# Automatic Segmentation of Individual Grains From a Terrestrial Laser Scanning Point Cloud of a Mountain River Bed

Agata Walicka and Norbert Pfeifer

***Abstract*—In this article, we propose a method for instance segmentation of individual grains from a terrestrial laser scanning point cloud representing a mountain river bed. The method was designed as a classification followed by a segmentation approach. The binary classification into either points representing river bed or grains is performed using the random forest algorithm. The point cloud is classified based only on geometrical features calculated for a local, spherical neighborhood. A multisize neighborhood approach was used together with the feature selection method that is based on correlation analysis. The final classification was performed using a set of features calculated for the neighborhood size of 5, 15, and 20 cm. The achieved classification results have the overall accuracy of 85–95%, depending on the test site. The segmentation is performed using the density-based spatial clustering of applications with noise algorithm in order to cluster the point cloud based on Euclidean distances between points. The performed experiments showed that the proposed method enables us to correctly delineate 67–88% of grains, depending on the test site. However, the resulting point cloud based completeness expressed as Jaccard index is similar for each of the test sites and is approximately 88%. Moreover, the proposed method proved that it is robust to the shadowing effect.**

***Index Terms*—Density-based spatial clustering of applications with noise (DBSCAN), instance segmentation, sediment transport, terrestrial laser scanning (TLS).**

## I. Introduction

GEOMORPHOLOGICAL and hydrological studies of the processes driven by the water flow often require a 3-D model of the river bed topography. To create a topography model, accurate measurements of the terrain surface are required. These measurements are usually performed using remote sensing techniques, including terrestrial laser scanning (TLS). The value and importance of TLS as a monitoring technique in earth science, including river morphology, is investigated by Telling *et al.* [1]. During the laser scanning measurements, a point cloud representing a river bed and its surrounding area is produced. In the further stages, it is processed to obtain a digital elevation model (DEM) that is employed for geomorphological and hydrological analyzes. Examples of such studies include estimation of the volume of transported rock material, evaluation of morphological changes, roughness determination, and description of gravel structure [2]–[4]. Similar investigations were also performed for other measurement techniques, such as mobile laser scanning (MLS) [5], [6]. However, the processing of DEMs does not allow for obtaining complete information about all changes taking place in the river bed. For instance, the thorough investigation of the sediment transport, including the movement path of each grain, requires the studies to be performed directly on the point cloud [7]–[9], [5]. What is more, a number of the applications require segmentation of individual grains from the entire data. The segmented grains can be used for various applications. For instance, they can be directly utilized for grain size estimation or can be a part of a larger workflow aiming at sediment transport monitoring.

The information about grain size distribution can be employed for hydraulic modeling, prediction of boundary shear stress, sediment transport estimation, e.g., [10], [11], and many others [12]. Conventionally, grain size distributions have been obtained by clast measurements and counting grains in a river bed, e.g., [13]. However, these methods are time-consuming and disruptive to the river bed morphology [12], [14]. Therefore, recently, remote sensing techniques are increasing their popularity in this application. Typically, remote sensing data are used for the estimation of grain size distribution by the analysis of statistical properties of images, such as semivariograms of image texture, or autocorrelation of image intensity, e.g., [15], [16]. On the other hand, there are methods that enable delineation of individual grains and measurement of their size directly in the images or orthomosaics, e.g., [17], [14]. However, these methods enable the calculation of only the 2-D size of the grain. As a result, photogrammetry and laser scanning point clouds and their products, such as DEMs are commonly used to derive the surface roughness. This information is then utilized for the estimation of the grain size distribution, e.g., [18]. However, the results are highly influenced by poor sorting of grains [12]. Yet, these challenges may be overcome by segmentation of individual grains directly from the point cloud and measurement of their 3-D size.

The knowledge about sediment transport is important for a number of applications, such as watershed management, ecology, channel restoration [19], engineering design, or flood

Manuscript received September 30, 2021; revised December 13, 2021; accepted December 24, 2021. Date of publication January 11, 2022; date of current version February 3, 2022. This work was supported by the project GATHERS funded by the European Union's Horizon 2020 research and innovation programme under Grant agreement no. 857612. *(Corresponding author: Agata Walicka.)*

Agata Walicka is with the Institute of Geodesy and Geoinformatics, Wrocław University of Environmental and Life Sciences, 50-357 Wrocław, Poland (e-mail: agata.walicka@upwr.edu.pl).

Norbert Pfeifer is with the Department of Geodesy and Geoinformation, Technische Universität Wien, 1040 Vienna, Austria (e-mail: norbert.pfeifer@geo.tuwien.ac.at).

Digital Object Identifier 10.1109/JSTARS.2022.3141892

control [20]. Conventionally, sediment transport is investigated using sediment traps, tracers or is estimated based on sediment equations, e.g., [18], [21], [22]. Recently, also remote sensing techniques are utilized for the measurement of the volume of transported rock material and detection of accumulation and erosion zones by DEM differentiation, e.g., [3]. However, the conventional methods require time-consuming field works and DEMs do not provide information about horizontal changes in the river bed. This information can be filled by a direct analysis of the obtained point cloud. The analysis can provide information about movement parameters, including translation and rotation parameters, for each grain. However, the individual grains need to be segmented to achieve the goal [7].

Since manual segmentation of grains is time-consuming, Wang *et al.* [6] proposed an algorithm that uses DEM as a tool to automatically obtain point clouds representing individual grains. However, the proposed DEM-based approach fails for complicated scenarios, because it does not allow for segmentation of overlapping grains. What is more, the utilization of this method is especially challenging in the presence of extensive shadows caused by large grains obscuring smaller rock fractions. During the creation of the DEM, it is necessary to interpolate over the missing data. The errors and introduced artifacts resulting from the interpolation may substantially affect the final result.

Shadows are caused by the TLS technique and result directly from the measurement methodology, which assumes that scans are performed from a limited number of stations. Hence, while it is possible to partially reduce a shadowing effect, it cannot be completely eliminated. The problem is especially enhanced when the access to the measured object is limited. This is the case, for instance, for many mountain rivers as they have reduced accessibility due to steep slopes and lush vegetation. Therefore, the placement possibilities for the laser scanner are very limited. As a result, in the presence of objects of different sizes (such as grains in the river bed), the space between large objects is shadowed. However, despite these disadvantages, the use of other techniques seems to be less justified. Due to the limited accessibility of some of the river beds, the use of MLS would not be possible as it requires direct access to the measured object. On the other hand, the use of UAVs would be limited due to the safety reasons, impact of the vegetation and changes in the terrain height that would cause complications in the execution of the flight plan. Another alternative technique is terrestrial photogrammetry. However, it provides less accurate description of the grain geometry than laser scanning [8], which also agrees with a statement by Schwendel and Milan [23] that "river reaches with high spatial heterogeneity are challenging for Structure from Motion processing." This may lead to improper identification of corresponding grains during further analyzes. A detailed discussion on alternative measurement techniques and their usefulness for river bed mapping is provided in [8].

To overcome the above limitations, we propose a new method for individual grain extraction that does not require interpolation as it performs an automatic segmentation directly on the point cloud. The development of such a method is complicated because of several reasons. First, the point cloud obtained for the mountain river bed is often of a very irregular density due to shadows. What is more, the number of grains that will be segmented is not known. Second, the sizes of the grains vary greatly. Moreover, the grains are not grouped according to any known pattern. As a result, the use of clustering algorithms is difficult. Third, there are a lot of very small rock fractions and sand in the river bed that cover a substantial part of the grains of a size ca. 10 cm making the grain border ambiguous. Fourth, the grains overlay each other. Thus, it is not possible to reduce the problem to a 2-D or 2.5-D space at any stage of the processing. Finally, the grains have oblong and rounded shapes that locally resemble parts of the river bed.

The developed method is divided into two steps. First, the TLS data are classified into "grains" and "background." The latter consists of river bed, small fractions and vegetation if present in the point cloud. The "grains" class consists of grains with a visible geometry that are larger than 10 cm. The classification uses only geometrical features calculated for each point using a multisize neighborhood approach. In the presented approach a random forest classifier is used and the relevant features are selected based on correlation analysis. Second, the point cloud classified as grains is segmented into individual objects using a clustering method that is based on a spatial proximity. In the presented approach, the "density-based spatial clustering of applications with noise" (DBSCAN) algorithm is applied for this purpose.

The proposed method is validated on a real-world dataset representing a mountain river bed and acquired using TLS. A proof of concept was presented by Walicka *et al.* [24] with lower demands on accuracy and for a small dataset only. Here, we present the full algorithm with several modifications including the simplification of the procedure, results of higher quality, methodology to reduce the number of resulting clusters that do not represent grains, and a more robust classification procedure.

Delineating one small point cloud per particle offers the possibility to monitor sediment transport on the level of individual grains, as suggested in [8]. It requires multiple measurement campaigns and a method to match point clouds of (moved) grains between epochs [7]. The point cloud based approach allows matching the shape, not only under translation but also under rotation, in order to automatically and objectively obtain individual particle movement.

## II. Related Work

The goal of this study is to develop an instance segmentation algorithm to extract individual grains from the TLS point cloud. In contrast to the semantic segmentation methods, instance segmentation algorithms assign each point a label representing an instance of an object of interest instead of the general object category. 3-D instance segmentation algorithms are commonly used to delineate both man-made objects, such as bricks [25], buildings [26], or vehicles [27], [28] and natural objects, such as trees [29]–[31], leaves [32], grains [6], rock blocks [33], or petrified oysters [34]. However, the utilized methods vary depending on the application, complexity of the analyzed scene, and character of the object of interest.

Objects with relatively simple geometry, such as tree stems, building roof facets or columns, usually can be robustly described using a mathematical model. In this case, the most commonly used approaches are object-based methods that rely on robust fitting algorithms used to separate individual objects.

Therefore, during the segmentation, different geometric shapes, such as planes, spheres, or cylinders are fitted to the point cloud. The set of points that are well-matched with the mathematical description of the shape are then labeled as one segment. For example, Rabbani and van den Heuvel [35] used Hough transform to automatically detect cylinder-shaped objects in the laser scanning point cloud. Rusu *et al.* [36] utilized Randomized M-Estimator SAmple Consensus for the planar decomposition of kitchen utilities. Boesch [37] used the Random SAmple Consensus algorithm to detect cylinder-shaped objects representing tree stems. Although model-driven methods allow for quick and accurate segmentation, they cannot be used for irregularly shaped objects, such as grains. In this case, the data-driven approaches are utilized to segment individual objects from the 3-D point clouds.

The data-driven instance segmentation of less complicated scenes can be reduced to a 2-D or 2.5-D problem. For instance, Wang *et al.* [6] used a DEM of the river bed and pouring algorithm to delineate individual grains from the MLS point cloud. Djuricic *et al.* [34] proposed a method for the segmentation of individual oysters from the point cloud registered by TLS. The proposed DEM-based algorithm utilizes negative openness, morphological operations, and connected component analysis to achieve the goal. Miraki *et al.* [38] created a canopy height model (CHM) and tested different 2.5-D algorithms to segment individual trees from the point cloud obtained from UAV photogrammetry. The evaluated algorithms included inverse watershed, local maxima, and region growing. Similarly, Yang *et al.* [39] divided the individual tree extraction method into two steps. First, CHM is created and the watershed algorithm is applied to obtain a coarse position of individual trees. Then, a 3-D analysis of point cloud is performed using principal component analysis (PCA) and K-means clustering algorithm to extract individual trees. On the other hand, Weinmann *et al.* [40] proposed to first classify the 3-D point cloud into points representing trees and background. Then, they project the point cloud representing trees onto a horizontal plane and utilize the mean shift clustering algorithm to delineate individual trees.

Although the methods that exploit 2-D and 2.5-D data proved to be useful and accurate in many applications, in some cases, their use leads to significant omissions in the detected objects. For example, these approaches do not allow for the detection of vehicles parked under the trees [41] or delineation of overlying grains [6]. To overcome these limitations, various methods directly based on 3-D point clouds have been proposed. Most of the commonly used 3-D instance segmentation methods can be categorized as belonging either to the segmentation followed by classification or the classification followed by segmentation approaches. In both types, the instance segmentation is performed in two steps: classification and segmentation but in a different order.

In segmentation followed by classification approaches, the point cloud is first divided into segments. Then, the resulting segments are classified and connected to obtain individual objects of interest [29]. This procedure is also referred to in the literature as object-based point cloud analysis [42]. The classification followed by the segmentation approach is exploited, for example, by Yao *et al.* [43] who proposed to apply adaptive mean shift clustering followed by the modified normalized cut method to generate segments corresponding to different objects. Then, the classification procedure is utilized to assign a semantic label to each segment. As a result, the segments representing individual vehicles are obtained. The proposed method enabled to achieve the completeness of 82%–83% and correctness of 82%–85%, depending on the test site. Zhang *et al.* [44] proposed to divide the individual vehicle extraction method into four steps. First, the point cloud is filtered to exclude the points representing the ground surface. Second, the points with a large above-ground height are detected and excluded. Third, the 3-D CCA method is used to cluster the resulting point cloud. In the last step, the segments are interpreted based on their area, rectangularity, and elongatedness. The achieved correctness varied between 64% and 85%, whereas the completeness varied between 76% and 80%, depending on the test site. Zhang *et al.* [45] divided the method for the detection of individual vehicles into three steps. First, the potential vehicle-occupied areas (PVOA) are identified based on ground points. Then, the nonground points located in PVOA are clustered. Finally, the resulting segments are classified using a dynamic time warping based algorithm. The experiments performed on four datasets showed that the algorithm enables to achieve the correctness of 89%–100%, and completeness of 84%–92%, depending on the test site. Yu *et al.* [46] proposed a method for individual tree extraction that also consists of three steps. First, the ground points are excluded from the point cloud. Second, the nonground points are segmented into clusters representing individual objects using Euclidean distance clustering and voxel-based normalized cut. Third, the segmented point cloud is classified to identify clusters representing trees. Koma *et al.* [32] proposed a two-step method for individual leaves detection. First, the point cloud is segmented using a region growing algorithm, and then the classification is performed to divide the segments into leaves and nonleaves classes. All of the presented methods follow a similar pattern: first, the point cloud is segmented, and then the segment-based classification is performed to obtain individual objects.

In contrast, in classification followed by segmentation approaches, the classification is first performed to eliminate the background and to extract the points representing a whole set of objects of one type. Then, the classified point cloud is segmented into clusters representing individual objects. For example, Xu *et al.* [47] proposed a method for individual tree identification. First, the ground points are removed from the point cloud and the nonground points are classified into tree points using the support vector machine (SVM) algorithm. Then, the individual trees are detected using supervoxel strategy. The reported completeness was equal to 100% and correctness was equal to 94%. Luo *et al.* [29] presented a method for individual tree detection that can be divided into four steps. First, the semantic segmentation is performed using a deep learning approach to extract all points on trees. Second, the tree points are clustered into tree clusters using the Euclidean distance clustering algorithm. Third, the deep learning method is utilized to predict the vectors pointing at the tree tops and the aggregation-based approach is used to detect the tree centers. Finally, the region growing algorithm is used to extract individual trees. The experiments showed that the proposed method allowed to achieve precision between 94% and 96% and recall between 97% and 98%. Kang *et al.* [48] proposed a voxel-based algorithm for extraction of pole-like objects from MLS point cloud that consist of three steps. First, voxels are classified into belonging to the pole-like objects and other using

Markov random field model and shape descriptors. Second, the vertical region growing algorithm is used to segment individual objects. Finally, the objects are classified into specific types of objects (e.g., posts, traffic lights, tree trunks, utility poles, street signs) based on semantic rules. The achieved completeness varied between 75% and 90%, whereas correctness varied between 75% and 99%, depending on the object of interest. Zarea and Mohammadzadeh [49] proposed to utilize both LiDAR point cloud and aerial photographs to delineate individual buildings and trees. First, the terrain points were removed based on digital surface model (DSM) and the remaining points were classified using the SVM algorithm. Then, the individual instances of objects were segmented using connected component analysis and k-means algorithm. The reported completeness was between 82% and 89% and the correctness was between 81% and 97%.

Although the above-mentioned data-driven methods provide high correctness and completeness of the results (64%–100% and 75%–100%, respectively), they cannot be directly applied for instance segmentation of individual grains in the problem at hand. This is because of two reasons. First, the algorithms for instance segmentation of grains and objects that resemble grains (e.g., oysters) are based on DEMs [6], [34]. Therefore, they cannot be used for the delineation of overlapping objects. Second, due to general high adaptation of data-driven 3-D instance segmentation methods to the delineation of one or more types of predefined objects, the above methods need to be modified or tuned to enable segmentation of grains. Therefore, there is a need to develop an instance segmentation algorithm that will enable segmentation of individual, overlying grains and that will thus be based solely on 3-D point cloud analysis.

Therefore, in this article, we present a new classification followed by the segmentation approach for instance segmentation of the individual grains. In contrast to the previous methods for the instance segmentation of the grains, our method is based on 3-D point cloud analysis. This approach enables distinguishing overlying and partially covered grains, which is a very important advantage over a 2.5-D approach, especially in the mountain river environment. We propose to segment the individual grains from the classified point cloud by applying the DBSCAN [50] algorithm.

## III. Materials and Methods

### A. Study Site

As a test site, a 30-m-long section of Łomniczka river was chosen. Łomniczka is a medium-size mountain stream located in the south-west part of Poland. It is characterized by high variability of hydraulic conditions throughout the year. During the autumn, the water level is very low. Thus, it is possible to accurately measure the geometry of the grains of different sizes that are present in the river bed (see Fig. 1), whereas during the spring, high water flows occur that cause intensified displacement of the grains. A detailed description of the test site is provided in [8].

### B. Measurements and Data

The TLS data were collected using the Leica ScanStation C10 laser scanner in 2013. The measurements were performed from two laser scanner positions. The resulting point clouds were coregistered and georeferenced based on three evenly distributed targets. The targets were placed outside of the changing area and their coordinates were measured with a centimeter-level accuracy. The resulting point cloud is characterized by the density of ca. 20 points/cm$^2$ and covers the area of approximately 300 m$^2$.

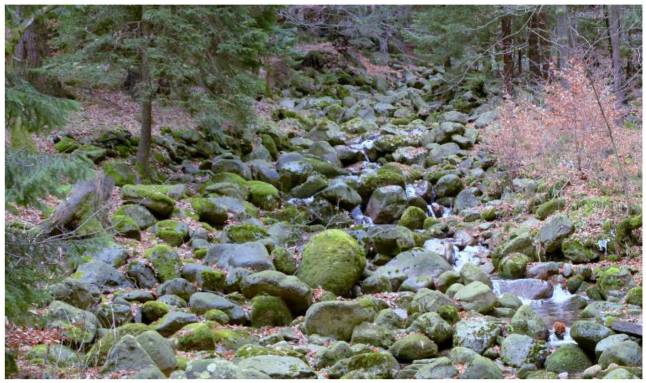

Fig. 1. Test site—Łomniczka river bed.

Based on the collected data, four datasets were created in order to develop and validate the methodology. Datasets consist of manually segmented individual grains and points belonging to the river bed (vegetation, dead wood, coarse and fine sediments—diameter smaller than 10 cm). In each dataset, the grains of sizes greater than 10 cm and visible geometry were manually segmented for the purposes of algorithm development and evaluation. First, the grains with a longer axis larger than 10 cm and visible geometry were identified in the point cloud by a visual analysis and manual measurements performed on the point cloud. Then, the grains were manually segmented from the point cloud. Both steps were performed using Cloud Compare software. The datasets were selected to be spatially disjunctive in order to ensure the independence of the results. The selected datasets are characterized by different spatial distribution of grains, different point cloud density, and were affected by shadowing effect, i.e., not complete visibility of grains and river bed, to varying degrees. Moreover, the ratio of grain points to river bed points fluctuates between the datasets.

The summary of the characteristics of all datasets is presented in Table I. The grain sizes presented in Table I are estimated based on point cloud analysis.

Dataset 1 covers the area of approximately 16 m$^2$ and has the average point density of ca. 32.1 points/cm$^2$. Most of the points in this area belong to the segmented grains (59%). However, the difference between the number of grain and river bed points is not substantial. In total, the dataset consists of 49 manually segmented grains. The grain sizes vary between 12 and 114 cm but the distribution of sizes is not uniform. However, most of the grains are in the size range from 25 to 35 cm. This area is characterized by a relatively small shadowing effect [see Fig. 2(a)]. Thus, the average point cloud density is higher than in the case of the other datasets.

Dataset 2 covers the area of approximately 22 m$^2$ and has the average point density of ca. 11.1 points/cm$^2$. Most of the points in this area belong to the river bed (58%). However, this difference is negligible in the further experiments. In this area, there are 50 grains of sizes from 22 to 177 cm. However, most

TABLE I
SUMMARY OF THE SELECTED DATASET CHARACTERISTICS

| | Area [$m^2$] | Density [points/$cm^2$] | Number of grains | Points belonging to grains [%] | Grain sizes [cm] |
|---|---|---|---|---|---|
| Data set 1 | 16 | 32.1 | 49 | 59 | 12-114 |
| Data set 2 | 22 | 11.1 | 50 | 42 | 22-177 |
| Data set 3 | 29 | 11.3 | 49 | 44 | 18-134 |
| Data set 4 | 26 | 5.2 | 52 | 70 | 15-121 |

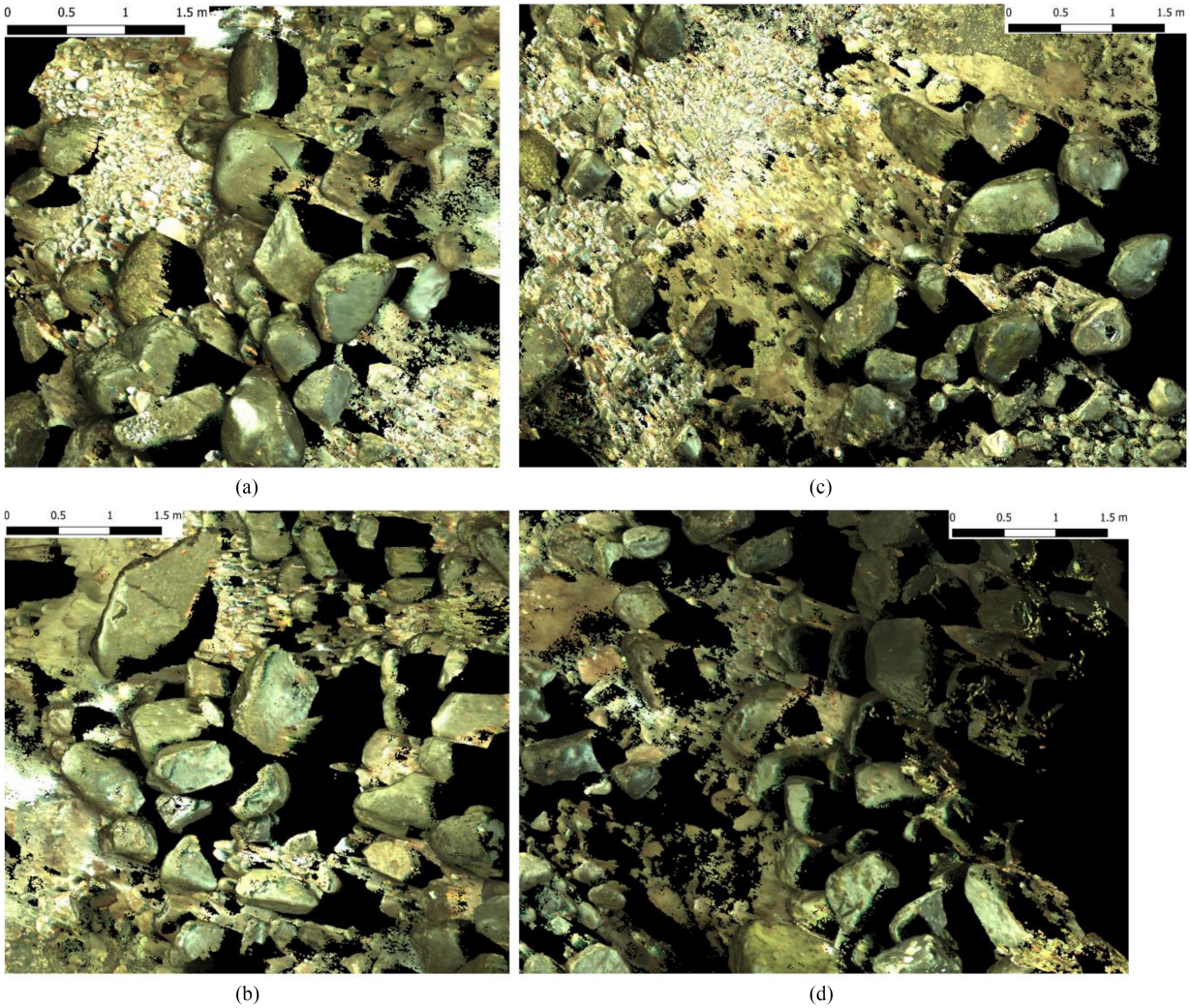


Fig. 2. Point clouds used in the performed experiments. (a) Dataset 1. (b) Dataset 2. (c) Dataset 3. (d) Dataset 4. The black color indicates the lack of data.

of the grains are in the size range from 35 to 50 cm. This area is characterized by a larger shadowing effect than dataset 1 and therefore the density of the point cloud is lower in this case [see Fig. 2(b)].

Dataset 3 covers the area of approximately 29 $m^2$ and has the average point density of ca. 11.3 points/$cm^2$. Most of the points in this area belong to the river bed (56%) but the difference is not substantial. In this area, 49 grains of the size greater than 10 cm were manually segmented for the evaluation purposes. The grain sizes in this area vary from 18 to 134 cm. However, the grain sizes are not equally distributed. Most of the grains are in the size range of 30–40 cm. This area is characterized by a larger shadowing effect and different spatial grain distribution than in the case of previous datasets [see Fig. 2(c)].

Dataset 4 covers the area of approximately 26 $m^2$ and has the average point density of ca. 5.2 points/$cm^2$. Most of the points in this area belong to the grains (70%). Therefore, this dataset should not be used for training purposes. In this area, 52 grains were manually segmented. The grain sizes in this area vary from 15 to 121 cm. However, most of the grains are in the size range of 35–40 cm. This area is relatively highly affected by the shadowing effect [see Fig. 2(d)].

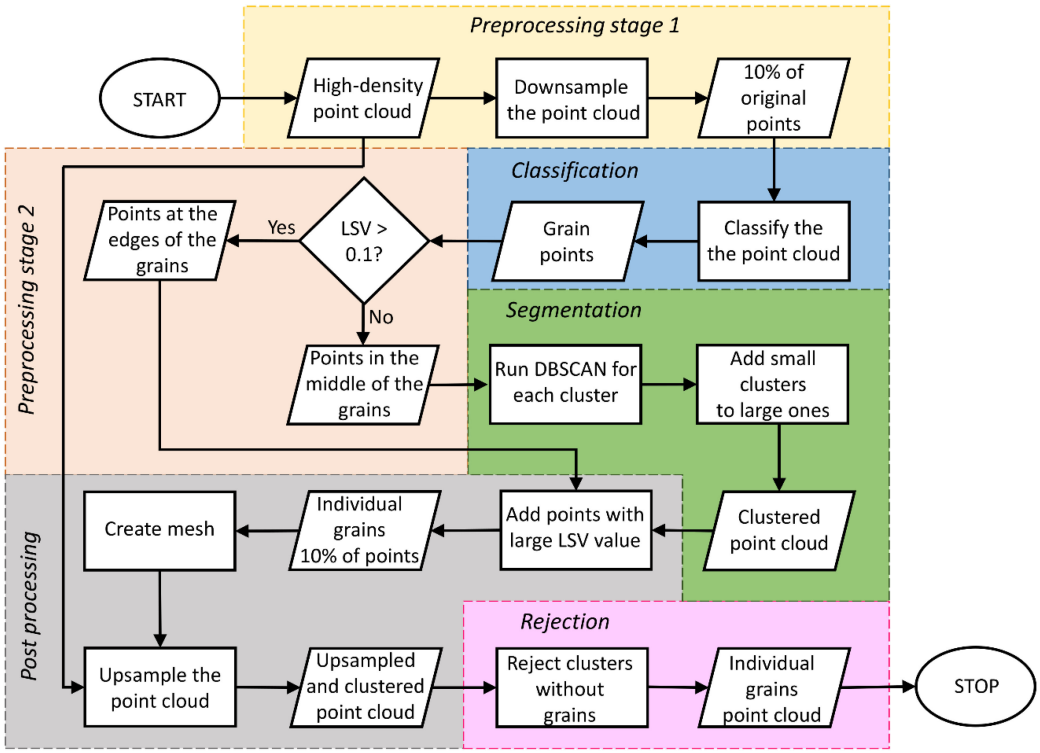


Fig. 3. Flowchart of the proposed method. Each step of the method was marked with a different background color.

### C. Proposed Method

The aim of the proposed method is to segment individual grains from a dense TLS point cloud. To achieve this goal, a procedure based on two main steps, classification and segmentation, is proposed. First, the point cloud is classified to exclude the background points. Then, the segmentation of the point cloud classified as grains is performed in order to delineate individual objects. Additionally, four auxiliary phases including two-step preprocessing, postprocessing, and rejection stages (see Fig. 3) were introduced. The purpose of preprocessing is to speed up calculations and increase the separability of individual grains, whereas the postprocessing and rejection steps aim at handling point clusters that are too small to be of interest, increasing the density of the resulting point cloud, and rejecting incorrectly detected clusters.

The first stage of preprocessing takes place before the classification and is applied to the whole dataset. Since both the classification and segmentation steps are performed in the point-based manner, the aim of this stage is to reduce the number of points that need to be processed. In this stage, 10% of points are randomly selected for further analysis. This enabled substantially reducing computation effort while maintaining sufficient representation of grains' geometry. Since the geometrical features are calculated for the neighborhoods of at least 3 cm, more points located in the immediate proximity to the currently selected points will not provide much new information for the classifier. Moreover, with the current cloud density and the smallest considered neighborhood radius, the neighborhood of most points contains at least 3 points. Therefore, it is possible to calculate the geometrical parameters for most of the points. The second stage of preprocessing is performed after the classification and is applied only to points that were classified as grains. The aim of this stage is to increase the separation between the individual grains in the point cloud. To do this, points characterized by high local surface variance (LSV) (1) were eliminated. The LSV calculated for 5 cm radius was used. The threshold for a high value of the feature was established as 0.1.

The postprocessing stage aims at including all of the previously eliminated points in the resulting clusters to acquire the highest possible point cloud completeness and is performed after the segmentation step. First, the points with high LSV are included. For each point, the distance to the nearest neighbor among all grain points is computed and evaluated. If this distance is smaller than 5 cm, the point is added to the cluster of its neighbor. In the next step, the point cloud is completed with the remaining 90% of points. Points are added to the point cloud based on its proximity to the mesh calculated for each cluster. If the distance between point and mesh does not exceed 1 cm, the point is added to the processed cluster. The mesh is calculated in Cloud Compare software with the maximum triangle size restriction of 10 cm.

The rejection step is the last stage in the procedure and aims to eliminate the clusters that represent the river bed instead of the grains. If smooth sandy regions occur, they may wrongly be identified as grains. However, these clusters are flat (see examples in dataset 2). Therefore, the rejection is performed based on the third eigenvalue of the covariance matrix of the coordinates of points belonging to the local neighborhood. The threshold value for this feature was estimated to $10^{-4}$ (using dataset 2).

*1) Point Cloud Classification:* The aim of the point cloud classification was to extract the background points from the

points belonging to the objects of interest. Thus, a binary classification was performed into sets of points representing background (river bed, small rock fractions, vegetation, etc.) and grains with a visible geometry that are larger than 10 cm. To classify the point cloud a random forest algorithm in a point-based approach was used. The classifier was chosen based on several reasons. First, random forest is an ensemble shallow learning algorithm that is based on the creation of multitude decision trees. Therefore, it often provides better classification results than other shallow learning algorithms (e.g., SVM) [51]. Second, due to its tree-based character, it is computationally effective, rarely overfits, and is robust to data of different ranges. On the other hand, in comparison to deep learning techniques, it enables training based on relatively small data samples. The point cloud was classified using only geometrical parameters (1)–(9) calculated for a spherical neighborhood

$$\text{Local surface variance } C_\lambda = \frac{\lambda_3}{\sum_{i=1}^{3} \lambda_i} \tag{1}$$

$$\text{Sphericity } S_\lambda = \frac{\lambda_3}{\lambda_1} \tag{2}$$

$$\text{Linearity } L_\lambda = \frac{\lambda_1 - \lambda_2}{\lambda_1} \tag{3}$$

$$\text{Planarity } P_\lambda = \frac{\lambda_2 - \lambda_3}{\lambda_1} \tag{4}$$

$$\text{Anisotropy } A_\lambda = \frac{\lambda_1 - \lambda_3}{\lambda_1} \tag{5}$$

$$\text{Omnivariance } O_\lambda = \sqrt[3]{\lambda_1 \cdot \lambda_2 \cdot \lambda_3} \tag{6}$$

$$\text{Eigentropy } E_\lambda = \sum_{i=1}^{3} \lambda_i \cdot \ln(\lambda_i) \tag{7}$$

$$\text{Sum of eigenvalues } \Sigma_\lambda = \sum_{i=1}^{3} \lambda_i \tag{8}$$

$$\text{Verticality } V = \left| \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}^T e_3 \right| \tag{9}$$

where $\lambda_i,\ i = 1,\ 2,\ 3$, $\lambda_1 > \lambda_2 > \lambda_3$ are the eigenvalues of the covariance matrix of the coordinates of points belonging to the local neighborhood, $e_3$ is the eigenvector corresponding to the smallest eigenvalue.

In order to achieve better classification results, a multisize neighborhood approach was used [52]. Therefore, the above features were calculated for 15 neighborhood sizes varying from 3 to 70 cm radius. However, due to the large number of neighborhood sizes, the number of features increased from 9 to 135. Although the features calculated for varying neighborhood sizes provide additional information to the classifier, the classification based on a very large feature set may lead to overfitting and to a decrease in the quality of classification results. Therefore, a proper feature selection strategy should be introduced to assure high overall accuracy (OA) of classification results for both training and testing datasets.

Feature selection strategies aim at finding the smallest possible set of relevant and meaningful features. As a result, they can lead to the reduction of overfitting probability, and improvement of accuracy and computational efficiency of the classification [53], [54]. The approaches for feature selection can be divided into filter-based, wrapper-based, and embedded methods. All of them are commonly used for point cloud classification problems [55]–[57]. The detailed overview of the feature selection methods is provided by Saeys *et al.* [54].

Since all of the feature selection strategies have their advantages and limitations, the technique used should be adapted to the problem under consideration. Therefore, to select a proper technique for this investigation, we defined the following goals. A selected strategy should satisfy the following conditions:

1) ensure lack of overfitting;
2) enable the proposed method to work with small training dataset;
3) reduce the dependence on the classifier;
4) allow to retrieve the additional information about the classification problem;
5) minimize computing time by limiting the number of neighborhood sizes for feature calculation.

Therefore, we decided to combine wrapper-based and filter-based strategies to achieve the defined goals by exploiting the advantages of both of them.

In the first step, the classification accuracy is evaluated by training and testing the classifier using all features (1)–(9) for each neighborhood size separately. Moreover, for the comparison purposes, also the accuracy of the classification using all features calculated for all neighborhood sizes is assessed. This step enables us to investigate the influence of the neighborhood size on the classification accuracy, and, as a result, select a smaller subset of neighborhood sizes and to justify the usefulness of feature selection in this particular classification problem. What is more, it is possible to draw the conclusions about the relationship between best performing neighborhood radius and minimal grain size. As a result of the analysis, the best performing neighborhood size is selected as a base one and the neighborhood size can be restricted if a substantial drop of the accuracy can be observed.

In the second step, the selected neighborhood sizes are analyzed individually using a filter-based approach to determine the smallest possible subset of meaningful features for each neighborhood. This step aims at identifying correlated features by analysis of Spearman correlation rank [58]. As a result, several sets of highly correlated features are created. The features are considered highly correlated when the Spearman correlation rank between them exceeds 0.9. Then, for each set, the feature correlated the most with the reference classification results is selected based on Kendall correlation rank. If the Kendall correlation is the same for several features, the feature selected for the base neighborhood size is selected. As a result, for each of the selected neighborhood sizes, a set of uncorrelated features is determined. In the last step, the Spearman correlation ranks between the selected features for all neighborhood sizes are analyzed. The features calculated for the best performing neighborhood size are considered the base ones and the feature set is expanded by features calculated for larger neighborhoods

when the Spearman correlation rank is smaller than 0.3. The proposed feature selection strategy enables to ensure the lack of overfitting due to the application of filter-based approaches and to reduce the computation time over the wrapper-based methods while allowing to retrieve additional information about the classification problem. Moreover, the dependence on the classifier is reduced only to the base neighborhood size selection and limitation of the considered neighborhood sizes. As a result, both, number features and neighborhood sizes were reduced.

After the selection of appropriate features, the hyperparameters (number of trees, maximum depth of the tree, the minimum number of samples required to split the internal node, the minimum number of samples required to be at a leaf node) tuning was performed. Finally, to minimize the salt and pepper effect and to achieve more concise results, a majority filter with a spherical neighborhood of 10 cm radius was applied to the dataset.

The classifier was trained using dataset 1 (training dataset), whereas the hyperparameters fitting, neighborhood size selection, and feature selection were performed based on dataset 2 (validation dataset). The final classification results were evaluated based on datasets 3 and 4 (testing datasets).

*2) Point Cloud Segmentation:* Point cloud segmentation was performed using DBSCAN, which is a density-based clustering algorithm. The algorithm was applied directly on the coordinates of points classified as grains. However, to ensure the sufficient distance between the individual grains, the points with a high LSV were excluded from the classified point cloud.

The DBSCAN algorithm needs two parameters to be specified: the neighborhood radius and the number of points placed within the defined neighborhood. However, regardless of the selected parameters, during the procedure, some small and not meaningful clusters are produced. Because of the classification inaccuracy, these clusters may belong both to the river bed and to the grains. Therefore, it is beneficial to include some of them in the final result to ensure the highest possible completeness of the resulting point cloud for the real grain clusters. To do this, a nearest neighbor approach was employed. Points belonging to the clusters smaller than ca. 300 points were attached to the large ones if their distance to the nearest neighbor in the large cluster was smaller than 5 cm. Dataset 2 was used both to develop the segmentation procedure and to adjust its parameters.

*3) Validation of the Proposed Method:* The validation of the proposed method was performed in two stages. First, the accuracy of the classification was evaluated. The validation was performed based on datasets 3 and 4, which were neither used for the classifier training nor for the parameter adjustment. However, for the comparison and to exclude the possibility of the overfitting occurrence, also the results achieved on training (dataset 1) and validation (dataset 2) data were analyzed. Most of the datasets are characterized by a balanced number of points representing both classes. Therefore, for the evaluation purposes, the OA (10) was used. However, to provide more detailed information about classification accuracy, also precision (11), recall (12), and $F1$ score (13) statistic values were provided. Additionally, the visual analysis of the resulting classification was performed

$$\mathrm{OA} = \frac{\mathrm{TP}+\mathrm{TN}}{\mathrm{TP}+\mathrm{TN}+\mathrm{FP}+\mathrm{FN}} \tag{10}$$

TABLE II
USE OF INDIVIDUAL DATASETS DIVIDED INTO THE STAGES OF THE ALGORITHM PROCESSING

| Data set | Purpose | |
|---|---|---|
| | Classification | Segmentation |
| 1 | Training | Testing |
| 2 | Validation | Training |
| 3 | Testing | |
| 4 | Testing | |

$$\mathrm{precision} = \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FP}} \tag{11}$$

$$\mathrm{recall} = \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}} \tag{12}$$

$$F1 = \frac{2\mathrm{TP}}{2\mathrm{TP}+\mathrm{FP}+\mathrm{FN}} \tag{13}$$

where $\mathrm{TP}$, $\mathrm{TN}$, $\mathrm{FP}$, $\mathrm{FN}$ are the elements of a confusion matrix standing for true positive, true negative, false positive, and false negative, respectively.

In the second step, the accuracy assessment of the whole algorithm was performed. The analysis was mostly based on datasets 3 and 4. Also, the results achieved for dataset 1 were compared as it enabled us to evaluate the accuracy of the segmentation in the condition of the best possible classification results. Moreover, it enabled us to assess the influence of the classification procedure on the results of the whole method. The evaluation of the method was performed based on three metrics: number of correctly recognized grains, number of segmented clusters that are not grains, and completeness of the resulting point cloud. Based on two first metrics, the completeness and correctness, in the terms of detected objects, were calculated based on (12) and (11), respectively. However, in this case $\mathrm{TP}$ means number of correctly detected grains, $\mathrm{FP}$ means number of resulting clusters that are not grains (additional clusters), and $\mathrm{FN}$ means incorrectly detected grains. The completeness of the resulting point cloud was assessed using the Jaccard index [59] (14) that is also commonly referred in the literature as intersection over union, and two other metrics. First one ($K_1$) describes a percent of points in the reference point cloud that are not present in the resulting point cloud and describes points belonging, for example, to the omitted grains (15). Second one ($K_2$) describes the percent of points that are present in the resulting point cloud but are not included in the reference one (16). This metric informs, for example, about clusters incorrectly detected as grains. Consequently, the better algorithm results, the higher value of the Jaccard index and the lower values of $K_1$ and $K_2$ metrics. Additionally, for a better understanding of the intermediate results, the following statistics were provided for each processing step: number of clusters, average size of the cluster, Jaccard index, $K_1$, and $K_2$. Moreover, for the first stage of processing, the number of small clusters and their average size, and the number of large clusters and their average size were provided.

A summary of the ways in which individual datasets are used in these investigations is presented in Table II

$$J = \frac{A \cap B}{A \cup B} \tag{14}$$

TABLE III
CLASSIFICATION RESULTS FOR A SINGLE-NEIGHBORHOOD APPROACH FOR ALL NEIGHBORHOOD SIZES

| Neighborhood radius | Overall accuracy | | |
|---|---|---|---|
| | Training set | Validation set | Difference |
| 0.03 | 0.988 | 0.810 | 0.178 |
| 0.05 | 0.991 | 0.834 | 0.157 |
| 0.10 | 0.993 | 0.807 | 0.186 |
| 0.15 | 0.995 | 0.772 | 0.223 |
| 0.20 | 0.997 | 0.733 | 0.264 |
| 0.25 | 0.998 | 0.708 | 0.290 |
| 0.30 | 0.998 | 0.694 | 0.304 |
| 0.35 | 0.999 | 0.677 | 0.322 |
| 0.40 | 0.999 | 0.676 | 0.323 |
| 0.45 | 0.999 | 0.673 | 0.326 |
| 0.50 | 0.999 | 0.653 | 0.346 |
| 0.55 | 0.999 | 0.610 | 0.389 |
| 0.60 | 0.999 | 0.617 | 0.382 |
| 0.65 | 0.999 | 0.646 | 0.353 |
| 0.70 | 0.999 | 0.628 | 0.371 |
| All | 1.000 | 0.833 | 0.167 |

$$K_1 = \frac{A - B}{A \cup B} \tag{15}$$

$$K_2 = \frac{B - A}{A \cup B}. \tag{16}$$

In the definition of Jaccard's index ($J$), $K_1$ , and $K_2$, $A$ is the reference dataset and $B$ is the final result after segmentation, postprocessing, and rejection. Both sets $A$ and $B$ represent only those points that belong to the grains.

The algorithm's performance was also evaluated in regards to the grain sizes. To do this, the length of the longer axis of each grain was estimated based on the reference dataset using PCA [60]. As a result, the influence of shadows is included in the grain size. However, for method development, the size of the scanned part of the grain is of most importance. Thereafter, the number of correctly segmented and the number of connected grains was evaluated in the defined size ranges.

## IV. RESULTS AND DISCUSSION

### A. Point Cloud Classification

The classification results for all features calculated for different neighborhood sizes are presented in Table III. To select the base neighborhood size, 15 different neighborhood radii were tested. The evaluated sizes vary from 5 to 70 cm, in steps of 5 cm. Tested sizes were selected taking into consideration the properties of objects of interest. Since the algorithm should enable proper classification of grains with a diameter equal to at least 10 cm, the smallest neighborhood radius was selected as half of this value. Moreover, a large neighborhood radius would prevent the proper classification of smaller grains. Additionally, the neighborhood size of 3 cm was included in the experiments to ensure that the classification accuracy will not increase when further reducing the size of the neighborhood. The best OA was obtained for 5 cm radius, whereas the worst accuracy was reported for the 55 cm radius. For the 5 cm neighborhood size also the smallest difference between OA for training and validation dataset was reported. A constant downward trend of the OA on the validation dataset with increasing neighborhood size can be observed. Moreover, the classification results dropped for the neighborhood size smaller than 5 cm. This was caused by an increasing number of cases in which the features could not be calculated due to insufficient number of points in the neighborhood. The classification accuracy achieved for all neighborhood sizes was slightly worse than the accuracy for the 5 cm neighborhood. However, the classification accuracy on the training dataset almost reached 1, which indicates a high probability of overfitting. Therefore, in the further investigations, the neighborhood with a radius of 5 cm was treated as the base. Moreover, the performed analysis resulted in the restriction in the neighborhood sizes to a maximum of 25 cm due to the substantial drop in the classification accuracy reported for the validation dataset (less than 70% OA) when using larger neighborhoods.

In the second step, the feature selection within the 5 cm radius was performed. The calculated Spearman correlation rank values are presented in Fig. 4. The analysis of the results showed that two groups of mutually correlated features can be constructed. Group 1 includes sphericity, anisotropy, omnivariance, eigentropy, and LSV, whereas group 2 includes linearity and planarity. In group 1, LSV feature was the most correlated with the reference result, thus it was selected for the further processing. In group 2, the planarity feature was selected. Additionally, the features that were not highly correlated with any other feature were added to the feature set. As a result, the feature set consisted of LSV, planarity, sum of eigenvalues, and verticality calculated for 5 cm neighborhood radius.

In the next step, the correlation of each of the selected features for different neighborhood sizes was calculated and analyzed (see Fig. 5). The analysis was performed taking into account the conclusions from step 1 regarding the maximum size of the neighborhood. Consequently, the resulting feature set included the following features: LSV, planarity, sum of eigenvalues and verticality calculated for 5 cm radius neighborhood, LSV and planarity calculated for 20 cm radius neighborhood, and sum of eigenvalues calculated for 15 cm radius neighborhood.

In the last step, the random forest hyperparameters were investigated. The following parameters were tested for the specified values.

1) Number of trees—1, 3, 5, 10, 50, 80, 100, 300.
2) Maximum depth of the tree—5, 8, 15, 25, 30, not defined.
3) Minimum number of samples required to split internal node—2, 5, 10, 15.
4) Minimum number of samples required to be at a leaf node—1, 2, 5, 10.

All the parameter combinations were tested in order to achieve the highest OA for the validation dataset. As a result of the experiments, the following hyperparameters were selected: number of trees equal to 300, maximum depth of the tree of 8, minimum number of samples required to split internal node equal to 2, and minimum number of samples required to be at a leaf node equal to 10.

After the features and classification parameters were selected, the accuracy on training (dataset 1), validation (dataset 2), and two testing (datasets 3 and 4) datasets was evaluated. The achieved accuracy was equal to 91%, 85%, 86%, and 81% on dataset 1, 2, 3, and 4, respectively. Although the achieved OA

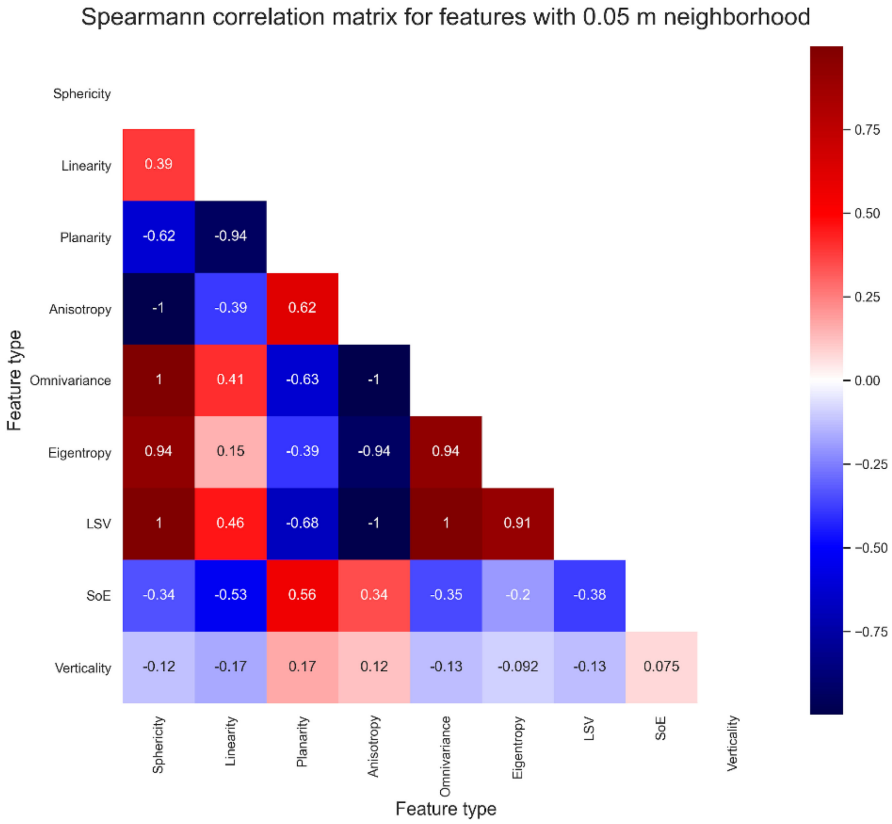


Fig. 4. Spearman correlation matrix for features with 5 cm neighborhood.

TABLE IV
QUANTITATIVE ANALYSIS OF THE CLASSIFICATION RESULTS

| | | Grains | | | | | Background | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | precision | recall | f1-score | TP [%] | FN [%] | precision | recall | f1-score | TP [%] | FN [%] |
| Before majority filter | Data set 1 | 0.92 | 0.92 | 0.92 | 54 | 5 | 0.88 | 0.89 | 0.89 | 36 | 5 |
| | Data set 2 | 0.83 | 0.93 | 0.88 | 54 | 4 | 0.89 | 0.74 | 0.80 | 31 | 11 |
| | Data set 3 | 0.84 | 0.84 | 0.84 | 37 | 7 | 0.88 | 0.88 | 0.88 | 49 | 7 |
| | Data set 4 | 0.84 | 0.89 | 0.87 | 63 | 7 | 0.71 | 0.62 | 0.66 | 19 | 11 |
| After majority filter | Data set 1 | 0.95 | 0.96 | 0.95 | 56 | 6 | 0.94 | 0.93 | 0.93 | 38 | 3 |
| | Data set 2 | 0.89 | 0.96 | 0.91 | 57 | 1 | 0.96 | 0.77 | 0.86 | 32 | 10 |
| | Data set 3 | 0.90 | 0.90 | 0.90 | 40 | 4 | 0.92 | 0.93 | 0.92 | 52 | 4 |
| | Data set 4 | 0.85 | 0.96 | 0.90 | 67 | 3 | 0.85 | 0.60 | 0.71 | 18 | 12 |

was high, the results contained a lot of salt and pepper noise (see Fig. 6). Therefore, in order to solve this problem, the majority filter was applied to the results, which enabled us to achieve 95%, 89%, 91%, and 85% OA on dataset 1, 2, 3, and 4, respectively. Thus, the achieved accuracy significantly increased and the results are much more concise (see Fig. 6). These conclusions were also confirmed by the analysis of precision, recall, and $F1$-score for each class (see Table IV). Although the classification results are very accurate and concise, there are still some errors in the classified point cloud. The most common error is classification of flat parts of the river bed as grains (example: Fig. 6—green ellipse). This is caused by large, locally flat grains present in the training dataset and, at the same time, lack of flat parts of the river bed. Moreover, the classification of small grains (10–25 cm) is challenging and many errors occur in this case. This is caused by a limited number of small grains whose geometry is sufficiently well described to distinguish them from the river bed. They are also very commonly shadowed by larger fractions. Therefore, it is highly probable to confuse them with the river bed. Moreover, the number of points belonging to a small grain is substantially smaller than the number of points belonging to the large one. Thus, this effect is not visible in the case of very large grains that also occur very rarely in the datasets. Another type of error that is visible mostly in datasets 3 and 4 is classification of part of the points belonging to the grains as a river bed (example: Fig. 6—yellow ellipse). This happens mostly for the concave parts of the grains. Although the classification errors still occur in all of the datasets, they are not major and some of them will be eliminated or reduced in the postprocessing or in the rejection stage.

### B. *Point Cloud Segmentation*

To segment the individual grains, a clustering procedure based on the DBSCAN algorithm was performed. The segmentation was based on Euclidean distance between points and was applied to the classified data. In the next step, the postprocessing of data

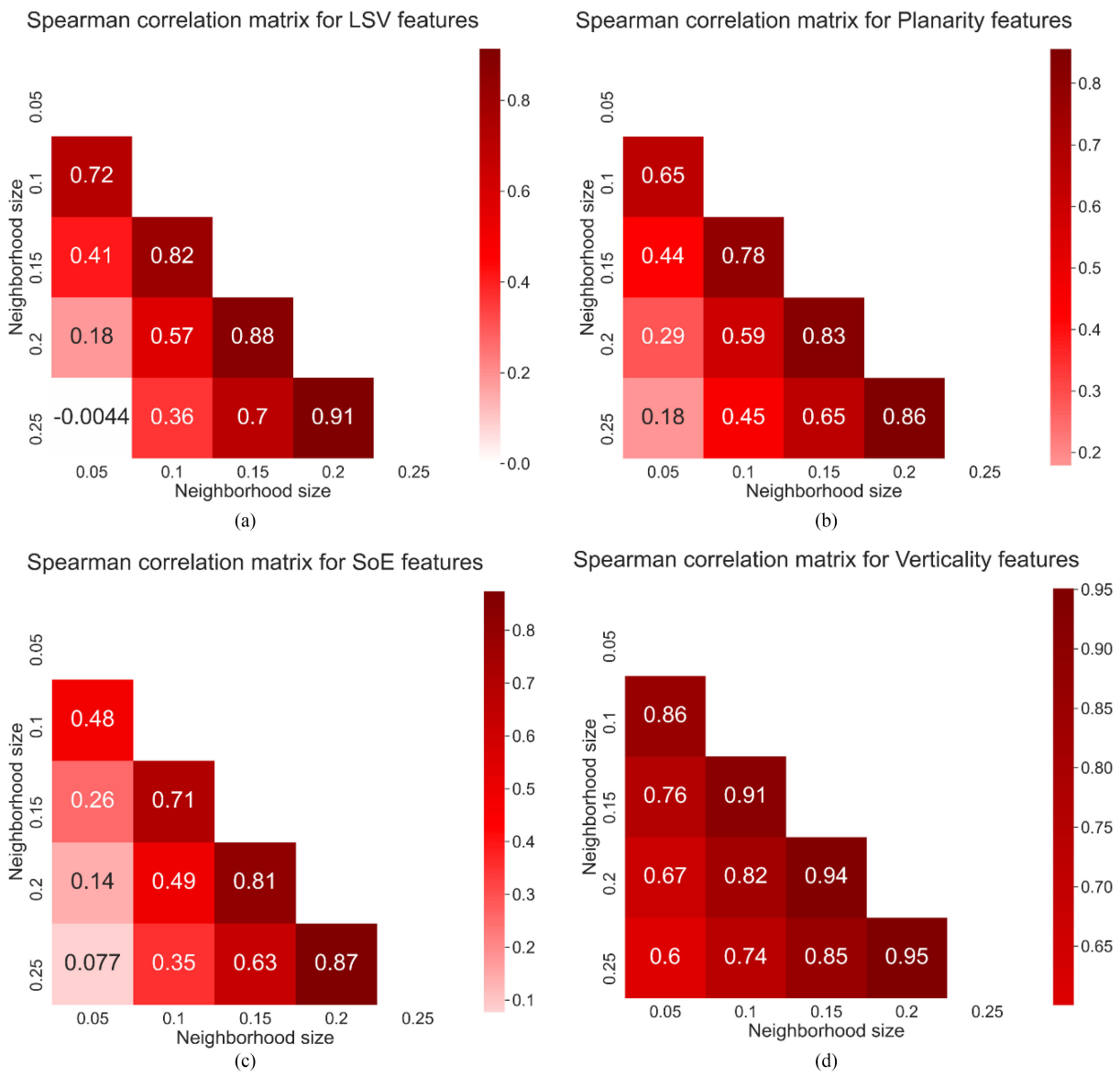


Fig. 5. Correlation between features calculated for different neighborhood sizes. (a) LSV. (b) Planarity. (c) Sum of eigenvalues. (d) Verticality.

and rejection of clusters were executed. The statistics representing the results achieved for each processing step are presented in Table V. The postprocessing procedure enabled reduction of number of clusters by 84%–91%, depending on a test site. The number of clusters rejected in the last processing step varies between 3 and 9, depending on the test site. For two out of four datasets, this procedure enabled to increase the value of Jaccard index. For all datasets, this step enabled to increase the mean number of points in cluster.

The results of the whole procedure executed on each dataset are presented in Figs. 7 and 8. The reference data for instance segmentation are visualized in the Appendix. First, the results were visually examined. The analysis revealed that most of the grains are correctly segmented. However, in some cases, a few clusters exist that contain more than one grain (usually two, rarely three grains). This was mainly caused by too small distance between two grains. However, in some cases, classification errors also led to the formation of such clusters because of incorrect classification of the river bed between grains as grain class. Moreover, for a small number of clusters, the result is slightly spatially extended by points belonging to the river bed or reduced by points belonging to the grain but incorrectly classified as river bed. In the very rare situations, one grain can be segmented as two. This might be caused by smaller fractions covering the grain surface.

The quantitative analysis of the results (see Table VI) confirmed that the proposed methodology enabled correct segmentation of most of the individual grains. The highest accuracy (88% completeness and 77% correctness) was achieved for dataset 2, which was used for segmentation methodology development. Additionally, very high accuracy (76% completeness and 93% correctness) was reached by dataset 1 that was used

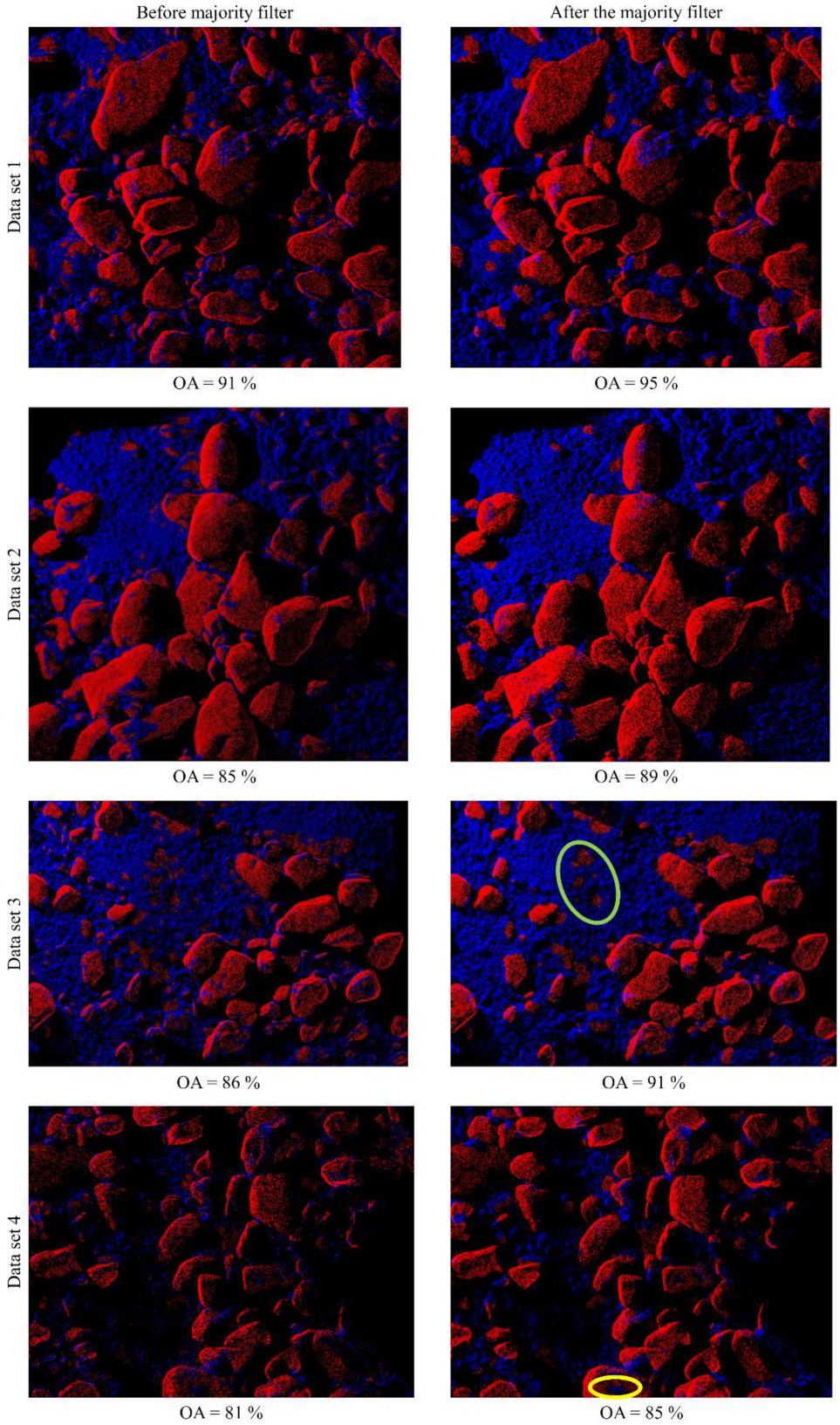


Fig. 6. Classification results on training (dataset 1), validation (dataset 2), and testing (datasets 3 and 4) datasets.

TABLE V
SUMMARY OF THE CHARACTERISTICS OF THE RESULTS OF EACH STEP OF DATA PROCESSING

| | # clusters | # large clusters | # small clusters | Average size of a cluster large | small | J | $K_1$ | $K_2$ |
|---|---|---|---|---|---|---|---|---|
| | | | | Data set 1 | | | | |
| Clustering | 269 | 52 | 217 | 5694 | 16 | - | - | - |
| Individual grains (10% of points) | 52 | - | - | 5824 | | 0.10 | 0.90 | 0.00 |
| Upsampled point cloud | 52 | - | - | 58749 | | 0.91 | 0.04 | 0.05 |
| Rejected non-grain clusters | 44 | - | - | 66288 | | 0.91 | 0.06 | 0.03 |
| | | | | Data set 2 | | | | |
| Clustering | 413 | 66 | 347 | 2292 | 20 | - | - | - |
| Individual grains (10% of points) | 66 | - | - | 2355 | | 0.10 | 0.89 | 0.01 |
| Upsampled point cloud | 66 | - | - | 23613 | | 0.83 | 0.04 | 0.13 |
| Rejected non-grain clusters | 59 | - | - | 25442 | | 0.86 | 0.04 | 0.10 |
| | | | | Data set 3 | | | | |
| Clustering | 278 | 49 | 229 | 2796 | 26 | - | - | - |
| Individual grains (10% of points) | 49 | - | - | 2829 | | 0.09 | 0.90 | 0.01 |
| Upsampled point cloud | 49 | - | - | 28597 | | 0.83 | 0.10 | 0.07 |
| Rejected non-grain clusters | 40 | - | - | 33040 | | 0.85 | 0.12 | 0.03 |
| | | | | Data set 4 | | | | |
| Clustering | 575 | 56 | 519 | 1713 | 18 | - | - | - |
| Individual grains (10% of points) | 56 | - | - | 1764 | | 0.09 | 0.90 | 0.01 |
| Upsampled point cloud | 56 | - | - | 17520 | | 0.83 | 0.08 | 0.09 |
| Rejected non-grain clusters | 53 | - | - | 18204 | | 0.83 | 0.08 | 0.09 |

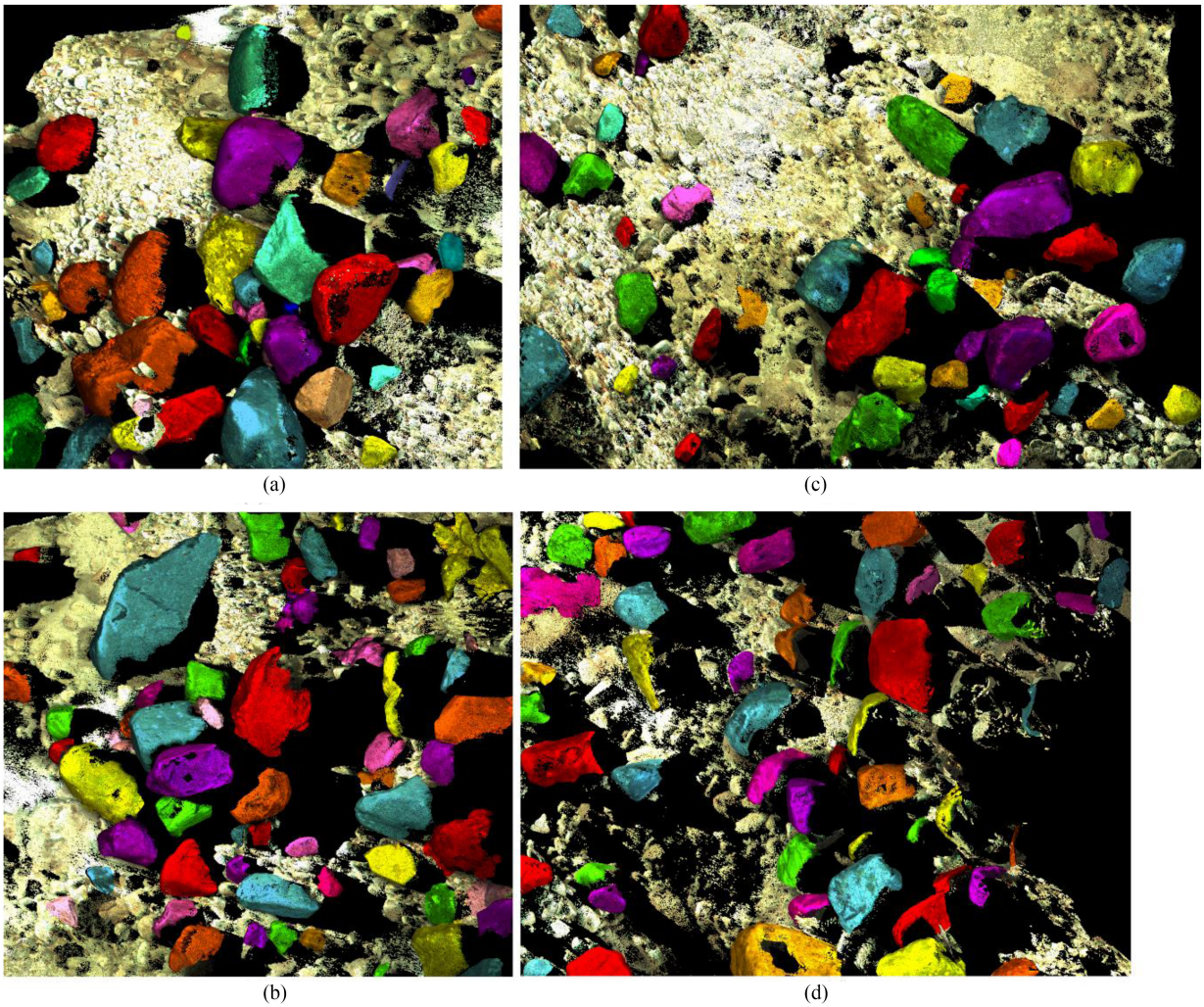


Fig. 7. Results of individual grain segmentation—top view. (a) Dataset 1. (b) Dataset 2. (c) Dataset 3. (d) Dataset 4. Each segmented grain is presented in a different color. The background (river bed) is shown with the original color of the colorized point cloud.

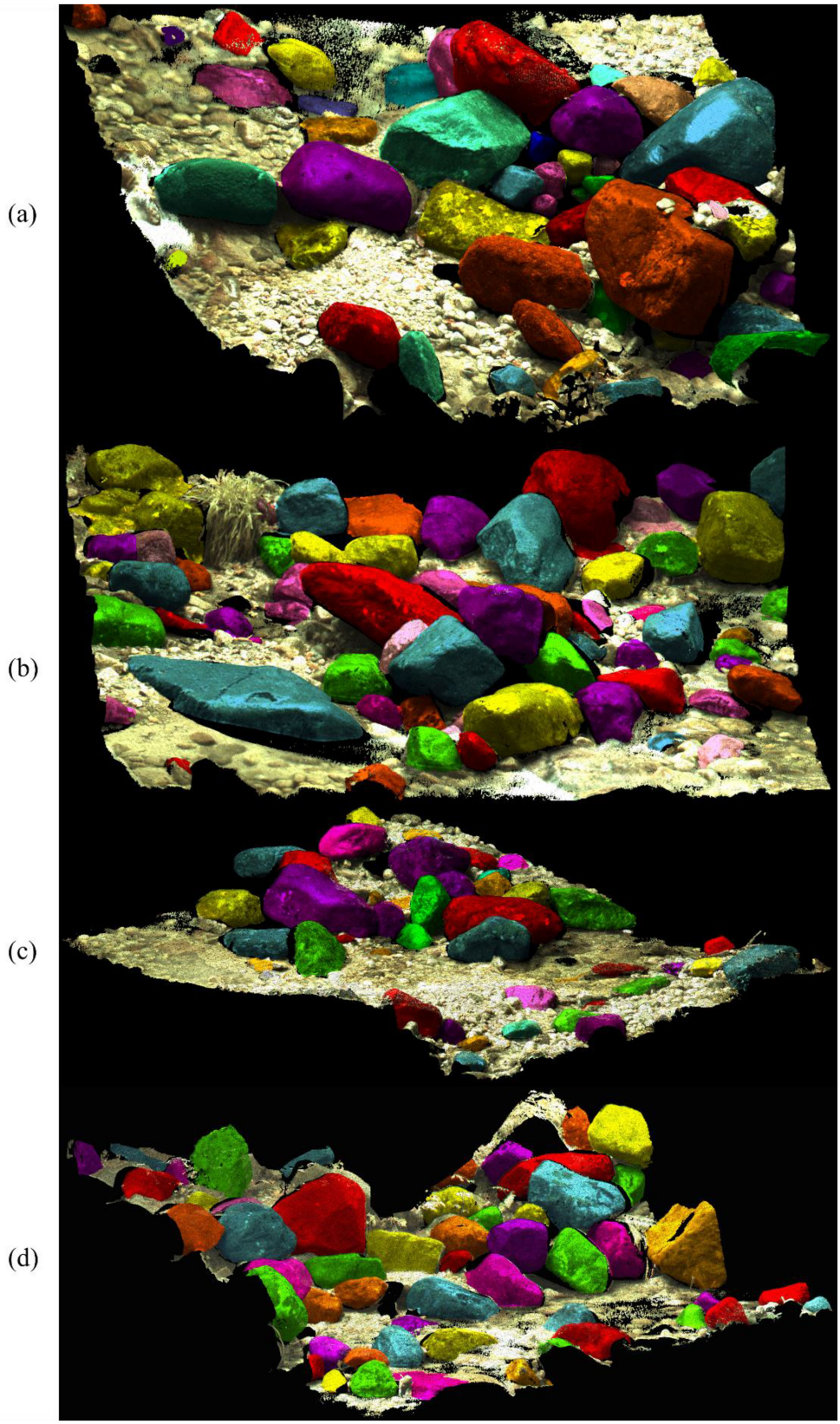


Fig. 8. Results of individual grain segmentation—side view. (a) Dataset 1. (b) Dataset 2. (c) Dataset 3. (d) Dataset 4. Each segmented grain is presented in a different color. The background (river bed) is shown with the original color of the colorized point cloud.

TABLE VI
QUANTITATIVE ANALYSIS OF THE RESULTS AFTER SEGMENTATION, POSTPROCESSING, AND REJECTION

| Data set | Point-based statistics | | | | Grain-based statistics | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | # reference grain points | J | $K_1$ | $K_2$ | # reference grains | Recognized correctly | Completeness | Correctness | Connected | Omitted | Additional clusters |
| 1 | 3053026 | 0.91 | 0.06 | 0.03 | 49 | 37 | 0.76 | 0.93 | 8 | 4 | 3 |
| 2 | 1416561 | 0.86 | 0.04 | 0.10 | 50 | 44 | 0.88 | 0.77 | 5 | 1 | 13 |
| 3 | 1452216 | 0.85 | 0.12 | 0.03 | 49 | 33 | 0.67 | 0.87 | 6 | 10 | 5 |
| 4 | 961491 | 0.83 | 0.08 | 0.09 | 52 | 40 | 0.77 | 0.78 | 4 | 8 | 11 |
| Total | 6883294 | 0.88 | 0.07 | 0.05 | 200 | 154 | 0.77 | 0.83 | 23 | 23 | 32 |

TABLE VII
RESULTS OF THE VISUAL ANALYSIS OF ADDITIONAL CLUSTERS

| Data set | Additional clusters | | |
|---|---|---|---|
| | A grain | Not a grain | Total |
| 1 | 0 | 3 | 3 |
| 2 | 5 | 8 | 13 |
| 3 | 1 | 4 | 5 |
| 4 | 4 | 7 | 11 |

for the training of the classifier. Moreover, for this dataset, the smallest number of clusters without grains was produced. Since dataset 1 was used as a training dataset for the classifier and, as a result, was characterized by very high classification accuracy (OA of 95%), it can be concluded that the classification step is crucial for the accuracy of the whole method. The completeness of the method for datasets 3 and 4 was high and reached 67% and 77% of correctly recognized grains, respectively, whereas the achieved correctness was equal to 87% and 78% for datasets 3 and 4, respectively. The slightly lower completeness for dataset 3 was caused by the occurrence of many small grains ($< 25$ cm) that were incorrectly assigned to the river bed class during the classification step. The instance segmentation algorithms presented in Section II showed the results with both completeness and correctness between 64% and 100%. This fits well with the instance segmentation results presented here.

The rejection step enabled removing most of the clusters that did not represent grains. However, some of the clusters that were not included in the reference data still remain in the results (see Table V). Several of these clusters represent river bed, thus their presence in the results can be classified as an error. However, an additional, thorough analysis of the point cloud revealed that, in a few cases, the additional clusters represent grains that were placed at the edge of the datasets. As a result, they were partially cut. This led to the identification errors during the manual analysis, since the examination of the surrounding point clouds was required to reveal that a certain part of the point cloud represents a grain. Moreover, some of the additional clusters represent small grains whose geometry was partially covered by sand. As a result of this and because of very poor quality of RGB data, these grains were also missed during the manual analysis. The results of thorough analysis of the source of additional clusters are presented in Table VII.

The procedure of joining small clusters and points with high LSV together with the upsampling algorithm enabled obtaining high completeness of the resulting point cloud (see Table VI). The experiments revealed that the achieved point cloud based completeness is similar for all datasets and varies from 83% to 91% (Table VI, column J). These values do not directly correspond to the percentage of correctly segmented grains. This observation additionally confirms that in dataset 3, most of the omitted grains belong to the smallest of the target rock fractions. Thus, the loss in the point cloud based completeness is minor.

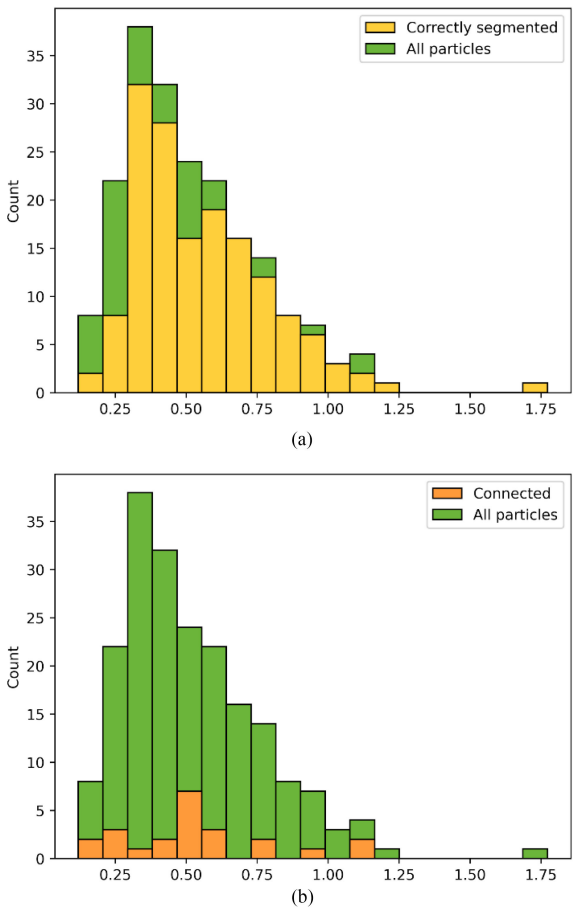


Fig. 9. Histogram of the grain distribution in regards to the length of the longest axis. (a) Histogram of correctly segmented grains in regard to all grains. (b) Histogram of connected (merged) grains in regard to all grains.

To gain a better understanding of the quality of the results, the segmentation outcomes were analyzed as a function of grain size. During the experiments, the status of each grain (correctly segmented, connected, omitted) was analyzed in relation to grain size calculated based on the manually segmented reference data. As a result, the number of correctly recognized and connected grains in different size ranges was calculated. The results are shown in Fig. 9. The analysis confirmed that the highest error in grain segmentation occurs for the smallest grains with the size of 10–30 cm. The larger the grain, the smaller the probability

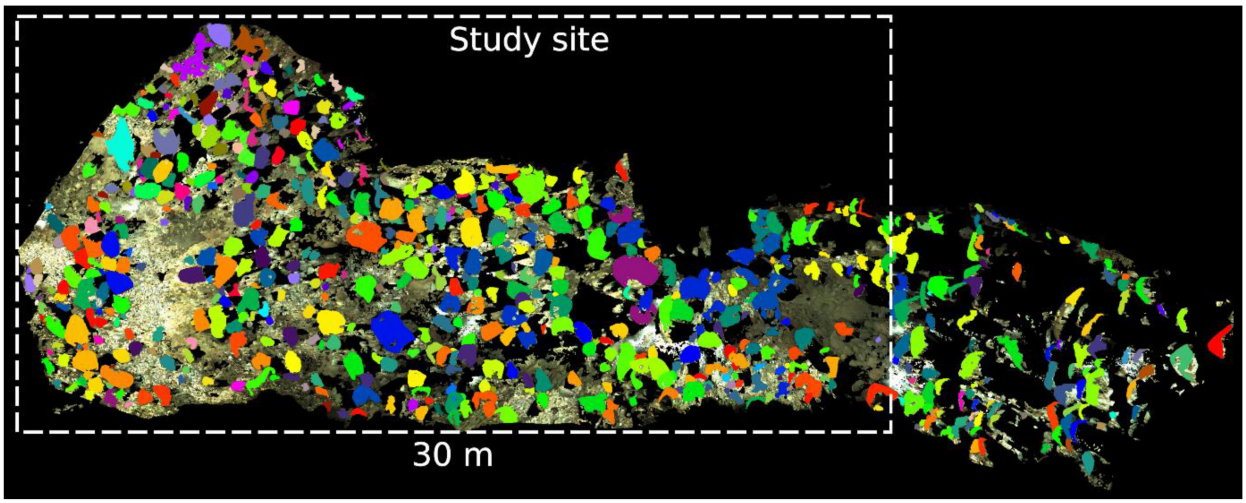


Fig. 10. Segmentation results for the whole study site. Each segmented grain is presented in a different color. The background (river bed) is shown with the original color of the colorized point cloud.

of segmentation error. The number of connected grains in each grain size range was similar for each grain size range.

Instance segmentation of grains from a TLS point cloud proved to be a challenging task because of several reasons. First, the reduction of the problem to the 2-D case is not possible, even after the classification step since it would lead to omitting small grains covered by very large fractions and, as a result, not visible in the top view. On the other hand, the processing of point clouds representing a complex structure of a river bed is complicated and leads to the generation of numerous clusters that do not represent individual grains. This effect is typical for instance segmentation tasks and is solved by the proper rejection method. However, it is a very complicated problem in the case of grains because of the following reasons. First, large sediments (diameter larger than 10 cm) present in the mountain river bed are distributed in a disordered way. Second, they are characterized by high variability of shapes and sizes. As a result, the possibilities to apply rejection criteria based on size, geometrical shape description, or the grain location are very limited. Finally, these problems are enhanced by the shadowing effect produced by TLS (line of sight obstruction). However, typically the grains are of a rounded shape. Therefore, in this investigation, the incorrectly detected grains were rejected when the delineated point cloud represented a flat object.

The proposed algorithm was also used to segment individual grains from the point cloud representing the whole river bed. The results of the experiment were visually inspected and suggest that the same quality as shown for datasets 1–4 can be obtained for the entire dataset (see Fig. 10).

### C. Comparison Method

Instance segmentation methods are highly adjusted to the delineation of a specified type of objects. Therefore, most of the methods developed for trees, buildings, or vehicles would need large adjustments to produce meaningful results of instance segmentation of individual grains. Therefore, the method described in this article is compared to the algorithm introduced in [6], which proved to be effective in the delineation of individual grains.

The method proposed by Wang *et al.* [6] is one of the data-driven approaches that are based on DEMs. As a result, the data are transformed to a 2.5-D space during the procedure. Although this approach enables to expedite the whole procedure, it is expected that some of the 3-D components will be missing in the results.

The algorithm works iteratively and is divided into the following five steps.

1) The DEM is created and smoothed using morphological operations (opening and closing).
2) The morphological pouring is applied for a rough division of the DEM into smaller regions that contain a dominative grain and/or some smaller fractions.
3) The active contours without edges algorithm are applied to delineate the dominant grain.
4) The values of pixels representing dominant grain are changed to background value.
5) If 95% of the DEM is covered with detected grains, the algorithm stops. Otherwise, it starts again from step 2).

For the comparison purposes, in this article, a DEM with a resolution of 1 cm was created based on the detrended point cloud of dataset 1. In the next step, the watershed algorithm was applied to the DEM to roughly divide the data into smaller regions. The watershed algorithm implemented in QGIS software was used for this purpose. The region expansion was started with local maxima values. The segments were joined based on seed to saddle distance. The threshold of this value was set to 5 cm. The results of first iteration of watershed algorithm are presented in Fig. 11. Then, the active contour without edges was applied to each region extracted by watershed. The algorithm was started in the pixel representing the maximum height value in the region to first detect a dominative grain. This was performed using scikit-image Python library [61].

After the application of the watershed algorithm, there are several regions that contain only one grain but typically each region consists of one dominative boulder and several smaller

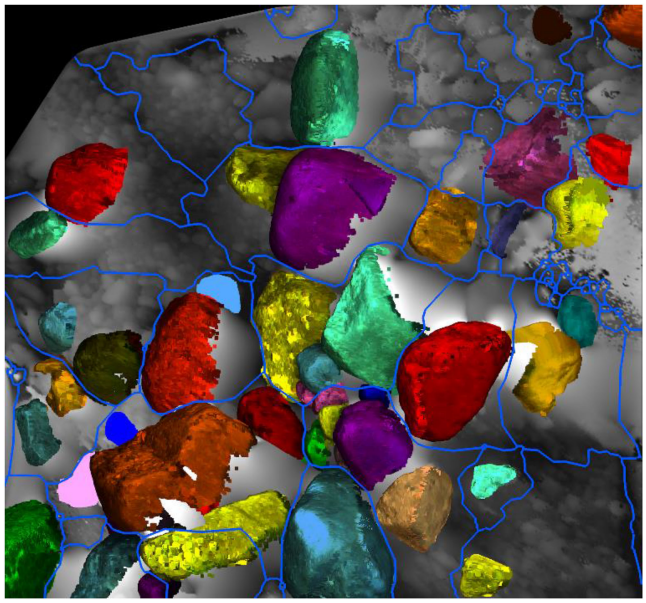

Fig. 11. Results of watershed segmentation. The background is DEM (in gray). The blue lines represent watersheds. The reference data are presented as grains in different colors.

boulders or cobbles. However, there are also individual regions that contain only very small fraction of the river bed grains. There is also one boulder that was divided into two regions. The selected results of the application of the active contour without edges algorithm are presented in Fig. 12. There are several cases when the algorithm presented very accurate results [e.g., Fig. 12(a)]. This happens when the grain lies flat at the river bed. As a result, there are no points belonging to the river bed or smaller grains under it. However, in the case of most of the larger boulders, the results present points belonging to both the grain and river bed or smaller grains under it [see Fig. 12(b)]. Moreover, additional errors caused by interpolation appear that result in both selecting several grains instead of one [see Fig. 12(c)] and only part of a grain.

The differences in the results between the algorithms are caused by different characteristics of the datasets used for their development. The data presented in [6] were acquired with MLS system for a river bed that consists of smaller rock fractions. As a result, the shadowing effect was very small. Moreover, most area of the river bed was covered with grains that were well sorted. Most of them do not overlay each other. Consequently, the algorithm presented the high accuracy of the results achieved in short processing time. However, in the complex scene scenario, when many poorly sorted grains of various sizes frequently overlay each other, the method does not produce accurate results. This effect is also enhanced with the influence of interpolation over shadows.

### *D. Advantages and Limitations*

In this article, we propose a method for an automatic segmentation of individual grains from TLS point clouds in the environment of the mountain river. Due to the character of the TLS measurement, which can be performed only from a limited number of stations, the resulting data are often influenced by shadows. The number of possible locations for a measurement station is additionally limited by a complex terrain relief in the proximity of mountain rivers. However, the conducted experiments proved that the proposed method is robust to the occurrence of shadowing effect. It was achieved by direct processing of point clouds instead of DSMs that require data interpolation in the shadowed areas. Moreover, contrary to the DSM-based approaches, the developed algorithm enables to segment grains that are partially or completely covered by larger fractions in the top view (see Fig. 13).

The proposed method is based only on the analysis of the geometrical characteristics of the point cloud in the neighborhoods of defined size. Since the laser scanning intensity information and RGB data interpolated from photographs are not used, the method does not require intensity correction and is independent from lighting conditions. Moreover, because of this reason, the method has a potential to be adapted to the processing of data collected by other techniques, such as photogrammetry or MLS.

However, since the method is based only on the analysis of geometry, small grains (longest axis smaller than 25 cm) are often incorrectly classified as a river bed. This problem is caused by the structure of a mountain river bed, which contains small rock fractions that, in some cases, may geometrically resemble a cobble (64–256 mm). Since the results of the proposed method are highly influenced by the classification quality, small grains are often omitted during the procedure. Moreover, because of both the classification inaccuracy and the point cloud density variation caused by shadowing effect, the individual grains segmented using the proposed method may contain holes (see Fig. 14). However, this situation does not occur often and it should not cause severe effects on the further results, because the grains are typically flat or slightly concave in the area of hole.

## V. Conclusion

The article presents research on the method of automatic segmentation of single grains from the TLS point cloud representing the mountain river bed. The performed experiments showed that the developed method enables correct identification and segmentation of 67%–88% of individual grains depending on the testing site. Moreover, in some cases, the developed method allowed for the segmentation of grains, the identification of which by humans was problematic and time-consuming.

The proposed method represents a classification followed by the segmentation approach and is based on random forest and DBSCAN algorithms. These algorithms are additionally supported by preprocessing and postprocessing stages enabling the reduction of processed points and ensuring proper separation of individual grains. The method was specifically designed for the robust processing of TLS data affected by shadowing effect and representing natural objects of oblong and rounded shapes. It enables segmentation of individual objects of different sizes, shapes, and represented by point clouds of different densities (5.2–32.1 points/cm$^2$).

The points belonging to individual grains were identified with a point cloud based completeness of ca. 88%, similar in

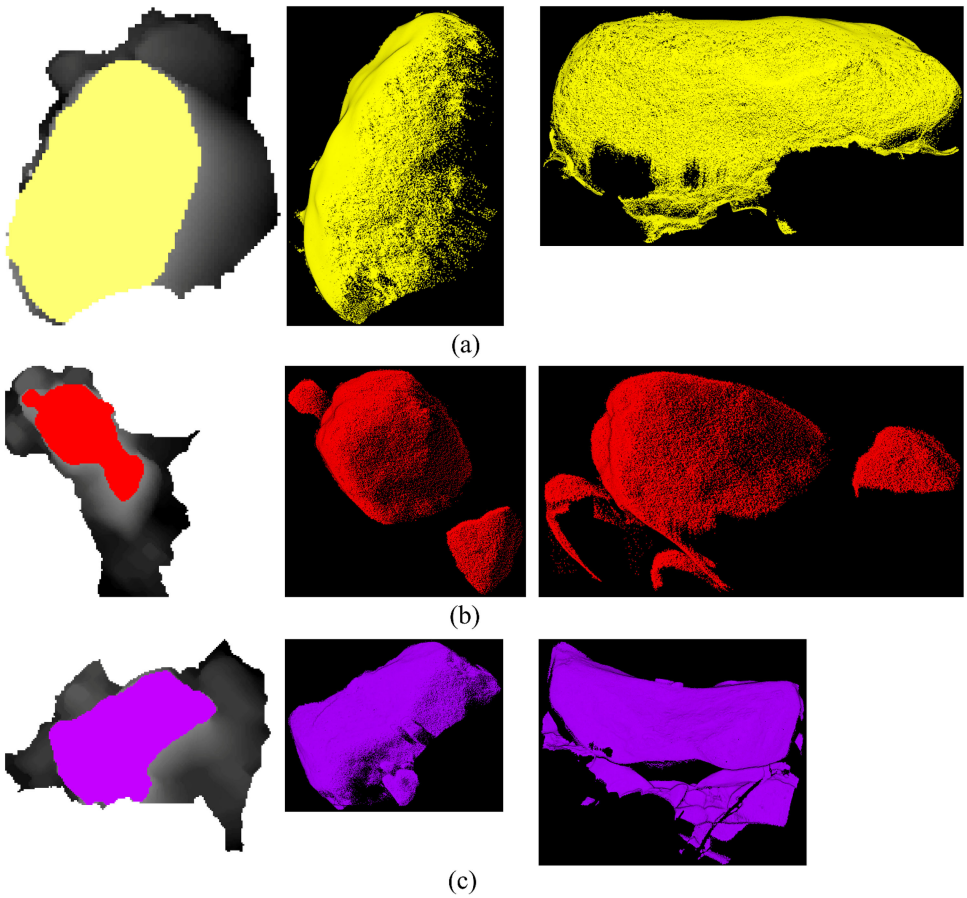


Fig. 12. Results of application of active contours without edges algorithm. Left: The resulting curve (colored) with a background of DEM. Middle: The top view of the point cloud delineated by a resulting curve. Right: The side view of the point cloud delineated by a resulting curve.

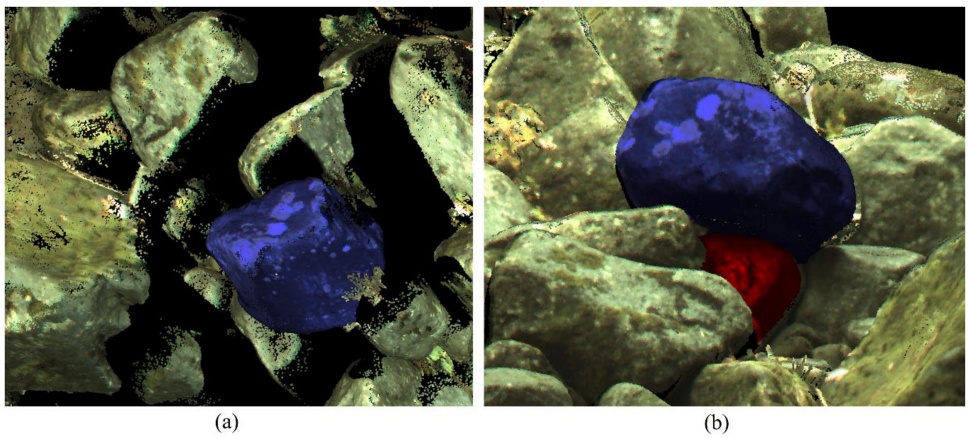


Fig. 13. Example of two segmented grains. (a) Top view. (b) Side view. The colors indicate segmented grains.

each dataset. The further investigations showed that the number of correctly recognized grains is highly dependent on the classification results. The classification of the point clouds was performed using random forest algorithm employing only geometrical features and supported by the custom feature-selection strategy. This approach enabled us to achieve the accuracy of 81–91%. The application of the majority filter allowed reducing the salt and pepper effect and led to more concise classification results. As a consequence, the classification accuracy increased to 85–95%, depending on the test site.

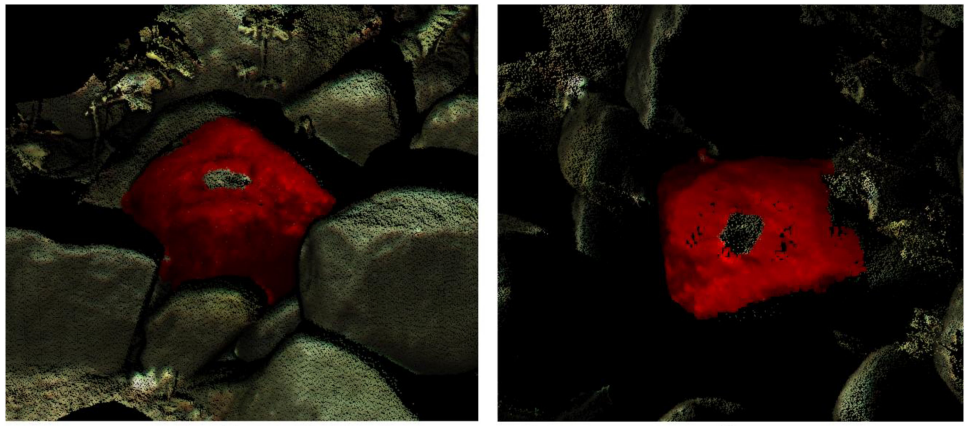


Fig. 14. Example of the segmented grain with a hole. (a) Top view. (b) Side view. The color indicates segmented grain.

The analysis of the number of correctly detected grains in regard to the estimated grain size showed that for the grains smaller than 25 cm only ca. 33% of grains are correctly segmented, whereas for the grains of sizes between 25 and 50 cm the segmentation accuracy achieves ca. 76% and for the grains larger than 60 cm the segmentation accuracy reaches the highest value of ca. 87%.

The proposed method proved to be useful for the segmentation of individual grains. In comparison to the existing methods, it processes directly point clouds instead of a DSM. As a result, it is robust to a high degree of shadowing and enables segmenting grains that are covered by large boulders and are not visible in the top view at the test site.

## APPENDIX

The reference data for the developed instance segmentation results are presented in Figs. 15 and 16.

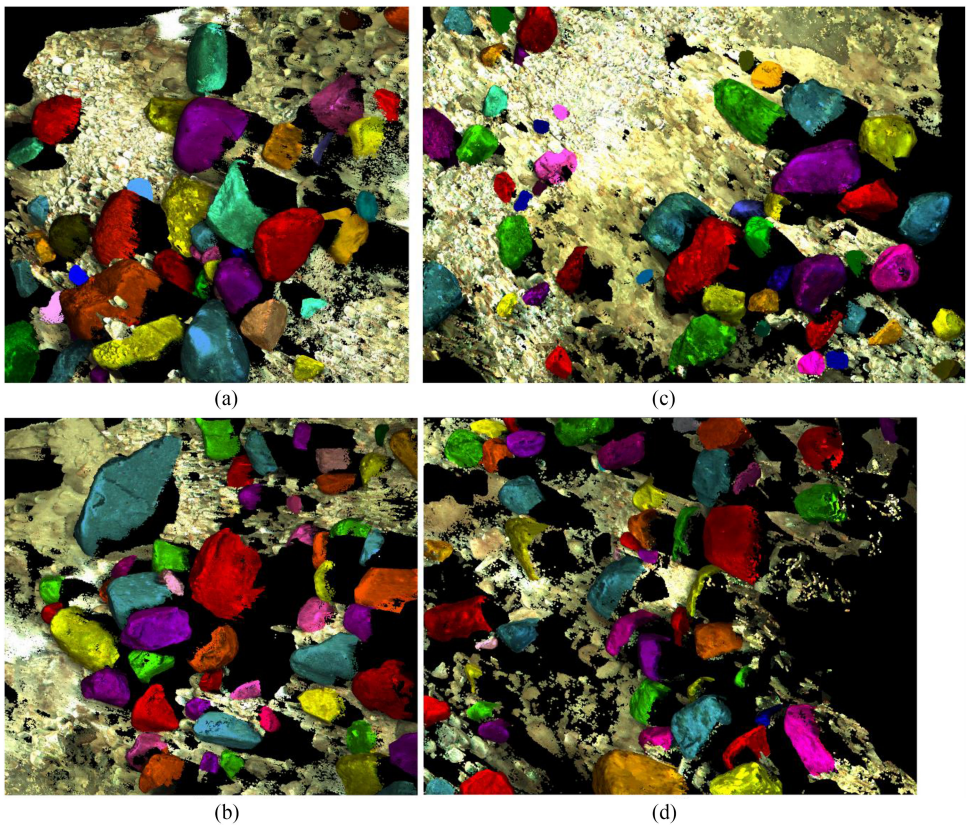


Fig. 15. Reference data for instance segmentation—top view. (a) Dataset 1. (b) Dataset 2. (c) Dataset 3. (d) Dataset 4. Each segmented grain is presented in a different color. The background (river bed) is shown with the original color of the colorized point cloud.

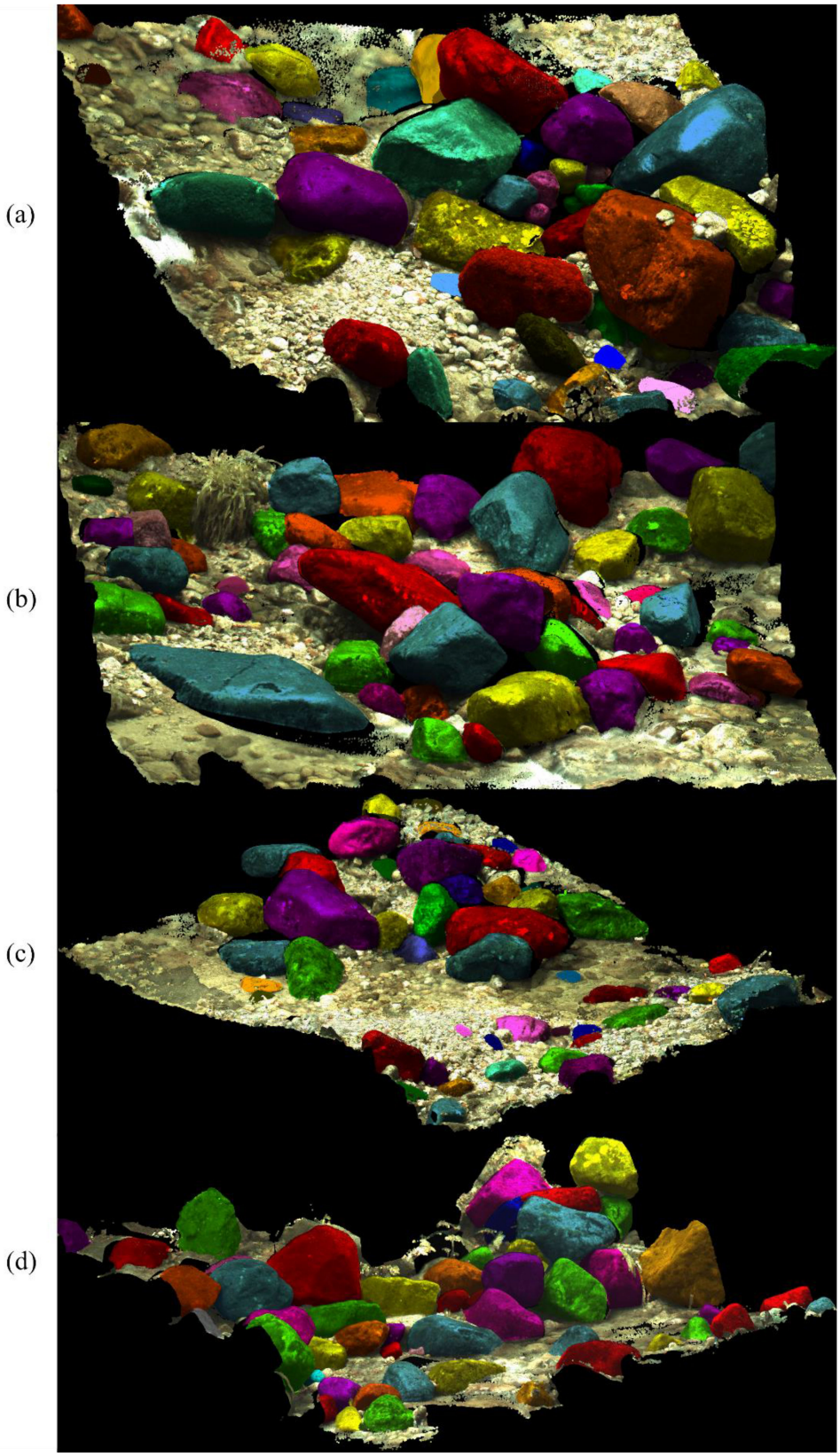


Fig. 16. Reference data for instance segmentation—side view. (a) Dataset 1. (b) Dataset 2. (c) Dataset 3. (d) Dataset 4. Each segmented grain is presented in a different color. The background (river bed) is shown with the original color of the colorized point cloud.

## ACKNOWLEDGMENT

The authors would like to thank the late Prof. A. Borkowski for his valuable comments on the early draft of the developed algorithm.

## REFERENCES

[1] J. Telling, A. Lyda, P. Hartzell, and C. Glennie, "Review of Earth science research using terrestrial laser scanning," *Earth-Sci. Rev.*, vol. 169, pp. 35–68, Jun. 2017, doi: 10.1016/j.earscirev.2017.04.007.

[2] A. R. Rice, R. Cassidy, P. Jordan, D. Rogers, and J. Arnscheidt, "Fine-scale quantification of stream bank geomorphic volume loss caused by cattle access," *Sci. Total Environ.*, vol. 769, May 2021, Art. no. 144468, doi: 10.1016/j.scitotenv.2020.144468.

[3] C.-W. Kuo, G. Brierley, and Y.-H. Chang, "Monitoring channel responses to flood events of low to moderate magnitudes in a bedrock-dominated river using morphological budgeting by terrestrial laser scanning," *Geomorphology*, vol. 235, pp. 1–14, Apr. 2015, doi: 10.1016/j.geomorph.2015.01.019.

[4] J. Brasington, D. Vericat, and I. Rychkov, "Modeling river bed morphology, roughness, and surface sedimentology using high resolution terrestrial laser scanning," *Water Resour. Res.*, vol. 48, no. 11, 2012, Art. no. W11519, doi: 10.1029/2012WR012223.

[5] E. Lotsari *et al.*, "Gravel transport by ice in a subarctic river from accurate laser scanning," *Geomorphology*, vol. 246, pp. 113–122, Oct. 2015, doi: 10.1016/j.geomorph.2015.06.009.

[6] Y. Wang *et al.*, "3D Modeling of coarse fluvial sediments based on mobile laser scanning data," *Remote Sens.*, vol. 5, no. 9, pp. 4571–4592, Sep. 2013, doi: 10.3390/rs5094571.

[7] A. Walicka, N. Pfeifer, A. Borkowski, and G. Jóźków, "An automatic method for the measurement of coarse particle movement in a mountain riverbed," *Measurement*, vol. 174, Apr. 2021, Art. no. 109029, doi: 10.1016/j.measurement.2021.109029.

[8] A. Walicka, G. Jóźków, M. Kasprzak, and A. Borkowski, "Terrestrial laser scanning for the detection of coarse grain size movement in a mountain riverbed," *Water*, vol. 11, no. 11, Nov. 2019, Art. no. 2199, doi: 10.3390/w11112199.

[9] G. Jozkow, A. Borkowski, and M. Kasprzak, "Monitoring of fluvial transport in the mountain river bed using terrestrial laser scanning," *Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci.*, vol. XLI-B7, pp. 523–528, Jun. 2016, doi: 10.5194/isprs-archives-XLI-B7-523-2016.

[10] G. Smart, J. Aberle, M. Duncan, and J. Walsh, "Measurement and analysis of alluvial bed roughness," *J. Hydraul. Res.*, vol. 42, no. 3, pp. 227–237, Jan. 2004, doi: 10.1080/00221686.2004.9728388.

[11] P. R. Wilcock and J. C. Crowe, "Surface-based transport model for mixed-size sediment," *J. Hydraul. Eng.*, vol. 129, no. 2, pp. 120–128, Feb. 2003, doi: 10.1061/(ASCE)0733-9429(2003)129:2(120).

[12] E. Pearson, M. W. Smith, M. J. Klaar, and L. E. Brown, "Can high resolution 3D topographic surveys provide reliable grain size estimates in gravel bed rivers?," *Geomorphology*, vol. 293, pp. 143–155, Sep. 2017, doi: 10.1016/j.geomorph.2017.05.015.

[13] M. G. Wolman, "A method of sampling coarse river-bed material," *EOS, Trans. Amer. Geophys. Union*, vol. 35, no. 6, pp. 951–956, 1954, doi: 10.1029/TR035i006p00951.

[14] B. Purinton and B. Bookhagen, "Introducing *Pebblecounts*: A grain-sizing tool for photo surveys of dynamic gravel-bed rivers," *Earth Surf. Dyn.*, vol. 7, no. 3, pp. 859–877, Sep. 2019, doi: 10.5194/esurf-7-859-2019.

[15] D. Buscombe, D. M. Rubin, and J. A. Warrick, "A universal approximation of grain size from images of noncohesive sediment," *J. Geophys. Res. F, Earth Surf.*, vol. 115, no. F2, 2010, Art. no. F02015, doi: 10.1029/2009JF001477.

[16] Z. Cheng and H. Liu, "Digital grain-size analysis based on autocorrelation algorithm," *Sedimentary Geol.*, vol. 327, pp. 21–31, Aug. 2015, doi: 10.1016/j.sedgeo.2015.07.008.

[17] F.-J. Chang and C.-H. Chung, "Estimation of riverbed grain-size distribution using image-processing techniques," *J. Hydrol.*, vol. 440–441, pp. 102–112, May 2012, doi: 10.1016/j.jhydrol.2012.03.032.

[18] D. Vázquez-Tarrío and R. Menéndez-Duarte, "Assessment of bedload equations using data obtained with tracers in two coarse-bed mountain streams (Narcea River Basin, NW Spain)," *Geomorphology*, vol. 238, pp. 78–93, Jun. 2015, doi: 10.1016/j.geomorph.2015.02.032.

[19] P. R. Wilcock, "Toward a practical method for estimating sediment-transport rates in gravel-bed rivers," *Earth Surf. Processes Landforms*, vol. 26, no. 13, pp. 1395–1408, 2001, doi: 10.1002/esp.301.

[20] A. Recking, P. Frey, A. Paquier, P. Belleudy, and J. Y. Champagne, "Feedback between bed load transport and flow resistance in gravel and cobble bed rivers," *Water Resour. Res.*, vol. 44, no. 5, 2008, Art. no. W05412, doi: 10.1029/2007WR006219.

[21] N. Bergman, J. B. Laronne, and I. Reid, "Benefits of design modifications to the birkbeck bedload sampler illustrated by flash-floods in an ephemeral gravel-bed channel," *Earth Surf. Processes Landforms*, vol. 32, no. 2, pp. 317–328, 2007, doi: 10.1002/esp.1453.

[22] L. Olinde and J. P. L. Johnson, "Using RFID and accelerometer-embedded tracers to measure probabilities of bed load transport, step lengths, and rest times in a mountain stream," *Water Resour. Res.*, vol. 51, no. 9, pp. 7572–7589, 2015, doi: 10.1002/2014WR016120.

[23] A. C. Schwendel and D. J. Milan, "Terrestrial structure-from-motion: Spatial error analysis of roughness and morphology," *Geomorphology*, vol. 350, Feb. 2020, Art. no. 106883, doi: 10.1016/j.geomorph.2019.106883.

[24] A. Walicka, G. Jóźków, and A. Borkowski, "Individual rocks segmentation in terrestrial laser scanning point cloud using iterative DBSCAN algorithm," *Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci.*, vol. XLII-2, pp. 1157–1161, May 2018, doi: 10.5194/isprs-archives-XLII-2-1157-2018.

[25] Y. Shen, R. Lindenbergh, J. Wang, and V. G. Ferreira, "Extracting individual bricks from a laser scan point cloud of an unorganized pile of bricks," *Remote Sens.,* vol. 10, no. 11, Nov. 2018, Art. no. 1709, doi: 10.3390/rs10111709.

[26] Y. Zhang, W. Yang, X. Liu, Y. Wan, X. Zhu, and Y. Tan, "Unsupervised building instance segmentation of airborne LiDAR point clouds for parallel reconstruction analysis," *Remote Sens.*, vol. 13, no. 6, Jan. 2021, Art. no. 1136, doi: 10.3390/rs13061136.

[27] W. Xiao, B. Vallet, K. Schindler, and N. Paparoditis, "Street-side vehicle detection, classification and change detection using mobile laser scanning data," *ISPRS J. Photogramm. Remote Sens.*, vol. 114, pp. 166–178, Apr. 2016, doi: 10.1016/j.isprsjprs.2016.02.007.

[28] J. Zhang, W. Xiao, B. Coifman, and J. P. Mills, "Vehicle tracking and speed estimation from roadside lidar," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 13, pp. 5597–5608, Sep. 2020, doi: 10.1109/JSTARS.2020.3024921.

[29] H. Luo, K. Khoshelham, C. Chen, and H. He, "Individual tree extraction from urban mobile laser scanning point clouds using deep pointwise direction embedding," *ISPRS J. Photogramm. Remote Sens.*, vol. 175, pp. 326–339, May 2021, doi: 10.1016/j.isprsjprs.2021.03.002.

[30] Y. Wang, T. Jiang, J. Liu, X. Li, and C. Liang, "Hierarchical instance recognition of individual roadside trees in environmentally complex urban areas from UAV laser scanning point clouds," *ISPRS Int. J. Geo-Inf.*, vol. 9, no. 10, Oct. 2020, Art. no. 595, doi: 10.3390/ijgi9100595.

[31] E. Hadas, G. Jozkow, A. Walicka, and A. Borkowski, "Apple orchard inventory with a LiDAR equipped unmanned aerial system," *Int. J. Appl. Earth Observ. Geoinf.*, vol. 82, Oct. 2019, Art. no. 101911, doi: 10.1016/j.jag.2019.101911.

[32] Z. Koma, M. Rutzinger, and M. Bremer, "Automated segmentation of leaves from deciduous trees in terrestrial laser scanning point clouds," *IEEE Geosci. Remote Sens. Lett.*, vol. 15, no. 9, pp. 1456–1460, Sep. 2018, doi: 10.1109/LGRS.2018.2841429.

[33] F. Di Salvo and M. L. Brutto, "Full-Waveform terrestrial laser scanning for extracting a high-resolution 3D topographic model: A case study on an area of archaeological significance," *Eur. J. Remote Sens.*, vol. 47, no. 1, pp. 307–327, Jan. 2014, doi: 10.5721/EuJRS20144719.

[34] A. Djuricic *et al.*, "High-resolution 3D surface modeling of a fossil oyster reef," *Geosphere*, vol. 12, no. 5, pp. 1457–1477, Oct. 2016, doi: 10.1130/GES01282.1.

[35] T. Rabbani and F. van den Heuvel, "Efficient Hough transform for automatic detection of cylinders in point clouds," in *Proc. ISPRS WG III/3, III/4, V/3 Workshop Laser Scanning*, 2005, pp. 60–65.

[36] R. B. Rusu, Z. C. Marton, N. Blodow, A. Holzbach, and M. Beetz, "Model-based and learned semantic object labeling in 3D point cloud maps of kitchen environments," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Oct. 2009, pp. 3601–3608, doi: 10.1109/IROS.2009.5354759.

[37] R. Boesch, "Model based automatic segmentation of tree stems from single scan data," *Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci.*, vol. XL-7/W2, pp. 49–53, Oct. 2013, doi: 10.5194/isprsarchives-XL-7-W2-49-2013.

[38] M. Miraki, H. Sohrabi, P. Fatehi, and M. Kneubuehler, "Individual tree crown delineation from high-resolution UAV images in broadleaf forest," *Ecol. Inform.*, vol. 61, Mar. 2021, Art. no. 101207, doi: 10.1016/j.ecoinf.2020.101207.

[39] J. Yang, Z. Kang, S. Cheng, Z. Yang, and P. H. Akwensi, "An individual tree segmentation method based on watershed algorithm and three-dimensional spatial distribution analysis from airborne LiDAR point clouds," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 13, pp. 1055–1067, Mar. 2020, doi: 10.1109/JSTARS.2020.2979369.
[40] M. Weinmann, M. Weinmann, C. Mallet, and M. Brédif, "A classification-segmentation framework for the detection of individual trees in dense MMS point cloud data acquired in urban areas," *Remote Sens.*, vol. 9, no. 3, Mar. 2017, Art. no. 277, doi: 10.3390/rs9030277.
[41] W. Yao, S. Hinz, and U. Stilla, "Extraction and motion estimation of vehicles in single-pass airborne LiDAR data towards urban traffic analysis," *ISPRS J. Photogramm. Remote Sens.*, vol. 66, no. 3, pp. 260–271, May 2011, doi: 10.1016/j.isprsjprs.2010.10.005.
[42] M. Rutzinger, B. Höfle, M. Hollaus, and N. Pfeifer, "Object-based point cloud analysis of full-waveform airborne laser scanning data for urban vegetation classification," *Sensors*, vol. 8, no. 8, pp. 4505–4528, Aug. 2008, doi: 10.3390/s8084505.
[43] W. Yao, S. Hinz, and U. Stilla, "3D object-based classification for vehicle extraction from airborne LiDAR data by combining point shape information with spatial edge," in *Proc. IAPR Workshop Pattern Recognit. Remote Sens.*, Aug. 2010, pp. 1–4, doi: 10.1109/PRRS.2010.5742804.
[44] J. Zhang, M. Duan, Q. Yan, and X. Lin, "Automatic vehicle extraction from airborne LiDAR data using an object-based point cloud analysis method," *Remote Sens.*, vol. 6, no. 9, pp. 8405–8423, Sep. 2014, doi: 10.3390/rs6098405.
[45] T. Zhang, Y. Kan, H. Jia, C. Deng, and T. Xing, "Urban vehicle extraction from aerial laser scanning point cloud data," *Int. J. Remote Sens.*, vol. 41.17, pp. 6664–6697, Jun. 2020, doi: 10.1080/01431161.2020.1742947.
[46] Y. Yu, J. Li, H. Guan, D. Zai, and C. Wang, "Automated extraction of 3D trees from mobile LiDAR point clouds," *Int. Arch. Photogramm., Remote Sens. Spatial Inf. Sci.*, vol. XL-5, pp. 629–632, Jun. 2014, doi: 10.5194/isprsarchives-XL-5-629-2014.
[47] S. Xu, N. Ye, S. Xu, and F. Zhu, "A supervoxel approach to the segmentation of individual trees from LiDAR point clouds," *Remote Sens. Lett.*, vol. 9, no. 6, pp. 515–523, Jun. 2018, doi: 10.1080/2150704X.2018.1444286.
[48] Z. Kang, J. Yang, R. Zhong, Y. Wu, Z. Shi, and R. Lindenbergh, "Voxel-based extraction and classification of 3-D pole-like objects from mobile LiDAR point cloud data," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 11, no. 11, pp. 4287–4298, Nov. 2018, doi: 10.1109/JSTARS.2018.2869801.
[49] A. Zarea and A. Mohammadzadeh, "A novel building and tree detection method from LiDAR data and aerial images," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 9, no. 5, pp. 1864–1875, May 2016, doi: 10.1109/JSTARS.2015.2470547.
[50] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proc. 2nd Int. Conf. Knowl. Discov. Data Mining*, 1996, pp. 226–231.
[51] D. Opitz and R. Maclin, "Popular ensemble methods: An empirical study," *J. Artif. Intell. Res.*, vol. 11, pp. 169–198, Aug. 1999, doi: 10.1613/jair.614.
[52] N. Brodu and D. Lague, "3D terrestrial lidar data classification of complex natural scenes using a multi-scale dimensionality criterion: Applications in geomorphology," *ISPRS J. Photogramm. Remote Sens.*, vol. 68, pp. 121–134, Mar. 2012, doi: 10.1016/j.isprsjprs.2012.01.006.
[53] M. Weinmann, B. Jutzi, S. Hinz, and C. Mallet, "Semantic point cloud interpretation based on optimal neighborhoods, relevant features and efficient classifiers," *ISPRS J. Photogramm. Remote Sens.*, vol. 105, pp. 286–304, Jul. 2015, doi: 10.1016/j.isprsjprs.2015.01.016.
[54] Y. Saeys, I. Inza, and P. Larranaga, "A review of feature selection techniques in bioinformatics," *Bioinformatics*, vol. 23, no. 19, pp. 2507–2517, Oct. 2007, doi: 10.1093/bioinformatics/btm344.
[55] Q. Li, F. K. K. Wong, and T. Fung, "Mapping multi-layered mangroves from multispectral, hyperspectral, and LiDAR data," *Remote Sens. Environ.*, vol. 258, Jun. 2021, Art. no. 112403, doi: 10.1016/j.rse.2021.112403.
[56] C. Mallet, F. Bretar, M. Roux, U. Soergel, and C. Heipke, "Relevance assessment of full-waveform lidar data for urban area classification," *ISPRS J. Photogramm. Remote Sens.*, vol. 66, no. 6, pp. S71–S84, Dec. 2011, doi: 10.1016/j.isprsjprs.2011.09.008.
[57] M. Weinmann, B. Jutzi, C. Mallet, and M. Weinmann, "Geometric features and their relevance for 3D point cloud classification," *ISPRS Ann. Photogramm. Remote Sens. Spatial Inf. Sci.*, vol. IV-1/W1, pp. 157–164, May 2017, doi: 10.5194/isprs-annals-IV-1-W1-157-2017.
[58] C. Spearman, "The proof and measurement of association between two things," *Amer. J. Psychol.*, vol. 15, pp. 72–101, 1904.
[59] P. Jaccard, "The distribution of the flora in the alpine zone," *New Phytologist*, vol. 11, no. 2, pp. 37–50, 1912.
[60] I. T. Jolliffe, Ed., "Principal components in regression analysis," in *Principal Component Analysis*. New York, NY, USA: Springer-Verlag, 2002, pp. 167–198, doi: 10.1007/0-387-22440-8_8.
[61] S. van der Walt *et al.*, "Scikit-image: Image processing in Python," *PeerJ*, vol. 2, Jun. 2014, Art. no. e453, doi: 10.7717/peerj.453.

**Agata Walicka** received the bachelor's and master's degrees in geodesy and cartography from Wrocław University of Environmental and Life Sciences, Wrocław, Poland, in 2015 and 2016, respectively, where since 2016 she has been working toward the Ph.D. degree in application of terrestrial laser scanning for fluvial transport monitoring with the Institute of Geodesy and Geoinformatics.

In 2019, she became a Research-Teaching Assistant with the Institute of Geodesy and Geoinformatics, Wrocław University of Environmental and Life Sciences. Her research interests include application of terrestrial and UAV laser scanning for environmental monitoring.

**Norbert Pfeifer** was born in Vienna, Austria, in 1971. He received the Dipl.Ing. and Ph.D. degrees in surveying engineering from Technische Universität Wien (TU Wien), Vienna, in 1997 and 2002, respectively.

From 2003 to 2006, he was a Research Assistant and Assistant Professor with TU Delft, Delft, The Netherlands. In 2006, he became a Lecturer with the Department of Geography, University of Innsbruck, Innsbruck, Austria, and a Senior Researcher with the Centre for Natural Hazard Management, alp-S, Innsbruck. Later in 2006, he became a Professor in photogrammetry with TU Wien. He has coauthored more than 100 articles in journals and six books. His research interests include LiDAR signal processing, photogrammetry, calibration of airborne and terrestrial laser scanning and photos, classification and segmentation of point clouds, 3-D modeling, and application of point clouds in the environmental sciences.

Prof. Pfeifer is a member of the International Society of Photogrammetry and Remote Sensing.