficient Semantic Segmentation of Large-Scale Point Clouds. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; IEEE: Piscataway, NJ, USA; pp. 11105–11114. [CrossRef]
29. Qi, C.R.; Yi, L.; Su, H.; Guibas, L.J. PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space. *arXiv* **2017**. [CrossRef]
30. Charles, R.Q.; Su, H.; Kaichun, M.; Guibas, L.J. PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; IEEE: Piscataway, NJ, USA; pp. 77–85. [CrossRef]
31. Che, E.; Jung, J.; Olsen, M.J. Object Recognition, Segmentation, and Classification of Mobile Laser Scanning Point Clouds: A State of the Art Review. *Sensors* **2019**, *19*, 810. [CrossRef]
32. Klapa, P.; Gawronek, P. Synergy of Geospatial Data from TLS and UAV for Heritage Building Information Modeling (HBIM). *Remote Sens.* **2022**, *15*, 128. [CrossRef]
33. Uciechowska-Grakowicz, A.; Herrera-Granados, O.; Biernat, S.; Bac-Bronowicz, J. Usage of Airborne LiDAR Data and High-Resolution Remote Sensing Images in Implementing the Smart City Concept. *Remote Sens.* **2023**, *15*, 5776. [CrossRef]
34. Jovanović, D.; Milovanov, S.; Ruskovski, I.; Govedarica, M.; Sladić, D.; Radulović, A.; Pajić, V. Building Virtual 3D City Model for Smart Cities Applications: A Case Study on Campus Area of the University of Novi Sad. *IJGI* **2020**, *9*, 476. [CrossRef]
35. BLKARC_SpecSheet.Pdf. Available online: https://shop.leica-geosystems.com/sites/default/files/2023-11/BLKARC_SpecSheet.pdf (accessed on 14 February 2024).
36. DJI Zenmuse L1 + DJI Terra | SNH Drones. Available online: https://snhdrones.pl/produkt/dji-zenmuse-l1 (accessed on 24 July 2024).
37. Besl, P.J.; McKay, N.D. A Method for Registration of 3-D Shapes. *IEEE Trans. Pattern Anal. Mach. Intell.* **1992**, *14*, 239–256. [CrossRef]
38. Aerial Lidar Semantic Segmentation Using PointNet++ Deep Learning—MATLAB & Simulink. Available online: https://www.mathworks.com/help/lidar/ug/aerial-lidar-segmentation-using-pointnet-network.html (accessed on 24 July 2024).
39. Varney, N.; Asari, V.K.; Graehling, Q. DALES: A Large-Scale Aerial LiDAR Data Set for Semantic Segmentation. *arXiv* **2020**. [CrossRef]
40. Tan, W.; Qin, N.; Ma, L.; Li, Y.; Du, J.; Cai, G.; Yang, K.; Li, J. Toronto-3D: A Large-Scale Mobile LiDAR Dataset for Semantic Segmentation of Urban Roadways. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Seattle, WA, USA, 14–19 June 2020; IEEE: Piscataway, NJ, USA; pp. 797–806. [CrossRef]
41. Ester, M.; Kriegel, H.-P.; Sander, J.; Xu, X. A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. In Proceedings of the Second International Conference on Knowledge Discovery and Data Mining, Portland OR, USA, 2–4 August 1996; KDD'96; AAAI Press: Palo Alto, CA, USA, 1996; pp. 226–231.
42. Li, Z.; Wu, B.; Li, Y.; Chen, Z. Fusion of Aerial, MMS and Backpack Images and Point Clouds for Optimized 3D Mapping in Urban Areas. *ISPRS J. Photogramm. Remote Sens.* **2023**, *202*, 463–478. [CrossRef]

43. Aijazi, A.K.; Checchin, P.; Trassoudaine, L. Automatic Removal of Imperfections and Change Detection for Accurate 3D Urban Cartography by Classification and Incremental Updating. *Remote Sens.* **2013**, *5*, 3701–3728. [CrossRef]
44. Zhu, Q.; Fan, L.; Weng, N. Advancements in Point Cloud Data Augmentation for Deep Learning: A Survey. *Pattern Recognit.* **2024**, *153*, 110532. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.