Periodica Polytechnica Civil Engineering, 68(4), pp. 1021–1030, 2024

Application of Machine Learning to Detect Building Points in Photogrammetry-based Point Clouds

Bence Péter Hrutka[1*]

[1] Department of Geodesy and Surveying, Faculty of Civil Engineering, Budapest University of Technology and Economics, Műegyetem rkp. 3., H-1111 Budapest, Hungary
[*] Corresponding author, e-mail: hrutka.bence@emk.bme.hu

Received: 03 May 2023, Accepted: 29 March 2024, Published online: 29 May 2024

Abstract

Different point cloud technologies such as Terrestrial Laser Scanners (TLS), Airborne Laser Scanners (ALS), Mobile Mapping Systems (MMS), and Unmanned Aerial Vehicles (UAV) have become increasingly more common in land surveying and geoinformatics over recent years. Thanks to these modern tools, experts can survey large areas cost-effectively with either high resolution or high accuracy. However, processing the point cloud, which consists of millions of points, can be a massive challenge. Manual processing of these large datasets can often be very time-consuming and hardware-demanding, and most of the time, only a limited part of the point cloud is used to derive the final products. The solution can be to automate the process as much as possible. Several advanced mathematical methods, especially Machine Learning (ML) algorithms, allow efficient automated processing of point clouds. This paper presents a processing chain to detect and separate building points from large-scale photogrammetry-based point clouds. The processing is based on the combination of Random Sample Consensus (RANSAC) and Machine Learning (ML) algorithms like Density-Based Spatial Clustering of Applications with Noise (DBSCAN) and Multi-Layer Perceptron (MLP). Presented methods were trained and tested on established and open available Heissigheim 3D (H3D) dataset to separate roof and vegetation points with over 90% accuracy in order to enhance the separation of building points on large-scale point clouds.

Keywords

UAV, point cloud, building reconstruction, classification, machine learning

1 Introduction

Unmanned Aerial Vehicle (UAV) photogrammetry-based measurements have become popular in the last two decades. There are several applications, from geoscience to engineering, where UAVs' utility has been firmly proven. With drones' flexibility and relatively low costs, experts can efficiently develop maps of large areas with high speed and resolution [1]. One of the many applications is updating land registry maps, where Unmanned Aerial Vehicles can supply fast and affordable solutions without entering private properties. In the literature, there are several examples where UAV photogrammetry-based point clouds were used to update old analog maps.

One of the first attempts was in the Netherlands in 2013 [2], where UAV photogrammetry-based point clouds and true-orthophotos were derived from high-resolution photos to identify property boundaries. As a result of the test, it was found that the required accuracy of land registry mapping is achievable. GNSS measurements were applied to check the reliability of the final product, and it proved to be below 10 cm. Similar studies were obtained in Albania [3] and Poland [4] to support land registration and improve the quality of existing maps. These studies have produced equivalent results.

While UAV-based photogrammetry is a prominent method for generating point clouds with the required accuracy, drones deployed with Light Detection and Ranging (LiDAR) are also increasingly prevalent in mapping applications. A study by He and Li [5] illustrated that LiDAR sensor-based campaigns could also be used in land registry mapping with 5–10 cm of accuracy. A comparable investigation was conducted in the Czech Republic [6], where an experiment was also carried out to compare the results of photogrammetry- and LiDAR-based UAV measurements. The results showed that both technologies achieved the required level of accuracy.

As the conclusion of the aforementioned articles, it can be stated with full confidence that point cloud techniques

Cite this article as: Hrutka, B. P. "Application of Machine Learning to Detect Building Points in Photogrammetry-based Point Clouds", Periodica Polytechnica Civil Engineering, 68(4), pp. 1021–1030, 2024. https://doi.org/10.3311/PPci.22496

can be used to update and renew land registry maps. However, the presented solutions are primarily based on manual processing. This paper will examine how automation is possible and how automation can support the production of 2D maps and 3D models.

Working on point clouds, several segmentation methods, such as edge-based, region-growing, attribute-based, model-based, and Machine Learning can support data processing. Grilli et al [7] demonstrated that by combining these techniques, efficient filtering can be achieved and used to segment groups of points from point clouds. Furthermore, augmenting these techniques with advanced mathematical methodologies enables high-level point cloud classification tasks.

Aerial Laser Scanning (ALS) campaign results were widely used to detect separate parts of point clouds. For instance, Shao et al [8] showcased that combining Cloth Simulation Filtering (CSF) and region growing can effectively detect building roofs in large areas. Another approach entailed fusing region growing with elevation histogram analysis to separate roof points from point clouds [9]. Szutor [10] proposed a combination of two clustering methods and that the Fast Fourier Transform (FFT) can also be used for point cloud classification.

Nowadays, the application of machine learning algorithms for different tasks, such as point cloud processing, is inescapable. Combining other segmentation methods with machine learning can be effective during point cloud classification tasks. Zeybek [11] demonstrated that UAV-based dense point clouds can be classified with the Random Forest ML algorithm according to their geometric characteristics and radiometric features. The overall accuracy of the classification was achieved at 96%. Chen et al. [12] applied a Support Vector Machines-based solution on ALS-based campaign's result, where combined features including the coordinates, the RGB values, normalized elevations, standard deviations of heights, and elevation differences of point cloud were used with overall classification accuracies of 97.69% and 99.13% for two test areas.

In addition to Machine Learning, it is unavoidable to involve Deep Learning (DL) methods. Unlike conventional Machine Learning approaches, where feature engineering is typically a manual and time-consuming task, DL models have the ability to directly extract significant features from raw data. In the case of point clouds, it can be completed on voxels [13], regions [14], or in a certain radius [15].

This article introduces a potential application of a Multi-Layer Perceptron algorithm to enhance the efficiency of point classification in large-scale UAV photogrammetry-based point clouds. The training data for the neural network was meticulously crafted by human intervention. The trained model was applied to three test areas, mainly with detached houses, to separate roof and vegetation points on pre-segmented point clouds. Building points were separated and classified in the point clouds using the separated roof points.

2 Materials and methods

The application and combination of segmentation methods offer a large variety of tools during point cloud processing. In the following, an efficient workflow from data collection to segmentation is presented to separate building points.

2.1 Field measurements

Before delving into the processing, it's imperative to ensure the input data meets requisite quality standards. Therefore, it is essential to lay down the parameters of campaigns. Focusing on buildings, UAV mission's flight parameters can significantly influence the number of points mapped on roofs and walls (Fig. 1). The best result could be achieved with a double grid mission and oblique camera at 20–30 degrees. To reach an accuracy of 10 cm, 2 cm of Ground Sample Distance (GSD) is required.

Ground Control Points (GCPs) are also required to improve the accuracy of the point cloud It is enough to measure GCPs position by RTK GNSS technique in order to have residuals at the GCPs around 1 cm.

2.2 Pre-processing

Working on large-scale point clouds, it is essential to pre-process the original point cloud to decrease the number of points before the main processing. Applying noise removal (e.g., Statistical Outlier Removal) is necessary but not enough.

Fig. 1 Large-scale point cloud of Barnag settlement

The simplest way would be to use a subsample, which can yield to lose even valuable points. Focusing on building points, the processing can be narrowed down to off-ground points. In practice, there are several algorithms. One of these is the Cloth Simulation Filter (CSF) algorithm [16], which can effectively separate ground and non-ground points regardless of point cloud origin. In case of settlements where detached houses are typical, the original point clouds' size can be reduced by nearly 70–80% without losing relevant data on buildings [17].

Another crucial aspect is a significant difference between the number of points generated on roofs and on other building parts, such as walls. Approximately the number of points generated on roofs is four times the number of wall points. Fig. 2 shows these differences by colorizing the calculated point density in a sphere of a specified radius (r = 0.30 m).

Dealing with the classification of building points on large-scale photogrammetry-based results, a method that considers this difference during processing should be used. During later process steps, rough separation of the roof and wall points is required, where using the points' relative height can be effective. Therefore, a Digital Terrain Model (DTM) can be generated using the separated ground points. By computing vertical differences of the non-ground points from the DTM, a Normalized Digital Surface Model (nDSM) can be created (Fig. 3).

2.3 Voxelization and segmentation of possible roof and wall points

By assuming that the roofs and walls of buildings mainly consist of planar surfaces, the processing can be focused on finding points that fit a plane within a specific threshold. A practical solution to this problem is offered by Sequential Random Sample Consensus (RANSAC) [18]. Rather than applying RANSAC across the entire area, the point cloud can be subdivided into voxels to enhance the algorithm's efficiency. Sequential RANSAC can be used to find significant planar surfaces with more details in a voxel. Another advantage of voxelization is that the process can be parallelized, which makes the processing more efficient on large-scale point clouds [17].

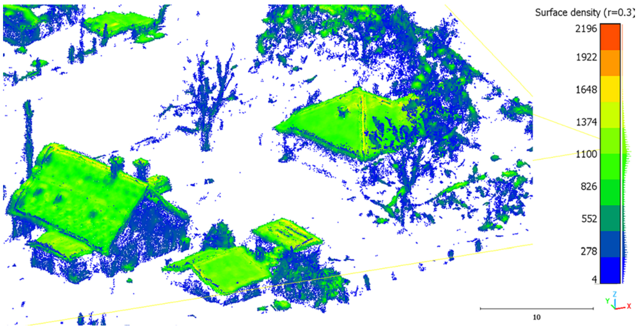

Fig. 2 Normalized point cloud colorized by surface density values (lower density is blue, medium density is green and high density is red)

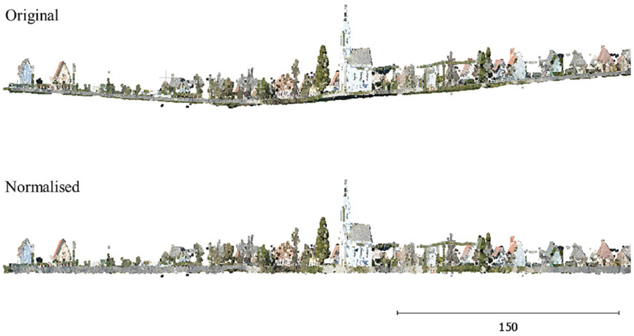

Fig. 3 Vertical sections of original and normalized point clouds

Utilizing the relative heights and parameters of the planes detected during the process, another filtering can be implemented to separate possible roof and wall points based on height values and the orientation of the normal vectors of the identified planes.

Majority of the noise and vegetation points can be filtered out thanks to robust model-based filtering, but there are specific cases where voxel-based RANSAC is less effective (Fig. 4). This limitation arises from RANSAC's failure to account for the spatial continuity of points on a plane. In these cases, the application of different ML algorithms could be practical.

2.4 Applying machine learning algorithms

Machine Learning (ML) methods are becoming increasingly widespread over the last few decades. There are hundreds of applications that can be solved by using

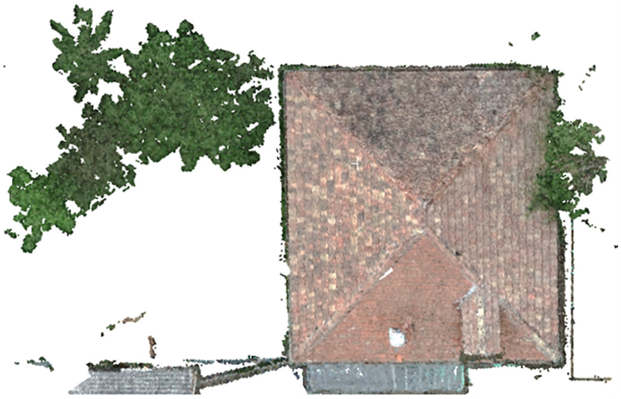

Fig. 4 Extreme case, where sequential RANSAC does not work efficiently

ML algorithms. In the following, I will expand on these by presenting a possible application of Multi-Layer Perceptron (MLP) and Density-Based Spatial Clustering of Applications with Noise (DBSCAN) algorithms for point cloud processing.

2.4.1 Prepare training data

In general, preparation is essential before applying Machine Learning algorithms, such as the MLP. This typically involves the creation of training, testing, and validation of data. Since my aim is to separate vegetation points from roof points, the classes are predefined. Through a manual process, these data can be assigned to each class. In Fig. 5, an example of colorized training data is shown.

2.4.2 3D feature calculation

Some preparatory steps are also required for the training data. Machine Learning algorithms often rely on indirectly defined values, known as features, to classify data effectively.

In the case of point clouds, these features can be derived from attributes such as eigenvalues (normal, sum, omnivariance, eigentropy, anisotropy, planarity, linearity, surface variation, sphericity), spectral (RGB, intensity), and geometrical features (different coordinate components, area, local point density) [19]. Values used during training usually depend on the input data and task. Identifying the necessary attributes (Fig. 6) is predicated on evaluating how effectively separation can be achieved.

Therefore, before training, each possible eigenvalue-based and geometric feature was computed within a specified radius sphere and visualized [20]. The most relevant attributes can be selected from these values, according to which can be used to separate different parts in a UAV photogrammetry-based point cloud. The chosen features for training are summarized in Table 1 with their equations:

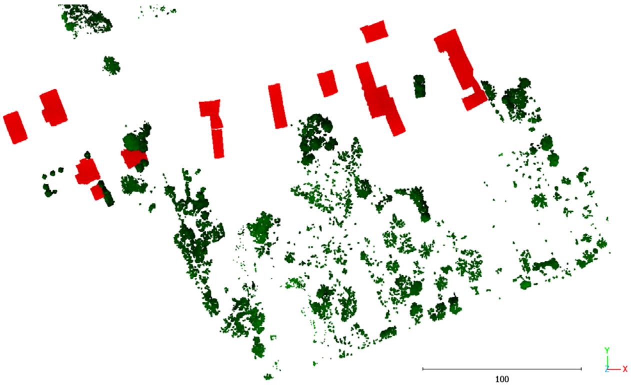

Fig. 5 Manually segmented training data of roofs (red) and vegetation (green)

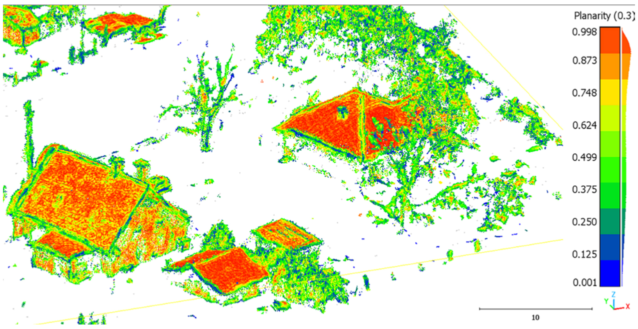

Fig. 6 Normalized point cloud segment colorized by eigen-based planarity values (lower density is blue, medium density is green, and high density is red)

Table 1 Eigenvalue in decreasing order ($\lambda_1 > \lambda_2 > \lambda_3$), geometric, and spectral-based features for point cloud classification

Feature	Equation
Relative height	Normalized Z-coordinate
Color values	R, G, B
Omnivariance	$\sqrt[3]{\lambda_1 \times \lambda_2 \times \lambda_3}$
Anisotropy	$\frac{(\lambda_1 - \lambda_3)}{\lambda_1}$
Planarity	$\frac{(\lambda_2 - \lambda_3)}{\lambda_1}$
Surface variation	$\frac{\lambda_3}{(\lambda_1 + \lambda_2 + \lambda_3)}$
Sphericity	$\frac{\lambda_3}{\lambda_1}$
Number of neighbors	N
Surface density	$\frac{N}{V_{\text{sphere}}}$
Normal change rate	$\frac{\lambda_1}{(\lambda_1 + \lambda_2 + \lambda_3)}$

2.4.3 Multi-Layer Perceptron

Following the selection and preparation of training data, Multi-Layer Perceptron (MLP) algorithm can be used to train, and a model can be created to separate possible roof points from vegetation points in a pre-segmented normalized point cloud.

MLP is a neural network architecture widely used in machine learning for regression and classification tasks. MLPs are usually arranged in three layers:

- input layer
- hidden layer(s)
- output layer.

Input and output layers are connected through neurons within the hidden layer (Fig. 7). MLP models may have one or more hidden layers depending on the complexity of the task.

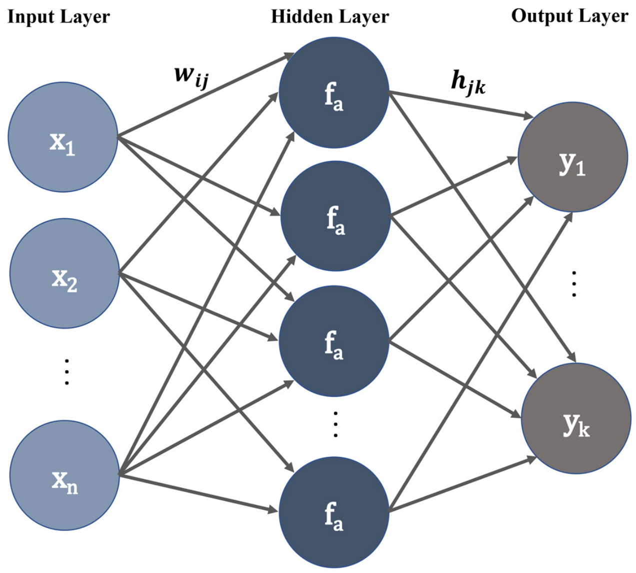

Fig. 7 Architecture of a Multi-Layer Perceptron (MLP) network with one hidden layer and f_i activation functions

A special feature of ML algorithms is that the relationship between the input and output data is unknown. The connection is established during the training process, whose main purpose is to find a correlation between the input and output through the hidden layers' neurons.

In each node (or neuron) of the hidden layer, an activation function (f_a) can be found. Depending on the task, a wide range of activation or transfer functions (such as linear, sigmoid, polynomial, softmax, and rectified linear unit, so-called ReLu) can be utilized.

Each activation function (neuron) relates to inputs (x_i) and is characterized by their weight (w_{ij}). The input is split into components, multiplied by the corresponding weight, and summed. An optional bias can be added to this summation. At the end of the process, a resultant sum is filtered by different activation functions (f_a), and output is produced (Fig. 8). In MLP, these outputs (h_{jk}) can also be inputs of another hidden layer.

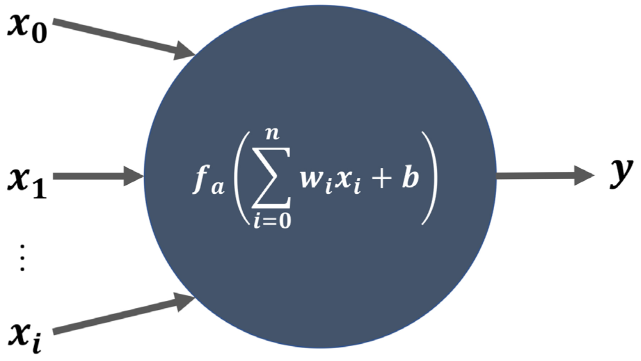

Fig. 8 Schematic representation of a neuron with input features (x_i), an activation function (f_a) and output (y)

Since the input and output values are known during the training process, the optimization step aims to determine the weight and bias values. This calculation is performed iteratively, facilitated by the utilization of a loss function.

The role of the loss function is to assess the disparity between the target and predicted results. By iteratively adjusting the weights and biases based on this assessment, the model endeavors to minimize the loss, thereby optimizing its predictive capability.

$$L = \frac{1}{2}\sum_i \left\| \overline{y}_i^{\text{Predicted}} - \overline{y}_i^{\text{Target}} \right\|^2 . \tag{1}$$

Using the result of loss function (L), gradients can be calculated to update weight values, and in the following iteration step, weights can be updated until a given threshold is reached. Through training, the neural network strives to converge towards an optimal set of weights, and the result is saved as a model for regression or classification tasks [21–23].

Multi-Layer Perceptron can be used to separate vegetation and roof points. Different formulas based on eigenvalues can be used as features to classify points with the MLP method.

2.4.4 Performance metrics

To ensure the performance of the model, different performance metrics can be utilized via validation and test data.

Among the many metrics used to describe the created model, accuracy stands out as a fundamental measure of the model's effectiveness. Accuracy is computed as the ratio of the number of correct predictions ($N_{\text{correct pred.}}$) to the total sample ($N_{\text{total pred.}}$):

$$\text{Accuracy} = \frac{N_{\text{correct pred.}}}{N_{\text{total pred.}}} . \tag{2}$$

While accuracy provides a straightforward measure of the model's overall correctness, other metrics offer deeper insights into its performance. These metrics are often encapsulated in a confusion matrix, which comprises four key elements:

- **True Positive (TP)**: number of correctly predicted true cases
- **True Negative (TN)**: number of correctly predicted false cases
- **False Positive (FP)**: number of falsely predicted true cases (Type I Error)
- **False Negative (FN)**: number of falsely predicted false cases (Type II Error).

From elements of the confusion matrix, additional metrics can be calculated to describe the model performance from different points of view. One of these metrics is recall or sensitivity, which describes how many times the model mistakenly diagnosed a positive element as negative:

$$\text{Recall} = \frac{TP}{TP + FN} . \tag{3}$$

Precision or specificity is the opposite of recall. It shows from the predicted positives how many were positive:

$$\text{Precision} = \frac{TP}{TP + FP} . \tag{4}$$

To summarize the performance of the classifier with a single metric, *F*–score can be calculated as the harmonic mean of the precision and recall:

$$F - \text{score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} . \tag{5}$$

These performance metrics provide valuable feedback about the model's efficacy.

On the one hand, by applying the trained model to the test data, we can get an overview of the performance of the model.

Relying solely on training and test data to evaluate the model's performance may lead to overfitting. To mitigate this issue, validation data is used to provide examples that the model has not seen before. After each training epoch or a certain number of training iterations, the model's performance is evaluated using the validation data. Presented metrics such as loss, accuracy, precision, or recall can be calculated on the validation set to gauge the model's performance. This process allows practitioners to monitor the model's progress during training, tune hyperparameters, and detect potential issues such as overfitting.

2.4.5 DBSCAN

After applying MLP to segment the roof and vegetation points, the next step is to separate the roof by buildings. There are a wide variety of clustering methods, such as *k*-Means, Mean-shift, Spectral clustering, Density-Based Spatial Clustering of Applications with Noise (DBSCAN), Ordering Points to Identify the Clustering Structure (OPTICS), and Balanced Iterative Reducing and Clustering using Hierarchies (BIRCH) providing the possibility to separate groups of points from each other [21]. In most of these methods, the number of clusters needs to be predefined, but in the case of point clouds, the number of clusters is usually unknown.

In many cases, Density-Based Spatial Clustering of Applications with Noise [24] emerges as a viable choice. The main idea (Fig. 9) of the algorithm is to search high-density areas surrounded by low-density ones.

DBSCAN algorithms rely on two parameters:

- ε – defines the maximum distance (radius) between two neighbors.
- $n_{\min}$ – defines how many surrounding points are necessary to define a core.

The procedure is mainly based on a metric function (usually the Euclidean distance). For each sample x_i, the algorithm checks the distance to other x_j samples. The number of neighboring points within the ε radius is at least $n_{\min}$, the sample is identified as a core point:

$$N\left(d\left(\bar{x}_i, \bar{x}_j,\right) \le \varepsilon\right) \ge n_{\min} . \tag{6}$$

From a core point, other x_j points' distances can be checked:

$$d\left(\bar{x}_i, \bar{x}_j\right) \le \varepsilon . \tag{7}$$

In that way, the size of the core points is expandable until the samples do not meet the requirements. If there are no more core points for a group, a cluster is found, samples can be labeled, and the search for other clusters can be continued. Points that are not part of any cluster are considered as noise [23].

DBSCAN algorithm can be effectively utilized to segment roof points into distinct groups corresponding to individual buildings. It is also a valuable tool for removing outlier points that may have persisted from previous processing steps.

2.5 Building point detection

Clustering the roof points, the lower part of the normalized point cloud can be analyzed to potentially separate wall points. This can be achieved by constructing the

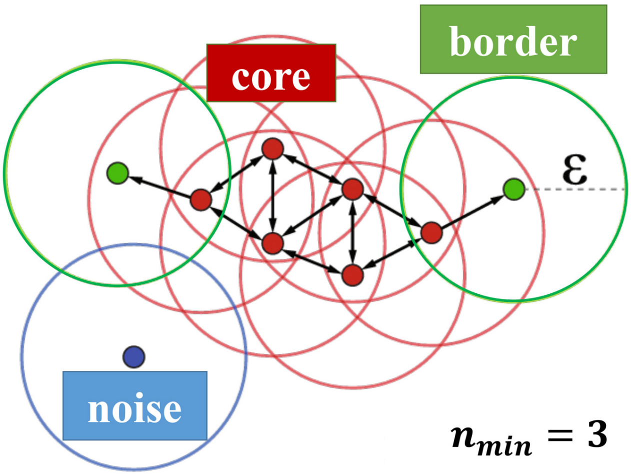

Fig. 9 Visualization of the DBSCAN method

convex hull for each clustered roof, essentially creating bounding polygons around each roof.

Once these convex hulls are established as bounding polygons, the wall points can be filtered based on their proximity to these polygons. Points falling within the convex hulls represent the walls of the buildings.

This approach can automatically process the point cloud data to produce separated point clouds for individual buildings. (Fig. 10).

For a concise overview, refer to summary flowcart of Fig. 11, which illustrates the key steps and processes discussed in the study.

3 Measurements and processing

The methods and algorithms described were implemented and evaluated across three distinct test areas [17]. During the campaigns, the photos were taken in each area by a DJI Phantom 4 Pro UAV, and Ground Control Points (GCPs) were measured by RTK GNSS technique. Standard photogrammetric processing was carried out to derive dense point clouds. [25]. The residuals on the GCPs were below 1 cm. The parameters of each flight are summarized in Table 2.

On each test area, nDSM was generated, voxel-based sequential RANSAC was applied, and possible roof and wall points were separated by between 0.3 and 2 meters of relative heights and normal directions.

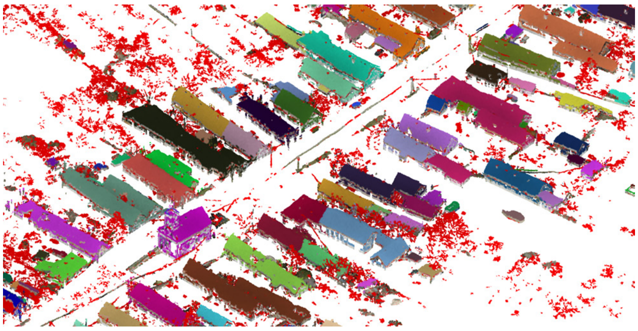

Fig. 10 Clustered building points colorized by random colors and vegetation with red color

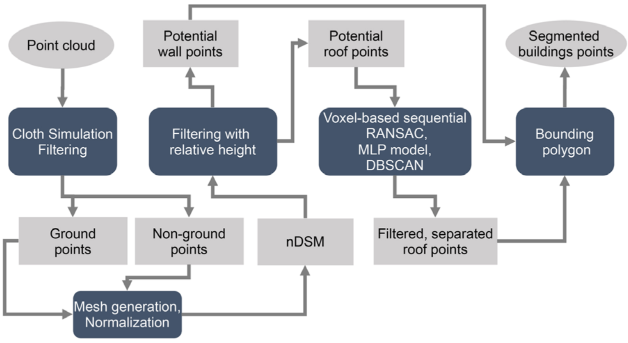
Point cloud
Cloth Simulation Filtering
Ground points
Non-ground points
Mesh generation, Normalization
nDSM
Filtering with relative height
Potential wall points
Potential roof points
Voxel-based sequential RANSAC, MLP model, DBSCAN
Filtered, separated roof points
Bounding polygon
Segmented buildings points

Fig. 11 Flowchart of the proposed point cloud processing

Table 2 Flying parameters on each test area

	Barnag	Üllő	Sződ
Mission type	Simple grid	Double grid	Double grid
Above Ground Level (AGL) [m]	55 m	55 m	70 m
Image overlap [%]	80	80	80
Oblique angle [°]	0 and 25	25	20
Ground Sample Distance (GSD) [cm/pixel]	1.5	1.5	1.9
Number of photos	1400 and 890	805	799
Number of points	452 million	116 million	221 million

To train Multi-Layer Perceptron (MLP) neural network, training dataset of 1 million points was created manually from a small part of the Barnag test area (Fig. 12).

Eigenvalue-based formulas were calculated and applied in the range of 30 cm around each point. The features in Table 1 were transformed by scaling between zero and one. Thus, a 12-element feature vector was used in the input layer to differentiate roof and vegetation points from each other. During the training (Fig. 13), ReLu activation function was applied with a batch size of 1024. Adaptive Moment Estimation (ADAM) [22] method was used as a solver for weight optimization, and the learning rate was set to 0.001. The MLP architecture comprised a single hidden layer consisting of 15 neurons.

To ensure that overfitting does not affect the results, 10% of the training data was reserved for validation during model training. By comparing the training and model loss (Fig. 13), it can be seen that training loss and validation loss values are nearly the same. This means that a simple neural network can effectively separate datasets with well-selected features.

To finalize the processing pipeline, DBSCAN was employed with the ε value of 0.3 m and the minimal points of 100 to separate roofs by buildings. Wall points were derived by using each roof cluster's bounding polygon. Thus, at the end of the process, separated building points

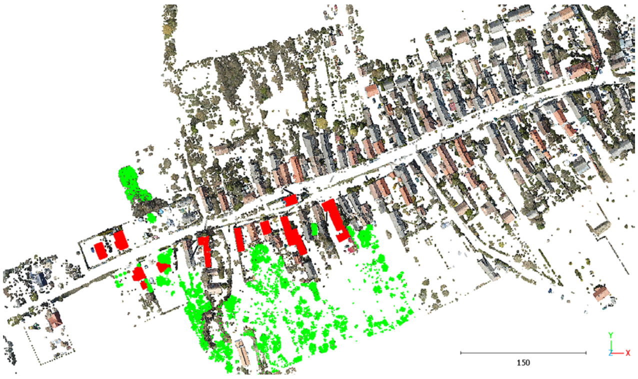

Fig. 12 Training data compared to the full Barnag test area

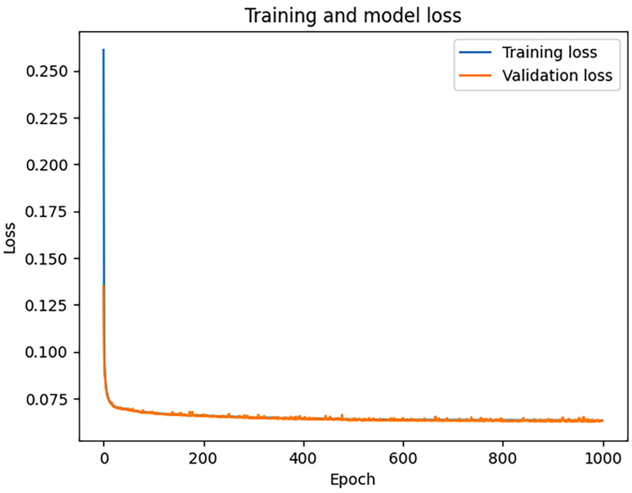

Fig. 13 Model accuracy during the training

were generated by combining the point cloud segmentation methods and machine learning techniques.

4 Results

When evaluating the results of the processing, particular attention should be paid to the performance of Multi-Layer Perceptron model. The overall accuracy of the model is 97.4%. This accuracy can be further understood by examining the elements of the confusion matrix (Fig. 14), which reveals that both Type I (false positive) and Type II (false negative) errors are minimal relative to the total number of training data.

Additional metrics can also be derived from the confusion matrix elements (Table 3). The performance of the created MLP model is summarized as follows:

These performance metrics affirm the effectiveness of the model in classifying roof and vegetation points. Consequently, the MLP model was applied to the entire Barang, Üllő, and Sződ test areas. The results of the extrapolated application of the model are depicted in the following figures with false-color displays (Fig. 15).

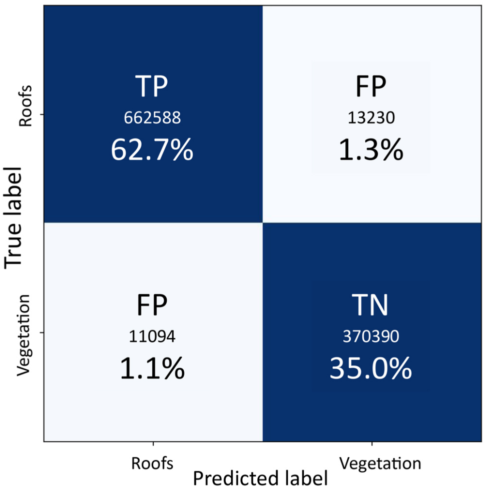

Fig. 14 Confusion matrix

Table 3 Performance of the created MLP model

	Precision	Recall	F_1-score	Num. of samples
Roofs	0.98	0.98	0.98	675818
Vegetation	0.96	0.97	0.96	381484
Accuracy	-	-	0.97	1057302
Macro avg.	0.97	0.97	0.97	1057302
Weighted avg.	0.97	0.97	0.97	1057302

Using the DBSCAN algorithm, roof patches were clustered by buildings, as illustrated in Fig. 16.

Utilizing the bounding polygons of the segmented roofs, potential wall points were also discerned by filtering the

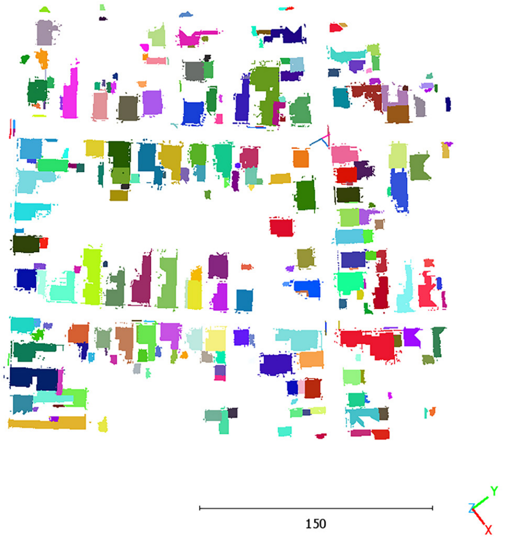

Fig. 16 Building clusters at Üllő test area

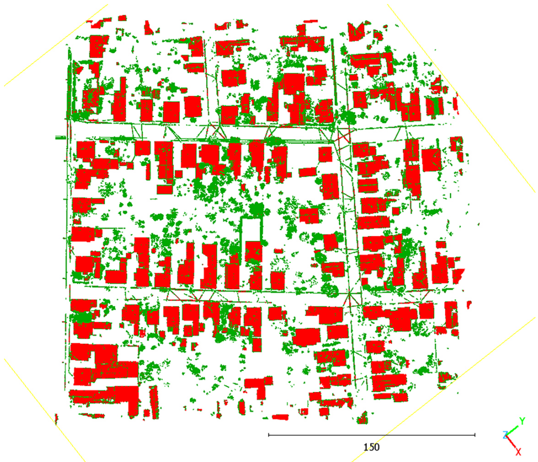

Fig. 15 Result of MLP-based classification at Üllő test area. Vegetation is represented by green, while roof points are represented by red.

remaining points located between 0.3 and 2 meters from the ground within each test area, as shown in Fig. 17.

MLP feature-based model achieved 93.5% accuracy on classifying roof and vegetation points on the Hessigheim 3D (H3D) benchmark dataset [26]. With the help of the process pipeline from the 23 975 119 building points 21 679 889 (90.4%) points were found on the dataset. Number of false detections was 1 648 546.

5 Conclusions

The primary objective of the study was to separate building points from UAV photogrammetry-based point clouds to develop 3D models and large-scale maps. To achieve this, voxel-based Random Sample Consensus and Machine Learning algorithms, such as Multi-Layer Perceptron and Density-Based Spatial Clustering of Applications with Noise were combined in an automated way.

The processing pipeline began with a normalization step by applying a Cloth Simulation Filter. Utilizing the normalized Digital Surface Model, a combination of height filtering and a voxel-based sequential RANSAC algorithm was applied to detect and separate possible roof and wall points from each other.

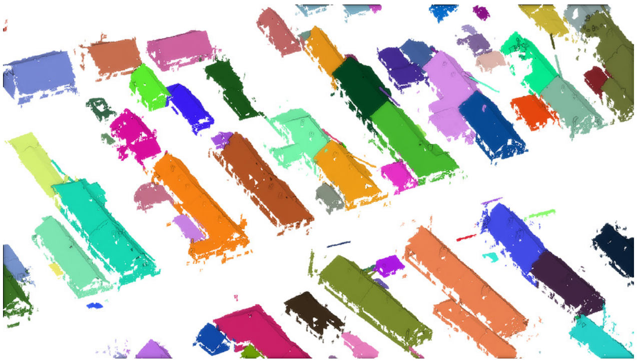

Fig. 17 Final result of the fully automated processing

Due to the specificity of the UAV photogrammetry-based point clouds, the processing has focused on separating roof points. Thus, machine learning techniques, namely MLP and DBSCAN were combined to extract roof points from the pre-segmented off-ground points.

For the MLP approach, a training dataset was established using measurements from one of the test areas (Barnag), with two labels: roof and vegetation points. After that, point cloud features were selected and computed at each point. A Multi-Layer Perceptron neural network model was trained to separate roof and vegetation points following voxel-based RANSAC segmentation. The resulting model achieved an overall accuracy of 97.4%, facilitating the separation of roof points.

Finally, a crop operation was executed using the bounding polygon of roofs to get the wall points of these roof clusters. Thus, the wall points were generated from the rest of the non-ground points, which were filtered out by the previous steps. At the end of the process, the building points have been produced for each building.

These methods were applied to test measurements from three different areas, yielding consistent results.

Data Availability Statement

The dataset used in this study is available upon request from the corresponding author.

Acknowledgement

The project presented in this article is supported by the ÚNKP-22-3-I-BME-38 New National Excellence Program of the Ministry for Culture and Innovation from the source of the National Research, Development and Innovation Fund.

The research was supported by the European Union within the framework of the National Laboratory for Autonomous Systems. (RRF-2.3.1-21-2022-00002)

References

[1] Nex, F., Armenakis, C., Cramer, M., Cucci, D. A., Gerke, M., Honkavaara, E., Kukko, A., Persello, C., Skaloud, J. "UAV in the advent of the twenties: Where we stand and what is next", ISPRS Journal of Photogrammetry and Remote Sensing, 184, pp. pp. 215–242, 2022. https://doi.org/10.1016/j.isprsjprs.2021.12.006

[2] Rijsdijk, M., van Hinsbergh, W. H. M., Witteveen, W., ten Buuren, G. H. M., Schakelaar, G. A., Poppinga, G., van Persie, M., Ladiges, R. "Unmanned Aerial Systems in the Process of Juridical Verification of Cadastral Border", The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, XL-1/W2, pp. 325–331, 2013. https://doi.org/10.5194/isprsarchives-xl-1-w2-325-2013

[3] Kedzierski, M., Fryskowska, A., Wierzbicki, D., Nerc. P. "Chosen aspects of the production of the basic map using UAV imgery", In: The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, XXIII ISPRS Congress, 2016, XLI-B1, pp. 873–877. https://doi.org/10.5194/isprsarchives-XLI-B1-873-2016

[4] Kurczynski, Z., Bakuła, K., Karabin, M., Markiewicz, J. S., Ostrowski, W., Podlasiak, P., Zawieska, D. "The Possibility of Using Images Obtained from the Uas in Cadastral Works", The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, XXIII ISPRS Congress, 2016, XLI-B1, pp. 909–915. https://doi.org/10.5194/isprsarchives-XLI-B1-909-2016

[5] He, G. B., Li, L. L. "Research and application of lidar technology in cadastral surveying and mapping", The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, XLIII-B1-2020, pp. 33–37, 2020. https://doi.org/10.5194/isprs-archives-XLIII-B1-2020-33-2020

[6] Šafář, V., Potůčková, M., Karas, J., Tlustý, J., Štefanová, E., Jančovič, M., Žofková, D. C. "The use of Uav in Cadastral mapping of the Czech Republic", ISPRS International Journal of Geo-Information, 10(6), 380, 2021. https://doi.org/10.3390/ijgi10060380

[7] Grilli, E., Menna F., Remondino F. "A review of point clouds segmentation and classification algorithms", The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, XLII-2/W3, pp. 339–344, 2017. https://doi.org/10.5194/isprs-archives-XLII-2-W3-339-2017

[8] Shao, J., Zhang, W., Shen, A., Mellado, N., Cai, S., Luo, L., Wang, N., Yan, G., Zhou, G. "Seed point set-based building roof extraction from Airborne LiDAR point clouds using a top-down strategy", Automation in Construction, 126, 103660, 2021. https://doi.org/10.1016/j.autcon.2021.103660

[9] Kurdi, F. T., Gharineiat, Z., Campbell, G., Awrangjeb, M., Dey, E. K. "Automatic Filtering of Lidar Building Point Cloud in Case of Trees Associated to Building Roof", Remote sensing, 14(2), 430, 2022. https://doi.org/10.3390/rs14020430

[10] Szutor, P. "FFT based Airborne LIDAR classification with Open3D and Numpy/Scipy", In: Proceedings of the 11th International Conference on Applied Informatics (ICAI 2020), Hungary, Eger, 451, pp. 350–357, 2020. [pdf] Available at: https://ceur-ws.org/Vol-2650/paper36.pdf [Accessed: 02 April 2023]

[11] Zeybek, M. "Classification of UAV point clouds by random forest machine learning algorithm", Turkish Journal of Engineering, 5(2), pp. 48–57, 2021. https://doi.org/10.31127/tuje.669566

[12] Chen, C., Li, X., Belkacem, A.N., Qiao, Z., Dong, E., Tan, W., Shin, D. "The mixed kernel function SVM-based point cloud classification", International Journal of Precision Engineering and Manufacturing, 20, pp. 737–747, 2019. https://doi.org/10.1007/s12541-019-00102-3

[13] Poux F, Billen R. "Voxel-based 3D point cloud semantic segmentation: unsupervised geometric and relationship featuring vs deep learning methods", ISPRS International Journal of Geo-Information, 8(5), 213, 2019. https://doi.org/10.3390/ijgi8050213

[14] Özdemir, E., Remondino, F., Golkar, A. "Aerial point cloud classification with deep learning and machine learning algorithms", The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, XLII-4/W18, pp. 843–849, 2019. https://doi.org/10.5194/isprs-archives-XLII-4-W18-843-2019

[15] Özdemir, E., Remondino, F. "Classification of aerial point clouds with deep learning", The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, XLII-2/W13, pp. 103–110, 2019. https://doi.org/10.5194/isprs-archives-XLII-2-W13-103-2019

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

[17] Hrutka, B. P., Siki, Z., Takács, B. "Voxel-based point cloud segmentation and building detection", The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, XLVIII-4/W1-2022, pp. 209–215, 2022. https://doi.org/10.5194/isprs-archives-XLVIII-4-W1-2022-209-2022

[18] Fischler, M. A., Bolles, R. C. "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography", Communications of the ACM, 24(6), pp. 381–395, 1981. https://doi.org/10.1145/358669.358692

[19] Feng, C. C., Guo, Z. "Automating parameter learning for classifying terrestrial LiDAR point cloud using 2D land cover maps", Remote Sensing, 10(8), 1192. 2018. https://doi.org/10.3390/rs10081192

[20] CloudCompare Development Team "CloudCompare software, 2.12.4 Kyiv" [online] Available at: https://www.danielgm.net/cc/ [Accessed: 02 April 2023]

[21] Bonaccorso, G. "Machine learning algorithms – second edition", Packt, 2018. ISBN 9781789347999 [online] Available at: https://www.packtpub.com/product/machine-learning-algorithms-second-edition/9781789347999 [Accessed: 02 April 2023]

[22] Elgendy, M. "Deep learning for vision systems", Manning, 2020. ISBN 9781617296192. [online] Available at: https://www.manning.com/books/deep-learning-for-vision-systems [Accessed: 02 April 2023]

[23] Awange, J., Paláncz, B., Völgyesi, L. (eds.) "Hybrid imaging and visualization employing machine learning with mathematica – Python", Springer, 2020. ISBN 978-3-030-26152-8 [online] Available at: https://link.springer.com/book/10.1007/978-3-030-26153-5 [Accessed: 02 April 2023]

[24] Ester, M., Kriegel, H. P., Sander, J., Xiaowei, Xu. "A density-based algorithm for discovering clusters in large spatial databases with noise", In: International conference on knowledge discovery and data mining, Portland, OR, USA, 1996. (CONF-960830-; TRN: 96:005928-0038), [online], Available at: https://www.osti.gov/biblio/421283 [Accessed: 02 April 2023]

[25] 3Dsurvey "3Dsurvey software" [online] Available at: https://3dsurvey.si [Accessed: 02 April 2023]

[26] Kölle, M, Laupheimer, D., Schmohl, S., Haala, N., Rottensteiner, F., Wegner, J. D., Ledoux, H. "The Hessigheim 3D (H3D) benchmark on semantic segmentation of high-resolution 3D point clouds and textured meshes from UAV LiDAR and Multi-View-Stereo", ISPRS Open Journal of Photogrammetry and Remote Sensing, 1, 100001, 2021. https://doi.org/10.1016/j.ophoto.2021.100001