applied optics

Classification of airborne 3D point clouds regarding separation of vegetation in complex environments

Dimitri Bulatov,[1,*] Dominik Stütz,[1] Jorg Hacker,[2,3] and Martin Weinmann[4]

[1]Fraunhofer IOSB, Gutleuthausstr. 1, 76275 Ettlingen, Germany
[2]ARA-Airborne Research Australia, Adelaide, SA, Australia
[3]College of Science and Engineering, Flinders University, Adelaide, SA, Australia
[4]Karlsruhe Institute of Technology (KIT), Institute of Photogrammetry and Remote Sensing, Karlsruhe, Germany
*Corresponding author: dimitri.bulatov@iosb.fraunhofer.de

Received 19 February 2021; revised 30 March 2021; accepted 31 March 2021; posted 12 April 2021 (Doc. ID 422973); published 17 May 2021

Classification of outdoor point clouds is an intensely studied topic, particularly with respect to the separation of vegetation from the terrain and manmade structures. In the presence of many overhanging and vertical structures, the (relative) height is no longer a reliable criterion for such a separation. An alternative would be to apply supervised classification; however, thousands of examples are typically required for appropriate training. In this paper, an unsupervised and rotation-invariant method is presented and evaluated for three datasets with very different characteristics. The method allows us to detect planar patches by filtering and clustering so-called superpoints, whereby the well-known but suitably modified random sampling and consensus (RANSAC) approach plays a key role for plane estimation in outlier-rich data. The performance of our method is compared to that produced by supervised classifiers common for remote sensing settings: random forest as learner and feature sets for point cloud processing, like covariance-based features or point descriptors. It is shown that for point clouds resulting from airborne laser scans, the detection accuracy of the proposed method is over 96% and, as such, higher than that of standard supervised classification approaches. Because of artifacts caused by interpolation during 3D stereo matching, the overall accuracy was lower for photogrammetric point clouds (74–77%). However, using additional salient features, such as the normalized green–red difference index, the results became more accurate and less dependent on the data source.

https://doi.org/10.1364/AO.422973

1. INTRODUCTION

Due to the increasing availability of inexpensive laser technologies and advanced processing pipelines for photogrammetric reconstruction, the generation of digital 3D point clouds of a high quality and density is becoming more readily possible, and with it also the scope and number of applications of 3D point clouds is growing. In the construction industry, 3D point clouds acquired from the ground are very useful for inventory purposes and quality control. In bathymetry, laser scans can provide information on the state of health of coral reefs. Civil engineers and city planners require airborne point clouds as the necessary basis for planning and concept development on disaster prevention (landslide, afforestation measures, etc.). Another interesting field of application is archaeology in terms of detection, documentation, and monitoring of cultural heritage. Hidden deeply underneath vegetation in remote places of the Earth, remnants of historic and even pre-historic civilization can be detected by analyzing point clouds from airborne LiDAR [1]. Not less important is to elaborate solutions for preventing degradation of monuments of cultural heritage caused by climate change and ever-advancing urbanization. A good example addressing this issue is the HERACLES [2] project funded by the Horizon 2020 research and innovation program of the European Union. To perform analysis of possible erosion within an *in situ* measurement campaign efficiently, a large-scale preparation of data and identification of particularly threatened spots is sensible. In the concrete case, it was achieved by unmanned aerial vehicle (UAV)-borne videos used for 3D reconstruction in terms of dense point clouds [3].

A prerequisite for point cloud analysis is an adequate description of its entities and/or objects of interest. For example, monitoring cultural heritage within a time series can be accomplished in a more efficient way if vegetation areas are identified; thus, differentiation between those points changing seasonally and those indicating deterioration of substance is made. This is important for two reasons. First, trees and bushes are not much

relevant for model representation and monument preservation and should ideally be omitted or modeled by generic models [4]. Second, especially in the case of HERACLES, where data transfer (under the field conditions, but also between different institutions in several European countries) must succeed quickly, context-aware compression of point clouds was helpful. Due to the complexity and variety of point clouds caused by strongly irregular sampling density, different types of objects, noise, and outliers, point cloud filtering is an increasingly active yet challenging field of research [5].

The recently developed superpoints in random sampling and consensus (RANSAC) planes (SiRP) approach allows us to subdivide airborne point clouds into locally planar and non-planar regions [6]. In case of only two classes (terrain and vegetation), the off-terrain points can be filtered away, thus allowing us to compress point clouds significantly. According to our extensive literature review, SiRP is one of the few algorithms allowing for both: pure 3D structure of data and scarcity, or even absence, of training examples. Otherwise, if there are more classes, the application of SiRP through a moderately interactive effort allows retrieving training examples for post-processing and/or successive supervised classification. For post-processing, a few salient features, particularly helpful for the data acquired by a certain sensor, can be analyzed point- or cluster-wise. For supervised classification, training data can be used with arbitrary feature sets used for the discriminative model computation (such as multiple splits for random forests or regression parameters for logistic regression). In this work, we assess two generic feature sets, namely, covariance-based features [7] and fast point feature histograms (FPFHs) [8]. Besides, we use specific features to boost the classification accuracy, in order to compare their performance with that of SiRP.

After providing an overview about the state-of-the-art on point cloud filtering and classification methods and presenting the used datasets in Sections 2 and 3, respectively, the SiRP method and supervised classification framework are explained in Section 4 and are evaluated in Section 5. More profound discussion on results regarding potential and limitations of the proposed method, comparison with state-of-the-art, and remarks on computing time, takes place in Section 6. Finally, Section 7 concludes this paper.

2. RELATED WORK

There is a large amount of methods on 3D point cloud filtering. The pioneering work [9] for extracting ground surface and, thus, a digital terrain model (DTM) from airborne laser scanning data was an iterative procedure for robust plane fitting. Hereby, the off-terrain points lie far above but not far below the plane, which was regulated by weights. This worked well in predominantly flat terrain, but, in hilly terrain, this procedure had to be replaced by a locally adaptive one. Therefore, algorithms based on slope-based filtering [10,11], edge-based filtering [12], segment-based filtering [13], progressive morphological filtering [14], and hierarchical filtering [15], as well as contour analysis [16], were proposed. Within an energy minimization framework, three properties characterizing the terrain could be connected and exploited [16]: (1) the terrain is supposed to pass through the local minima of the elevation function, (2) the slope between two terrain points is supposed to be moderate, and (3) the off-terrain objects are of limited size. Relying on these assumptions and trying to achieve more stability in certain cases, many procedures developed afterward contained two modules: ground point extraction and surface fitting [17–19]. For more references on DTM computation algorithms, we refer to the recently published survey [20].

However, for pure 3D point clouds, the concept of digital surface models (DSMs) and DTMs is widely senseless because the elevation cannot be represented as a continuous function of longitude and latitude [21]. Therefore, and also because a tighter focus of research was put on point classification tasks involving more than two classes, the number of criteria, or features, for differentiation needed to be increased as well. At the same time, application of machine learning methods allowed us to compute those thresholds automatically, which previously had to be selected interactively. A good example is [4], where the features planarity, elevation, scatter, and linear grouping were used to process pure 3D point clouds: the authors' concern was about the choice of the activation parameters σ for each of these features. Similarly, Guo *et al*. [22] utilized 26 handcrafted features to classify 3D points using JointBoost. In 1999, the authors of [7] introduced the features based on the eigenvalues of the structure tensor over points' neighbors. These features were used in remote sensing by numerous authors ([23–26] and others) in order to classify points with respect to their geometric saliency. For example, a point in a scene plane should have two eigenvalues $\lambda_{1/2}$ around 0.25, and the third eigenvalue λ_3 should be negligible. Thus, the planarity measure $(\lambda_2 - \lambda_3)/\lambda_1$ was defined by the authors of [23] who applied these covariance-based features in order to search for 3D line segments. More recently, [24] focused on determining the most relevant geometric features extracted in three typical types of neighborhoods: spherical, cylindrical, and those based on the k nearest neighbors (kNNs). Thus, varying values of k, radius, and cylinder height and calculating features for each of these configurations may produce arbitrary large feature sets.

Thus, in such overly large sets of features, also referred to as descriptors, the semantic interpretability of single features is gradually getting lost. Numerous examples of descriptors are listed by the survey [27], and we discuss two of them. Starting at the structure tensor, the authors of [28] compute the normal vector direction at the point and, with it, the local relative coordinate frame between two neighboring points. Then, triplets of angular differences are computed, and their discrete values are sampled into three histograms. Usually, there are five bins for every angle and, thus, the so-called point feature histogram (PFH) descriptor contains $5^3 = 125$ elements. This histogram is successively normalized to have the sum of entries of 100. That is, for a point with a chaotic vicinity typical for trees, the distribution of points is uniform, while for points located on a planar surface (including a degenerated one, that is, a linear structure), one strong peak around the 62nd ($\approx 125/2$) entry is expected. For points upon non-planar surfaces, there are several local maxima whose mutual distance to each other corresponds to a multiple of 5. The FPFHs [8] differ from PFHs in that the angular features of a point are computed once for all and then accumulated by a convex combination of features over adjacent points weighted by a distance-based factor. As a consequence,

the dependence of computational time on the neighborhood size is not anymore quadratic but linear, even though the accuracy slightly decreases in comparison with PFHs. Besides, a sophisticated approach was used to select the most relevant histogram entries, which greatly reduced their number (33 versus 125).

While the key idea of the descriptor is to compute features of features (e.g., histograms over deviations of normal vectors) and, thus, to capture shape primitives better by taking into account a broader context (or an increasing *receptive field*), this process can be iteratively repeated. In image processing, this idea culminated in the use of image pyramids and approaches based on deep learning, which are now often applied in 3D. In their literature review, the authors of [29] differentiated between the raster-based deep learning methods and those directly applied to point clouds. The raster-based methods for two- [30] and multi-class [31,32] problems are actually 2D, only that one channel or more is replaced by the DSM or some products thereof (like normalized DSM) or, possibly, rendering a mesh to such a local coordinate system from which it looks like 2.5D [33]. In a raster-based method in 3D, the convolutions are performed within a sparsely populated voxel grid, and the two max-pooling layers were unpooled within an U-net-like framework [34]. Two pioneering methods on point cloud processing using deep learning directly on the point cloud are PointNet [35] and its modification PointNet++ [36]. They create a neighborhood feature vector as a function acting on the point cloud. Then convolutional layers over neighbors are replaced by parsing some attributes over neighbors, while the pooling layers, needed to increase the receptive fields, are analogously statistical measures over the attributes. The difference between both networks is that subsampling is more consistently applied in PointNet++. Further examples of contributions involving deep learning in 3D are, among others, [37] (for point filtering) as well as [29,38,39]. Using points' coordinates, color values, and, optionally, several other features, the authors of [39] first perform an unsupervised classification of the point cloud into so-called superpoints. The PointNet-based features are computed over the support set of larger superpoints. At this stage, training data are required. After assigning *a priori* probabilities, the inference framework allows to propagate information and suppress noise in the classification result. In [38], the efficient pointwise pyramid pooling module is investigated to capture local structures at various densities by considering a multi-scale neighborhood. The neural network processes the multiscale features along sweeping directions. The contributions of [29] are, among others, taking into account further attributes (such as laser point intensities) and carrying out a thorough research on the choice of hyper-parameters of the neural network.

From the literature review, it is not surprising that with time, the methods are becoming more accurate and at the same time able to perform a multi-label classification, even though the framework conditions, such as datasets, settings, and the presence of absence of training data, are changing. Thus, according to [13], the method of [9] for binary classification has achieved an overall accuracy (OA) below 80%. Less than 20 years later, [36] achieved a performance over 84% on a 3D indoor point cloud with 17 classes using PointNet++ while [34] achieved an accuracy of 83.7% on the Vaihingen benchmark dataset with eight classes [40]. Another clearly observable recent trend is to compute as many higher-level features as possible for a thinned point cloud with respect to the original one. In fact, the computation of these features over all points negatively affects the performance speed while most of these points are actually easy to classify. For example, the authors of [8] show how to determine "interesting" points in order to appropriately align partial scans of an object or a scene of interest, while the authors of [41] propose to subsample the original point cloud based on voxel grids and kd-trees to achieve a more efficient classification. From a scale pyramid created by repeatedly downsampling the point cloud, a small number of nearest neighbors is used at each scale for feature computation [41]. The difference to our approach is that we neither compute a kd-structure for setting the centroids of voxels nor use a rigid grid but use a flexible structure in which points are clustered using a fixed tolerance value. Our task is filtering out vegetation, whereby what remains, e.g., terrain, is not supposed to represent a horizontal surface but can be vertical or explicitly 3D. To do this, we apply a modification of the well-known RANSAC algorithm [42], specially tailored to this task. The thinned points are then clustered to enable correction of gross errors by an everything-or-nothing interactive procedure, or to (re)label some of them, if more than two classes are of interest. For comparison, supervised classification based on standard sets of commonly used geometric features is applied to the original point cloud.

3. DATASETS

The proposed method is supposed to perform a good point filtering for both airborne laser scans and products of photogrammetric reconstructions. However, parameter settings and post-processing routines are somehow different. In order to address these differences and to emphasize the necessity of point filtering in regard with the applications mentioned in Section 1, we decided to bring this chapter forward and to present some application-related context of all chosen datasets.

A. Strathalbyn

The first dataset, named Strathalbyn, originates from Australia and was kindly provided by the authors of [43,44] in an already classified form which we used as ground truth. The data are from a basically flat landscape close to the Burdekin River in North Queensland, Australia, and feature severe gully erosion. The spread of this erosion, prevalent in and around river basins there, is usually linked to clearing native vegetation. It not only degrades the landscape in many ways, including the underlaying hydrology, but the sediments washed into the waterways also affect the water quality in the catchments, which causes degradation of parts of the Great Barrier Reef. To preserve the ecological balance, both the Australian and the Queensland Governments have launched major initiatives focused on gully rehabilitation. In the dataset, analyzed by [43] and presented in Fig. 1(a), there are many trees of different types. Apart from determining the exact shape of the bare ground, their detection is indispensable to track the erosion and the rehabilitation efforts by geo-engineering and tree planting. All data was acquired and processed to classified point clouds and

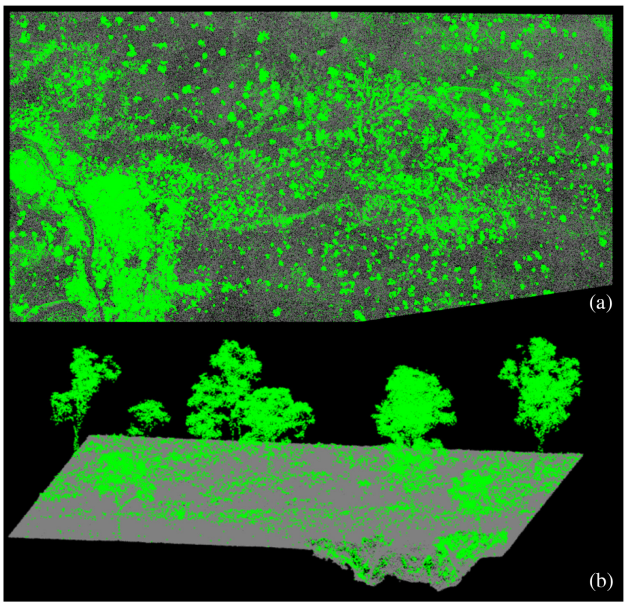

Fig. 1. Two views of the Strathalbyn dataset with the ground truth displayed in green color: (a) Nadir view of the dataset and (b) the detailed view.

Table 1. Dataset Properties (Approximately) and Algorithm Parameters for the Datasets from Section 3[a]

Dataset	S[b]	O[c]	G[d]
Data source	LiDAR	LiDAR	Phot.
Flight alt. (m)	100	2800	80
Pt. density (pts/m^2)	100	50	3700
Number pts. (mil)	37	81	1.15
SP density (ε) (m)	1	1	0.2
Cluster dist. (m)	2ε	2ε	1.5ε

[a]All other parameters are independent on the dataset and their values are provided in the text.
[b]S denotes Strathalbyn.
[c]O denotes Oberjettenberg.
[d]G denotes Gubbio.

DTMs by Airborne Research Australia (ARA) using its special research toolkit consisting of a small-footprint full waveform-resolving airborne LiDAR (RIEGL Q680i-S) mounted on a motor-glider, a Diamond Aircraft HK36TTC-ECO Dimona. The LiDAR unit was hard-mounted to a tactical-grade Novatel SPAN IMU/GPS system for determining position, altitude, and orientation. The raw data were georeferenced using ARA internal software, merged into one single point cloud, and then classified using the LAStools suite of utilities [45]. The capability of the motor-glider to fly comparatively low (250 m AGL) and slow (≈35 m/s) along closely spaced flight-lines in the form of a rectangular grid yielded a high density of points (see Table 1). A close-up view of the point cloud for an area around one of the gully edges is shown in Fig. 1(b). The vegetation and non-vegetation points (ground truth) are in green and gray color, respectively.

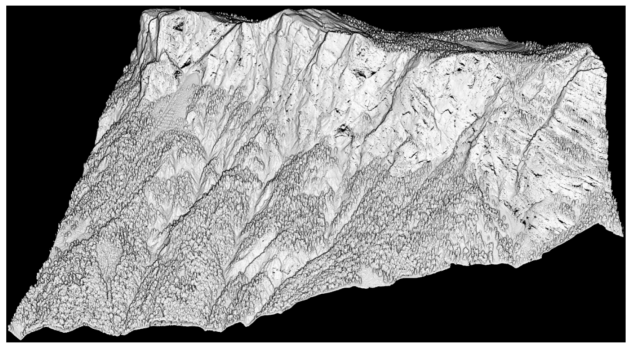

Fig. 2. Complete point cloud in the Oberjettenberg dataset.

B. Oberjettenberg

The second dataset is the point cloud visualized in Fig. 2. It results from some 50 airborne scans in an Alpine area in Southern Germany and will be denoted as Oberjettenberg since it was the name of a close-by settlement. Since the Alps form an overhang in this area, the point coverage below the rock is very scarce, making the data processing challenging. Mathematically, there is no practically feasible way to describe the z coordinate as a function of x and y because points within one square meter sometimes have more than 150 m of elevation difference. Thus, the conventional way to rely on the DSM and DTM does not make sense in this situation, strictly speaking. The main goal of the relevant project, described in more detail in [46], was to create a photo-realistic database and to use it for training and education in areas of disaster management and other quick response applications [47]. For such a reconstruction, it is important to clean the point cloud from points originating from the trees. However, because of the abrupt deviation of angles, trees are easy to confuse with large stones and other rock formations. Especially in the overhang region, deviations of tree trunks' directions from the z axis are almost arbitrary, and, in addition, point densities below the overhang were relatively scarce. The used sensor was the same as for the Strathalbyn dataset, but the average altitude of the sensor platform was around 2800 AGL (see Table 1 for more details).

C. Gubbio

Our third dataset was captured around the medieval wall in Gubbio, a town in central Italy. The walls are remnants of the 14th century; they represent a monumental witness of European history and culture, and should be valued and preserved as such [48]. During many centuries, the substance of the wall was severely affected. Rainfall caused accumulation of soil along the wall, which, in turn, increased the pressure sustained by the structure, leading to a higher risk of collapse, especially in the most fragile parts. Furthermore, alteration in moisture retention and air circulation, in combination with microorganisms developing in inorganic substrates, caused chemical and physical alterations in the materials. Thus, the overreaching goal of the project was to develop automatic algorithms on monitoring and preserving the state of the walls. Starting with a sequence of high-resolution daylight images, an important intermediate goal was to retrieve a 3D point cloud, to identify those points that belong to vegetation, and, finally, either to remove them

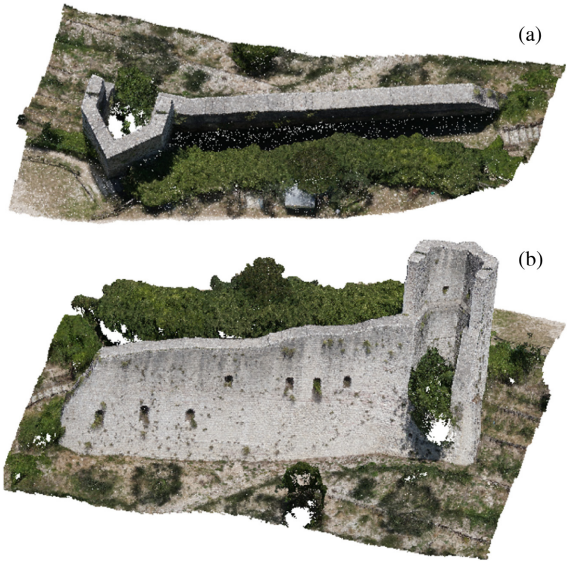

Fig. 3. Two views of the colored point cloud of the Gubbio dataset: (a) top view and (b) side view.

or to replace them by generic models of trees, bushes, and grass areas, thus obtaining a 3D model, which is highly compressed and, at the same time, very detailed in relevant areas [6]. For photogrammetric reconstruction, a typical pipeline containing modules for camera self-calibration, image alignment, and dense reconstruction by means of point matching techniques in images was applied by using commercial software [49]. During dense reconstruction, points are colored using the original images, which means that RGB values are available in addition to the xyz coordinates. For the classification task, it is important to take the existence of the vertical (walls) and even real 3D structures (erosion) into account, as well as the fact that point clouds retrieved from passive sensors are sometimes noisy and sometimes oversmoothed in areas of repetitive patterns, occlusions, and weak textures. The resolution of the 3D point cloud was slightly below 10 cm per pixel with the point density around 3,700 points per m^2. Figure 3 shows two views of the whole point cloud while a labeled subset had to be created for evaluation.

4. METHODOLOGY

A. Superpoints in RANSAC Planes

The SiRP method [6] is based on the assumption that locally terrain points are best described by a plane. Hence, there are two important components recurring through the methods: computation of a model (plane) and restriction to local neighborhoods (patches or clusters of points). We cluster the input 3D points into superpoints using a unique tolerance ε [see Fig. 4(a), left]. The term superpoint is used in a twofold sense: first, it contains a set of U indices from the original point cloud, whereby index sets for two superpoints may overlap, and, second, it has its own 3D coordinates $\mathbf{x}$, computed as coordinate-wise center of this indexed point subset. For every superpoint, a modification of the RANSAC algorithm is run in order to determine the best fitting plane. The essential modification is that, instead of a self-updating number of iterations proposed in [42], we have a

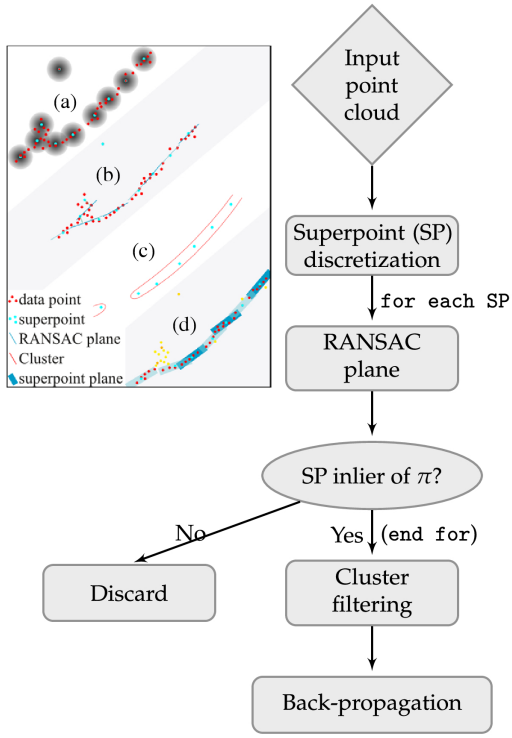

Fig. 4. Main image, workflow diagram of SiRP. Top left, schematic diagram of the SiRP method: (a) unique tolerance approach, (b) extraction of RANSAC planes and pre-filtering superpoints, (c) clustering-based filtering, and (d) backpropagation to the original point cloud to assess the points' class.

fixed number W of random triplets of integers between 1 and U. Thus, the best fitting plane $\mathbf{p}^* = \mathcal{P}_{w^*}$, $w^* \in \{1, \ldots, W\}$ is computed as

$$w^* = \text{argmax}_{\text{sum of colomns}} \left[|\mathcal{P} \cdot \mathcal{X}| < \tau\right], \text{ where}$$

$$\mathcal{P}_w(i) = (-1)^i \det\left(\mathcal{X}_w^{\{i\}}\right), i \in \{1, \ldots, 4\}, \quad \textbf{(1)}$$

whereby [] denotes Iversen brackets, $\mathcal{X}$ is a $4 \times U$ matrix describing the point cloud in homogeneous coordinates, and the 3×3 matrix $\mathcal{X}_w^{\{i\}}$ means selecting the wth triplet of columns and omitting the ith row of $\mathcal{X}$. Furthermore, $\mathcal{P}_w(i)$ is the ith coefficient of the wth plane equation (again using homogeneous coordinates) and, at the same time, the corresponding entry of the $W \times 4$ plane matrix $\mathcal{P}$, and τ is the RANSAC threshold. Using Eq. (1) has the advantage that the computation of W models (matrix $\mathcal{P}$) and best model selection are represented as vector-wise and matrix multiplications, respectively.

These steps are loopless and, therefore, fast and even implementable on GPUs. The superpoints for which $\mathbf{x}$ is not an inlier of the best fitting plane $\mathbf{p}^*$ are discarded, allowing us to remove the middle and lower vegetation layers [as in Fig. 4(b), left]. The remaining superpoints form clusters that can be analyzed automatically (according to the cardinalities) or manually, especially if other classes than vegetation and terrain or man-made object surfaces are relevant. For clustering, a hierarchical method is chosen: neighbors around a point within the radius are computed. For every point $\mathbf{x}$ not yet belonging to a cluster, the neighbors are examined. If two neighbors are lying in two

different clusters, both clusters are merged, the unlabeled neighbors are labeled according to the only remaining cluster, and, if there are no such labeled neighbors, a new cluster containing $\mathbf{x}$ is formed. This process terminates if all points are labeled.

The purpose of clustering for multi-class problems is to compute simple cluster-wise features (planarity) and interactively assign semantic classes. For vegetation separation, it was sufficient to delete superpoints located in too small clusters, as illustrated in the left of Fig. 4(c). Finally, the remaining superpoints are employed to classify the original points using backpropagation. Every point $\mathbf{y}$ is assigned to a superpoint $\mathbf{x}$, which, in turn, has its own plane $\mathbf{p}$ and several (N) nearest neighbors $\mathbf{x}_i$ with planes $\mathbf{p}_i$. Let J denote the number of planes $\mathbf{y}$ is an inlier of. Then, $\mathbf{y}$ is classified as terrain if and only if

$$J > \frac{\lambda_3}{\varepsilon} \cdot N, \tag{2}$$

where λ_3 is the measure of local dispersion of the plane $\mathbf{p}$, i.e., the smallest eigenvalue of the structure tensor over the support set.

As Table 1 shows, the most important parameter for SiRP is the superpoint density ε, introduced in order to accelerate the computations. The reason is that the number U of points in (1) used for RANSAC is strongly reduced and with it the number W of candidate planes. Note that decreasing ε means that RANSAC must be executed more often and its increasing value would lead, besides larger U and W, to a loss of accuracy since the locally uneven surface of the terrain plays a bigger part. The value of ε, set to 0.2 m for the photogrammetric (Gubbio) dataset and to 1.0 m otherwise, depends on the size of the smallest object to be detected and on the size of the terrain patch that approximately constitutes a plane. The clustering distance was about 1.5ε to 2ε in our experiments. The minimum cluster size needed in order to sort out outliers depends on the size of possible off-terrain objects (density of tree crowns) and sudden changes of steepness of the terrain. Taking into account the successive manual preparation of the point cloud, we recommend that this value should not exceed 1000. The RANSAC threshold τ for the point-to-plane residuum in Eq. (1) corresponds to $\varepsilon/2$. However, to find out whether the superpoint is the inlier of its plane, this value was slightly modified:

$$\tau' = \left(1 + 4\varepsilon^{-2}\right)^{-1/2} \xi \approx \tau\xi, \tag{3}$$

whereby ξ is the inlier proportion among points taken for computation of the RANSAC plane, which are those within the search radius of 4ε around the superpoint. The threshold τ' is computed analytically supposing that the superpoint grid is regular.

The last paragraph of this section goes about the alternative to SiRP strategies we have applied. The clustering rule is motivated by the idea to detect first the *possible* superpoints belonging to terrain and then to exclude the superfluous ones. A somewhat opposite strategy would be to detect those superpoints that are *probably* part of the terrain and afterward to extend them. The first step was achieved by checking consistency of every superpoint with the RANSAC planes of neighboring superpoints. The extension step was accomplished by a region growing method, such as [50]. Computationally, this strategy was more expensive than that based on clustering, and, since the performance was comparable, it was discarded. Yet another strategy was inspired by the modification of RANSAC in [51]. Instead of testing all planes with all points, the planes can be first tested with the superpoints, thus allowing a pre-selection. The remaining points are then tested with those planes the superpoint is an inlier of. Despite speed gain, this strategy was discarded as well since, too often, really bad planes remained for vegetation-like superpoints.

B. Evaluation Strategy

To perform the comparison of our algorithm with supervised classification methods, we chose the random forest classifier [52] to be our learning algorithm due to its robustness to several correlated, redundant, or even irrelevant features. We used 20 decision trees in all our experiments. If more than 10 voted for the class vegetation, we classified a point as such; otherwise, class terrain or manmade object surface was assigned. Additional post-processing with Markov random fields (MRFs) [6] has not been applied in order not to distort the results. We recall that SiRP is a rotational-invariant method; therefore, in order to provide a fair basis for comparison, we considered in this work two groups of such generic features. First, we considered the eight following covariance-based features [24]: planarity, omnivariance, linearity, scattering, anisotropy, eigenentropy, curvature change, and sum of the eigenvalues. The second feature set, already referred to in Section 2, is FPFHs [8], computed using multicore processing with five kernels. In contrast to the covariance-based features, the point set must be normalized to have its center of gravity in the origin of the coordinate system. As recommended by [53], concatenating covariance-based features computed with different radii of the spherical/cylindrical neighborhood or varying numbers of neighbors resulting from the kNN algorithm is helpful; hence, we carry out the analogous experiments with FPFHs and consider combinations of both feature types. However, to keep this section compact, we use neighborhoods based on kNN only (in order to guarantee the same number of neighbors for each point), while the range-based searches are omitted.

Classification may be greatly improved if certain properties of the underlying data are taken into account. For example, one could wonder to what extent rotational invariance is relevant in scenes without clear 3D structures, like in Strathalbyn. For this purpose, we defined the δ feature of a point by the difference of its z coordinate and the DTM value at its rounded position. The DTM is computed using a usual algorithm [47]. Similarly, variations of δ in the point neighborhood can be taken into account. In total, there are 16 additional features we used for the Strathalbyn dataset:

- The δ-DTM feature of $\mathbf{x}$ and its intensity (2),
- the means and standard deviations of both features for neighborhoods comprising 10 and 50 nearest neighbors (8),
- The means and standard deviations of 3D distances to 10 and 50 nearest neighbors (4), and
- The return number, current and total (2).

Since the random forest classifier provides the out-of-bag predictor importance, we assess these features with respect to

this measure (OOB). The available ground truth additionally enables us to analyze the dependence on the amount of training examples. That is, we selected three regions with 0.33, 0.54, and 1.21 millions of points lying in geometrically separated areas and performed the evaluation using once the first and then the first and second area as the training set. The third, biggest area was always used for validation.

The Strathalbyn dataset, for which the ground truth is available, is used for a more detailed evaluation, while both other datasets (Oberjettenberg and Gubbio), already analyzed in [6], are used for validation for a broader range of parameters; they represent challenging scenarios, particularly suitable to indicate capabilities and limitations of our method. However, the question about how to estimate the performance of SiRP for the Oberjettenberg and Gubbio datasets still remains open because no ground truth is available there. One obvious strategy is to select two geometrically separated fragments of the data, consisting of some hundreds of thousands of points, and to create the ground truth using a (couple of) very salient characteristics of the point cloud followed by an extensive manual relabeling with Cloud Compare or a similar tool. One fragment is always used for training, and the other is used for validation. In order to avoid correlation-based falsification of results, these salient characteristics should ideally not have much in common with our methodology. For the Oberjettenberg dataset, the results for SiRP were supported by the pulse number and the δ-feature, while for the Gubbio dataset we used the normalized color values [54] because of their ability to highlight color differences in shadows. Thus, in order not to bias the validation, our specific features must not be used for the supervised classification for these two datasets, different from the Strathalbyn dataset, where the ground truth was retrieved in a field campaign.

Partly, the good performance on interactively created ground truth and results of cross-validation with completely different features dissipated our doubts on the expedience of the proposed evaluation method. Finally, the proof of concept is provided by the following test: Using the fact that SiRP does not need training data, we run this algorithm—*with the same parameters as in the Oberjettenberg dataset* (see Table 1)—on the Strathalbyn dataset and measured the accuracy.

5. RESULTS

A. Quantitative Evaluation of SiRP

We depicted in Tables 2–4 the confusion matrices, from which the in remote sensing popular metrics—OA and Cohen's kappa coefficient (κ)—can be derived. The results are more encouraging for both LiDAR-based datasets than for the photogrammetric reconstruction. In case of the Oberjettenberg dataset (Table 3), we performed the computation with different values of ε, and the result was surprisingly stable as the right-hand side of this table shows. For the Strathalbyn dataset (Table 2), the result is surprising for two reasons: first, the set of parameters was the same as that for the Oberjettenberg dataset, and, second, because consideration of the δ parameter brought about only a marginal improvement. Because a coarse

Table 2. Performance of SiRP for the Strathalbyn Dataset without and with Correction Using the Thresholded δ[a]

	SiRP without Correction				SiRP with Correction			
Ref/Pred.	**Ter.**	**Veg.**	**Total**	**Prec.**	**Ter.**	**Veg.**	**Total**	**Prec.**
Ter.	81.19	0.85	82.04	98.97	81.58	0.46	82.03	99.44
Veg.	3.20	14.76	17.96	82.18	3.32	14.64	17.97	81.50
Total	84.39	15.61	**37,149,534**	OA = 95.95	84.90	15.10	**37,149,534**	OA = 96.22
Recall	96.21	94.57	EA = 72.03	κ = 85.52	96.09	96.97	EA = 72.36	κ = 86.32

[a]In Tables 2–4, all entries are given in percent except the bold one showing the total number of points (100%). EA means expected accuracy and is essential to compute κ.

Table 3. Performance of SiRP for the Oberjettenberg Dataset with Different Values of ε

	SiRP with $\varepsilon = 1$				SiRP with $\varepsilon = 1.6$			
Ref/Pred.	**Ter.**	**Veg.**	**Total**	**Prec.**	**Ter.**	**Veg.**	**Total**	**Prec.**
Ter.	90.67	2.25	92.92	92.92	90.62	2.30	92.92	92.92
Veg.	0.74	6.34	7.08	89.52	0.74	6.34	7.08	89.55
Total	91.41	8.59	**549,066**	OA = 97.01	91.36	8.64	**549,066**	OA = 96.96
Recall	91.41	73.78	EA = 85.55	κ = 79.28	91.36	73.80	EA = 85.50	κ = 79.03

Table 4. Performance of SiRP for the Gubbio Dataset without and with Correction Using the RGB Values of Points

	SiRP without Correction				SiRP with Correction			
Ref/Pred.	**Ter.**	**Veg.**	**Total**	**Prec.**	**Ter.**	**Veg.**	**Total**	**Prec.**
Ter.	27.46	0.92	28.38	96.76	26.81	1.57	28.38	94.48
Veg.	24.26	47.36	71.62	66.13	21.59	50.03	71.62	69.85
Total	51.72	48.28	**491,284**	OA = 74.82	48.40	51.60	**491,284**	OA = 76.84
Recall	53.98	98.10	EA = 49.26	κ = 50.38	96.09	96.97	EA = 50.69	κ = 53.02

interpolation was employed for the DTM computation, the DTM is oversmoothed and, therefore, is not suitable to distinguish between small elevations in the terrain and grass regions having insignificant heights. Nevertheless, thresholding the δ-feature can be applied to detect *gross* errors and to obtain a fair, though inferior to SiRP, performance: the OA and κ values were 90.8% and 69.6%, respectively. The Gubbio dataset (Table 4) led to the least accurate results. Even though the values on precision are encouraging and even better than for the Oberjettenberg dataset, the main source of errors is that the vegetation is arranged in piecewise smooth surfaces, as mentioned in Section 3. Finally, because of shadows, which affected both wall structure and vegetation, the improvement of the pure SiRP in OA and κ was only 2% and 2.6%, respectively, after applying the correction based on the normalized green–red difference index (NGRDI).

B. Quantitative Evaluation of Supervised Classification

Even though supervised classification results do not represent the main contribution of this work, they are helpful for the quantitative comparison with the proposed method and will play an important part for the multi-label classification in the future. Therefore, the main findings are briefly presented in the following, while Tables 5–11 with results for different feature and parameter settings. All datasets have in common that FPFHs outperform the covariance-based features not only in the peak values of OA and κ, but also in robustness (broader range for results of approximately same quality) and in computing time. Concatenation of two FPFHs allows for a minimally improved performance; more than two FPFHs may even lead to a degradation (not shown in tables). This confirms that single FPFHs are already descriptive enough and their concatenation does not bring a significant improvement [28]. Concatenation of two sets of covariance-based features usually allows us to improve the performance slightly; however, also here, more than two tend to produce overfitted models. The advantages

Table 5. Performance of FPFHs on the Strathalbyn Dataset (Left Table)[a]

	FPFHs only (33)				FPFHs and Specific Features (49)			
	less t. d.		more t. d.		less t. d.		more t. d.	
N	OA	κ	OA	κ	OA	κ	OA	κ
10 (5)	86.96	73.48	87.03	73.69	92.95	85.82	**94.10**	**88.17**
15 (10)	87.17	73.97	87.22	74.10	92.92	85.75	94.09	88.15
20 (5)	87.30	74.26	87.35	74.40	**92.96**	**85.84**	94.02	88.00
30 (5)	**87.40**	**74.50**	**87.58**	**74.90**	92.86	85.62	94.09	88.15
50 (5)	86.96	73.66	87.19	74.17	92.69	85.28	94.09	88.14
100 (25)	84.78	69.39	85.15	70.12	92.78	85.46	93.77	87.51
150 (50)	81.68	63.35	83.24	66.33	92.61	85.12	93.65	87.26
200 (25)	79.28	58.54	81.09	62.06	92.57	85.04	93.43	86.84
400 (50)	76.52	52.57	77.90	55.57	92.40	84.69	93.20	86.37
500 (25)	76.29	51.71	76.89	53.41	91.96	83.79	93.24	86.43
800 (100)	75.18	49.16	76.19	51.56	92.00	83.87	93.11	86.17

[a]Here and elsewhere, all entries are given in percent. The number in parenthesis after configuration, f, denotes the total number of features. That is, $f = 33$ in case of FPFHs only and if using specific features (right table), f is incremented by 16. All tests were carried out with two different amounts on training data (t. d.) (see Section 4.B). The configurations achieving the best performance are highlighted. We denote the number of neighbors used for determination of FPFHs by N, while the number in brackets (in the first row of the table) refers to neighbors considered for normal vector computation and is much less relevant.

Table 6. Performance of Covariance-Based Features[a] on the Strathalbyn Dataset[b]

	CFs Only (8)				CFs and Specific Features (24)			
	less t. d.		more t. d.		less t. d.		more t. d.	
N	OA	κ	OA	κ	OA	κ	OA	κ
5	81.67	63.08	81.31	62.52	**93.60**	**87.09**	**94.59**	**89.12**
10	**83.15**	**65.70**	**83.00**	**65.47**	93.47	86.84	94.50	88.95
25	70.52	39.55	70.05	39.08	93.47	86.86	94.44	88.84
50	66.63	31.30	66.66	32.29	93.21	86.32	94.31	88.57
100	66.00	29.90	66.21	31.18	92.81	85.51	94.22	88.39
200	64.73	27.52	65.49	29.67	93.16	86.22	94.05	88.05
300	64.08	26.17	65.07	28.77	93.02	85.95	94.07	88.10
500	62.88	23.88	64.13	26.89	92.88	85.65	94.09	88.13
1000	60.88	20.17	62.55	23.84	93.02	85.94	93.94	87.82

[a]CFs, left table.
[b]As explained in Section 4.B, $f = 8$. In case of using specific features (right table), f is incremented by 16.

Table 7. Performance of Multiple Sets of CFs and FPFHs on the Strathalbyn Dataset as Well as Concatenation of Both Feature Sets[a]

Combination of CFs (16)						Combination of FPFHs (66)						CFs and FPFHs (41)			
		less t. d.		more t. d.				less t. d.		more t. d.				less t. d.	
N_1	N_2	OA	κ	OA	κ	N_1	N_2	OA	κ	OA	κ	N_1	N_2	OA	κ
5	10	**85.5**	**70.5**	**85.3**	**70.2**	10 (5)	20 (5)	88.1	75.9	88.3	76.2	5	10 (5)	87.4	74.4
5	25	83.3	66.2	83.4	66.5	10 (5)	30 (5)	**88.5**	76.7	**88.7**	**77.1**	5	30 (10)	**88.1**	**75.8**
5	100	82.1	63.4	83.1	65.8	20 (5)	30 (5)	88.1	75.9	88.2	76.2	25	10 (5)	87.2	74.1
10	25	84.0	67.5	84.2	67.8	10 (5)	30 (10)	**88.5**	**76.8**	**88.7**	**77.1**	25	30 (10)	87.6	74.9
25	50	72.7	44.0	73.0	45.0	10 (5)	200 (50)	87.9	75.5	87.8	75.4				
50	100	68.9	36.0	69.7	38.4	10 (5)	400 (50)	87.2	74.0	87.4	74.4				
50	500	69.5	37.9	70.9	40.7	100 (50)	800 (100)	83.9	67.5	84.7	69.3				

[a]Because there are 33 FPFH-based features and 8 covariance based features, f takes on values 16, 66, and, respectively, 41 in the three main configurations of the table. Not documented in the table is the case of concatenation of FPFHs and covariance-based features with more training data, where the performance was similar. Also, if specific features are used ($f = 57$), OA remains under 95%.

Table 8. Performance of FPFHs and Their Combinations on the Oberjettenberg Dataset, Whereby the Number of Points Needed to Compute the Normal Vector Is Put in Parenthesis[a]

FPFHs (33)			Combination of FPFHs (66)			
N	OA	κ	N_1	N_2	OA	κ
50 (50)	7.1	0.0	100 (50)	500 (50)	41.6	7.00
100 (50)	94.5	62.2	500 (50)	500 (150)	91.9	49.7
150 (50)	**94.9**	53.9	1000 (50)	500 (150)	93.2	53.2
250 (50)	86.9	40.2	500 (50)	1000 (50)	**94.5**	58.8
500 (50)	94.7	**58.2**	500 (150)	100 (150)	91.7	50
750 (50)	94.1	57.2	500 (50)	1000 (150)	94.3	**59.7**
1000 (50)	93.1	52.5	1000 (50)	500 (150)	94.0	58.8

[a]For more details, please see the caption of Table 5.

Table 9. Performance of Covariance-Based Features and Their COMBINATIONS, Including Those with FPFHs, on the Oberjettenberg Dataset[a]

CFs (8)			Combination of CFs (16)				Combination of CFs and FPFHs (41)			
N	OA	κ	N_1	N_2	OA	κ	N_1	N_2	OA	κ
50	92.4	54.2	50	100	93.7	59.1	50	100 (50)	22.2	2.6
100	**93.3**	**56.2**	50	150	94.1	60.5	50	500 (150)	**95.0**	**65.4**
150	93.0	54.0	50	250	**94.2**	**60.7**	100	100 (50)	**95.0**	64.9
250	92.1	48.6	50	500	93.8	59.1	100	500 (150)	94.9	63.6
500	90.4	40.2	100	150	93.7	57.7				
750	89.2	34.9	100	250	94.0	58.7				
1000	87.4	29.9	150	500	93.9	57.1				

[a]For more details, please see the caption of Tables 5–7.

Table 10. Performance of FPFHs and Their Combinations on the Gubbio Dataset[a]

FPFHs (33)			Combination of FPFHs (66)			
N	OA	κ	N_1	N_2	OA	κ
50 (50)	73.0	18.0	500 (50)	1000 (150)	78.8	55.0
100 (50)	74.5	40.3	500 (50)	500 (150)	83.8	62.8
150 (50)	82.2	53.4	1000 (50)	1000 (150)	80.9	58.7
250 (50)	80.6	57.2	150 (150)	500 (50)	77.9	36.5
500 (50)	**83.5**	**62.6**	100 (150)	500 (150)	79.2	54.6
750 (50)	81.4	59.2	1000 (50)	150 (150)	50.0	52.5
1000 (50)	77.0	51.4	150 (50)	1000 (150)	**84.6**	**65.2**

[a]For more details, please see the caption of Table 5.

Table 11. Performance of Covariance-Based Features[a] and Their COMBINATIONS, Including Those with FPFHs, on the Gubbio Dataset[b]

CFs (8)			Combination of CFs (16)				Combination of CFs and FPFHs (41)			
N	OA	κ	N_1	N_2	OA	κ	N_1	N_2	OA	κ
50	57.4	25.5	50	100	**59.7**	**28.3**	150	1000 (150)	72.9	45.2
100	58.4	26.9	50	150	59.6	28.1	500	500 (50)	70.6	21.3
150	**58.6**	**27.1**	50	250	58.1	26.2	500	1000 (150)	65.5	35.0
250	57.3	25.2	50	500	57.0	24.8	150	500 (50)	**79.0**	**50.4**
500	56.1	23.3	100	150	59.6	28.1				
750	55.8	22.8	100	250	58.3	26.5				
1000	55.9	22.8	150	500	57.1	25.0				

[a]CFs.
[b]For more details, please see the caption of Tables 5–7.

of covariance-based features are a better interpretability, lower computation time for small neighborhoods, and an improved performance for concatenated feature sets.

As for dataset-dependent differences, in the Strathalbyn dataset, using more training data slows down the decay of accuracy for covariance-based features and is rather insignificant for FPFHs. It is remarkable that, using our 16 additional features from the previous section, the highest scores among single feature types could be obtained. This is particularly visible in the right-hand part of Table 6, but, in general, an interpretation that specific features always influence the performance positively is valid. The analysis of specific features' importance showed a clear choice of the δ value of the point itself, thus confirming our findings from before and manifesting the fact that working with 2.5D point clouds is easier than in 3D environments. The next important features were standard deviations of intensity, distances, and δ values (mostly in small neighborhoods). The least important feature was the total number of returns, preceded, in the majority of cases, by the mean value of δ within a large neighborhood. In the Oberjettenberg dataset, the differences in performance analysis between FPFHs and covariance-based features are the least significant. However, the neighborhood size N required to obtain the best results is by far larger than for the previous dataset. This represents a problem for the covariance-based features since the increase on computational time with a growing neighborhood size is linear. The Gubbio dataset yielded the lowest performance in terms of OA and κ. Moreover, in the peak results obtained for the Gubbio dataset, the FPFHs outperform the covariance-based features by far. In contrast to Strathlbyn (5%) and Oberjettenberg (2%), the peak results differ in some 25%. This emphasizes the necessity of semantically richer features in order to classify oversmoothed vegetation regions in photogrammetric datasets and insufficiency of geometric features alone.

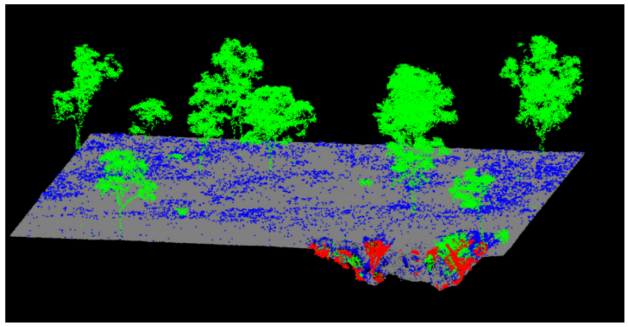

Fig. 5. Performance of SiRP visualized for the Strathalbyn data. Here and further, by green and gray points, we denote the correctly classified parts of vegetation and non-vegetation (terrain and wall for Gubbio, otherwise terrain only), respectively. Blue are points spuriously classified as terrain and red as vegetation.

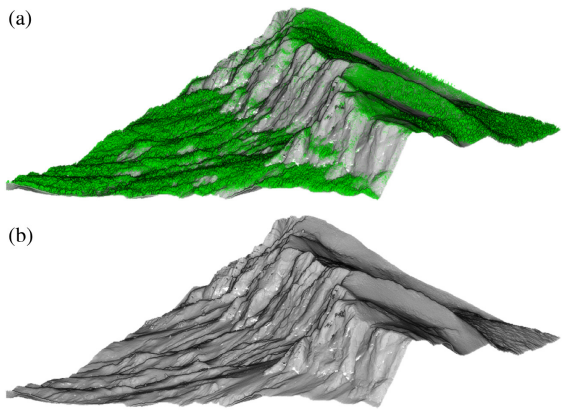

Fig. 6. Performance of SiRP visualized for the whole Oberjettenberg dataset: (a) with trees colored in green color and (b) without trees.

C. Qualitative Results

Figures 5–10 illustrate typical situations for point filtering and classification. For the Strathalbyn dataset with available ground truth, we can additionally plot the correctly and incorrectly classified points. Intuitively, it seems that it is significantly easier to obtain good results on separation of terrain and vegetation for this dataset than for the Gubbio dataset with its vertical walls and the Oberjettenberg dataset with its overhangs. The result shown in Fig. 7(d) confirms the observation that interactive training data acquisition (in this case, discarding points belonging to a few trees in the bottom right part of the figure, specified by red circles) starting at the results of SiRP is anything but a mammoth task. For the Gubbio dataset, the result is not as good, and more post-processing should take place. Several bushes remain undetected, as shown in Fig. 8(a) by blue ellipses. Considering an additional simple feature, like NGRDI, makes

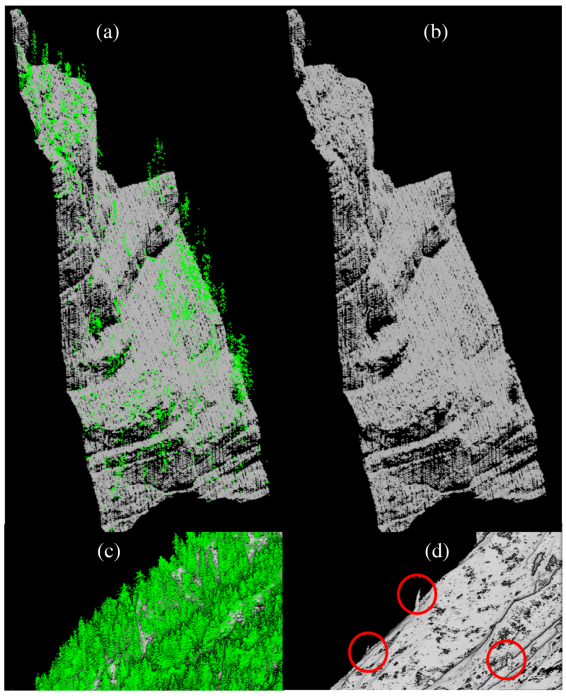

Fig. 7. Performance of SiRP visualized for the Oberjettenberg data. (a), (b) Top row, a steep slope with a sparse vegetation; (c), (d) bottom row, a flatter area with a dense forest. (a), (c) Left, with trees colored in green color; (b), (d) right, without trees.

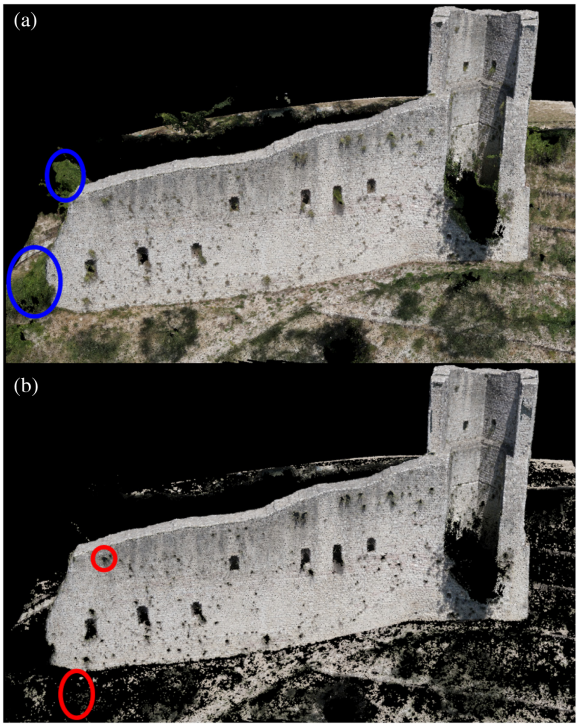

Fig. 8. Performance of SiRP, (a) without and (b) with color values taken into account, visualized for the Gubbio data. We denoted several mis-classifications by ellipses: blue and red ellipses are traced around points spuriously classified as terrain and vegetation, respectively.

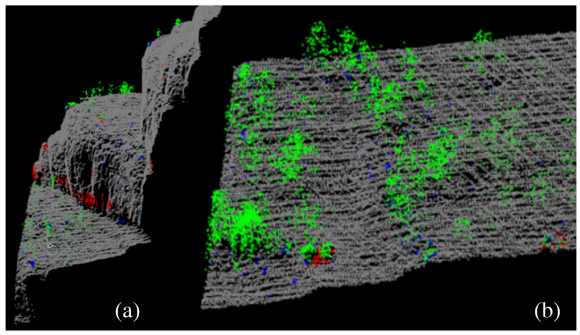

Fig. 9. Performance of FPFHs for the Oberjettenberg dataset: part of an overhang with (a) a sparse vegetation and (b) a flatter area with some more trees and shrubs. For colors, refer to Fig. 5.

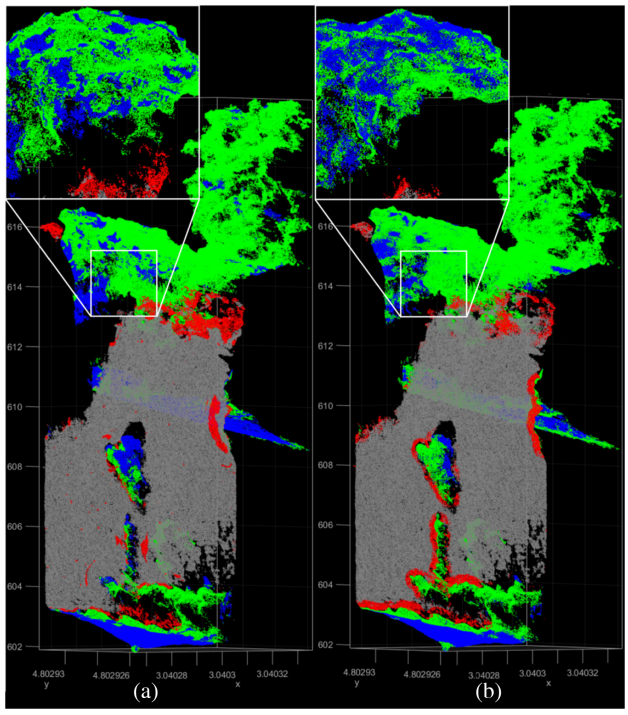

Fig. 10. Comparison in performance of (a) FPFHs and (b) covariance-based features for the Gubbio dataset. On the top, enlarged fragment, which helps to illustrate the differences. For colors, refer to Fig. 5.

the task of training data acquisition more manageable. For a rigorous choice of the NGRDI threshold, many false alarms are eliminated, as shown in Fig. 8(b). However, as indicated by red ellipses in Fig. 8(b), some planar green grass regions are often spuriously deleted. Finally, in shadow areas, the performance of NGRDI is limited.

Concerning supervised classification, Fig. 9 shows that the steep slope in Fig. 9(a) is less a problem for FPFHs than a strongly varying point density, but, nevertheless, the vast majority of points have been classified correctly, similarly to Fig. 9(b). In the areas occupied by dense forests, FPFHs occasionally assign points atop the tree canopies to an incorrect class.

Fortunately, these points lie mostly isolated; thus, the successive application of non-local optimization on MRFs is likely to correct these mis-classifications [6]. The covariance-based features are usually able to classify tops of tree crowns correctly, but have problems in regions with low vegetation. For the Gubbio dataset, shown in Fig. 10, the wrongly classified points sometimes form clusters, visible in particular, on the margin of the dataset as well as in the regions along borders between classes. Because this problem appears less obvious in Fig. 10(a) than in Fig. 10(b), setting it right using MRFs is expected to be more successful for the FPFH-based than for the covariance-based features.

6. DISCUSSION

A. Final Remarks on the Potential and Limitations of SiRP

The results from the previous section confirm a good and stable performance of the proposed method. For both LiDAR datasets, the small deviations from the ground truth are caused by the fact that the points located in trenches and other hollows on the ground are not consistent with the dominant planes derived by RANSAC and, thus, are incorrectly classified as vegetation. If the plane normal vectors were *oriented*, this problem could be mitigated because a more generous threshold toward the interior of the surface would allow a better classification. However, our RANSAC planes are not oriented, and imposing consistent orientation is a non-trivial task [55]. In the opposite direction, there were some grass tufts that can easily be confused with ground (Fig. 5). For a photogrammetric dataset, the results of SiRP are slightly more encouraging after being combined with color indices, as Table 4 shows. The sources of errors here were, on the one hand, porosity and erosion of the wall rock (which is actually one reason why the whole project was instituted) and, on the other hand, the oversmoothed tree crowns. As a result of the photogrammetric procedure for dense reconstruction, they tend to be approximated by piecewise smooth surfaces, similar to ground and wall. Then, occasional confusion with these classes seems inevitable, and using color values—and not just color indices like NGRDI—is indispensable. Fortunately, the superpoint clustering step is particularly helpful because it allows cutting away (possibly) large parts of data assigned to an incorrect class using an interactive tool, such as Cloud Compare.

B. Supervised Classification

FPFHs allow for a higher accuracy and an increased stability with respect to neighborhood size (the parameter for normal vector computation, shown in parenthesis in Tables 5 and 7–11, is basically irrelevant), which, in combination with a more constant computation time, makes these features superior to the covariance-based features, especially for the photogrammetric point cloud. The wrongly classified points sometimes form clusters, which makes a successive correction using MRFs less successful. In fact, starting at covariance-based features, the improvement for the photogrammetric point cloud leads to almost the same accuracy as for the LiDAR dataset while, for that from the photogrammetric reconstruction, using *a priori* probabilities from FPFHs led to 10% points surplus on accuracy compared to covariance-based features (results available in [6]).

For the dataset with annotated ground truth, we also tested our specific features. Here, our finding was that they were superior to the generic, rotationally invariant features (at least for the considered dataset). Overall, relatively high accuracy obtained for cross-validation of disjunctive point subclouds for all datasets points not only to a good quality of the training data, but also means that the employed features are sufficiently descriptive to solve the addressed classification task. Most importantly, at least in both LiDAR-based datasets and for only two classes to be distinguished, our proposed SiRP approach achieved a superior performance to all methods involving supervised classification. This is an encouraging fact, and it must be verified in future work whether this method can stand the comparisons with more complicated datasets with annotated ground truth.

C. Comparison with State of the Art

One may argue that rotational invariance of SiRP is an overkill in the majority of cases and relative elevation alone, derived, e.g., by a state-of-the-art method [47] should suffice for point filtering. However, even for the relatively simple Strathalbyn dataset, we could see that, doing so, the OA with 90.8% is clearly below the 95.6% achieved by the proposed method. Comparison with further methods on point filtering turned out to be difficult for two reasons: First, because they use other datasets, bearing different application scenarios in mind, and their utilized algorithms have a different parameter setup. The second reason is that also the evaluation metrics differ: most of the methods on ground point filtering output the result of DTM interpolation and, with it, the RMSE error. Confusion matrices in this context are most popular for multi-class problems, such as semantic segmentation addressed with the ISPRS benchmark datasets for Vaihingen [40] and Potsdam [56]. But problems with more than two classes need training data, with a few exceptions, while SiRP does not need any (we performed multi-class semantic segmentation on the Vaihingen dataset with our generic features; the results are available on request). To the best of the authors' knowledge, only one publication contains information on accuracy of ground point filtering in 3D (though several others [30,57–59] did in 2D): in his Ph.D. thesis [60], Sithole analyzed the Vaihingen dataset, after subdividing it into 15 samples. Assuming that the number of points in every sample was almost the same, the average OA would be 94.3%, which is below that of SiRP for Strathalbyn and Oberjettenberg datasets. While SiRP slightly outperforms machine learning approaches using shallow generic features for LiDAR data and lies between FPFHs and covariance-based matrices in the case of photogrammetric point clouds, the approach of [30] is based on deep learning in 2D and produces slightly more accurate results (98.78%) for LiDAR points. However, as mentioned above, comparability of approaches in this context is questionable due to the differences between the datasets.

D. Computing Time

SiRP allows processing some 6700/7800 points or, alternatively, 450/53 superpoints (in case of Strathalbyn/Gubbio) per second (recall that Oberjettenberg was processed with the same set of parameters as Strathalbyn). To do this, a MATLAB implementation using the parallel processing toolbox was applied. For FPFHs, using a C++ implementation available in the Point Cloud Library [61] on a five-core PC required a time under 1 min for the test point cloud consisting of almost 500,000 points. We can conclude that a single-core implementation would need some 5 min. Thus, both best-proven measures require the same order of magnitude for computing time. The factor for covariance-based features was around 0.47 for small and 1.4 and more for large neighborhoods compared to FPFHs. Turning back to SiRP, the time needed to process the whole Oberjettenberg dataset and to achieve the result from Fig. 6 was a little bit more than 3 h. For the diminished, but verified, point clouds, the time-per-run was between 1 and 5 min.

7. CONCLUSION AND FUTURE WORK

In this paper, the SiRP method for interpreting airborne 3D point clouds in terms of a separation of vegetation from the terrain or manmade structures was presented and compared with approaches for supervised classification. Both qualitative and quantitative results were encouraging. Even though no training examples are required for SiRP, accuracy values exceeded 96% for laser point clouds. Even for photogrammetric reconstruction, an OA of some 76% was achieved. Contrary to LiDAR-based datasets, the results are below those of supervised classification where almost 85% were achieved using a pair of FPFHs. This allows us to reason that the potential of SiRP in terms of its accuracy and efficiency is substantial since both of its crucial components (superpoints and RANSAC) may be modified. In [39], clustering is carried out using not only the spatial coordinates, but also other features. Thus, also in our case a better consideration of, e.g., RGB values of points would help to create better clusters. As for RANSAC, numerous modifications exist in the literature; we refer, e.g., to RANSAC with $T_{d,d}$ test, mentioned before, MSAC, or to a more general review by [62]. We concentrated on planar patches because the efficient version of RANSAC was employed for plane fitting, but as [63] pointed out, a choice of other shape primitives would be possible, too.

With respect to the supervised classification using FPFHs and covariance-based features, the former ones seem to be superior for all presented datasets, especially for the photogrammetric one. Combination with specific features, such as absolute elevation and its statistical moments, is also beneficial because the aforementioned feature types are rotationally invariant.

The sense and purpose of the feature descriptors is to be applied to multi-class problems, which has not been accomplished in this work and must be subject of future research. Fortunately, the way the preparation of ground truth for the Oberjettenberg and Gubbio datasets was carried out already provides a concept for the future work for multi-class problems: after separation of vegetation using SiRP, remaining points can be re-clustered within a semi-supervised framework using additional features (NGRDI, planarity, etc.) and, finally, classified in a supervised way using generic feature sets. This will allow further development of SiRP and supervised classification to go hand in hand in the future.

Acknowledgment. The Gubbio dataset was available thanks to the European Union's project HERACLES (Grant Agreement 700395). For providing the ground truth for the Strathalbyn dataset, we wish to thank Prof. Andrew Brooks (Griffith University Australia). We further thank Lukas Lucks (IOSB) for preparing the Gubbio dataset and our student assistant Marko Hecht (IOSB) who helped a lot with FPFHs. We finally thank the developers of freely available software, in particular, Cloud Compare and Point Cloud Library, who saved us a great deal of work.

Disclosures. The authors declare no conflicts of interest.

Data Availability. Data underlying the results presented in this paper are not publicly available at this time but may be obtained from the authors upon reasonable request.

REFERENCES

1. D. H. Evans, R. J. Fletcher, C. Pottier, J.-B. Chevance, D. Soutif, B. S. Tan, S. Im, D. Ea, T. Tin, S. Kim, C. Cromarty, S. De Greef, K. Hanus, P. Bâty, R. Kuszinger, I. Shimoda, and G. Boornazian, "Uncovering archaeological landscapes at Angkor using lidar," Proc. Natl. Acad. Sci. USA **110**, 12595–12600 (2013).
2. http://www.heracles-project.eu/.
3. D. Bulatov, L. Lucks, J. Moßgraber, M. Pohl, P. Solbrig, G. Murchio, and G. Padeletti, "HERACLES: EU-backed multinational project on cultural heritage preservation," Proc. SPIE **10790**, 107900D (2018).
4. F. Lafarge and C. Mallet, "Creating large-scale city models from 3D-point clouds: a robust approach with hybrid representation," Int. J. Comput. Vision **99**, 69–85 (2012).
5. E. Grilli, F. Menna, and F. Remondino, "A review of point clouds segmentation and classification algorithms," Int. Arch. Photogramm. Rem. Sens. Spat. Inf. Sci. **XLII-2/W3**, 339–344 (2017).
6. D. Bulatov, D. Stütz, L. Lucks, and M. Weinmann, "Superpoints in RANSAC planes: a new approach for ground surface extraction exemplified on point classification and context-aware reconstruction," in *15th International Conference on Computer Graphics Theory and Applications* (2020), pp. 25–37.
7. H.-G. Maas and G. Vosselman, "Two algorithms for extracting building models from raw laser altimetry data," ISPRS J. Photogramm. Remote Sens. **54**, 153–163 (1999).
8. R. B. Rusu, N. Blodow, and M. Beetz, "Fast point feature histograms (FPFH) for 3D registration," in *IEEE International Conference on Robotics and Automation* (IEEE, 2009), pp. 3212–3217.
9. K. Kraus and N. Pfeifer, "Determination of terrain models in wooded areas with airborne laser scanner data," ISPRS J. Photogramm. Rem. Sens. **53**, 193–203 (1998).
10. G. Vosselman, "Slope based filtering of laser altimetry data," Int. Arch. Photogramm. Rem. Sens. Spat. Inf. Sci. **33**, 935–942 (2000).
11. G. Sithole and G. Vosselman, "Filtering of laser altimetry data using a slope adaptive filter," in *International Archives of the Photogrammetry and Remote Sensing* (2001), Vol. **34**, pp, 203–210.
12. M. A. Brovelli, M. Cannata, and U. Longoni, "Managing and processing LIDAR data within GRASS," in *GRASS Users Conference* (2002), Vol. **29**.
13. G. Sithole and G. Vosselman, "Experimental comparison of filter algorithms for bare-Earth extraction from airborne laser scanning point clouds," ISPRS J. Photogramm. Remote Sens. **59**, 85–101 (2004).
14. K. Zhang, S.-C. Chen, D. Whitman, M.-L. Shyu, J. Yan, and C. Zhang, "A progressive morphological filter for removing nonground measurements from airborne lidar data," IEEE Trans. Geosci. Remote Sens. **41**, 872–882 (2003).
15. D. Mongus and B. Žalik, "Parameter-free ground filtering of LiDAR data for automatic DTM generation," ISPRS J. Photogramm. Remote Sens. **67**, 1–12 (2012).
16. M. Elmqvist, E. Jungert, F. Lantz, A. Persson, and U. Söderman, "Terrain modelling and analysis using laser scanner data," in

International Archives of the Photogrammetry and Remote Sensing (2001), Vol **34**, pp. 219–226.
17. R. Perko, H. Raggam, K.-H. Gutjahr, and M. Schardt, "Advanced DTM generation from very high resolution satellite stereo images," ISPRS Ann. Photogramm. Remote Sens. Spatial Inf. Sci. **II-3/W4**, 165–172 (2015).
18. Y. A. Mousa, P. Helmholz, D. Belton, and D. Bulatov, "Building detection and regularisation using DSM and imagery information," Photogramm. Rec. **34**, 85–107 (2019).
19. D. Bulatov, P. Wernerus, and H. Gross, "On applications of sequential multi-view dense reconstruction from aerial images," in *Proceedings of the 1st International Conference on Pattern Recognition Applications and Methods* (2012), pp. 275–280.
20. Z. Chen, B. Gao, and B. Devereux, "State-of-the-art: DTM generation using airborne LIDAR data," MDPI Sens. **17**, 150 (2017).
21. D. Bulatov and J. E. Lavery, "Reconstruction and texturing of 3D urban terrain from uncalibrated monocular images using L_1 splines," Photogramm. Eng. Remote Sens. **76**, 439–449 (2010).
22. B. Guo, X. Huang, F. Zhang, and G. Sohn, "Classification of airborne laser scanning data using JointBoost," ISPRS J. Photogramm. Remote Sens. **100**, 71–83 (2015).
23. H. Gross and U. Thoennessen, "Extraction of lines from laser point clouds," in *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences* (2006), Vol. **36**, pp. 86–91.
24. M. Weinmann, *Reconstruction and Analysis of 3D Scenes: From Irregularly Distributed 3D Points to Object Classes*, 1st ed. (Springer, 2016).
25. 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. **66**, S71–S84 (2011).
26. N. Chehata, L. Guo, and C. Mallet, "Airborne lidar feature selection for urban classification using random forests," in *International Archives of Photogrammetry and Remote Sensing* (2009), Vol. **38**, pp. 2007–2012.
27. X.-F. Hana, J. S. Jin, J. Xie, M.-J. Wang, and W. Jiang, "A comprehensive review of 3D point cloud descriptors," arXiv:1802.02297 (2018).
28. R. B. Rusu, Z. C. Marton, N. Blodow, and M. Beetz, "Persistent point feature histograms for 3D point clouds," in *International Conference Intelligent Autonomous System (IAS-10)* (2008), pp. 119–128.
29. L. Winiwarter, G. Mandlburger, S. Schmohl, and N. Pfeifer, "Classification of ALS point clouds using end-to-end deep learning," PFG–J. Photogramm. Remote Sens. Geoinf. Sci. **87**, 75–90 (2019).
30. X. Hu and Y. Yuan, "Deep-learning-based classification for DTM extraction from ALS point cloud," MDPI Remote Sens. **8**, 730 (2016).
31. H. Arief, G.-H. Strand, H. Tveite, and U. Indahl, "Land cover segmentation of airborne LiDAR data using stochastic atrous network," MDPI Rem. Sens. **10**, 973 (2018).
32. N. Audebert, B. Le Saux, and S. Lefèvre, "Semantic segmentation of earth observation data using multimodal and multi-scale deep networks," in *Asian Conference on Computer Vision* (Springer, 2016), pp. 180–196.
33. A. Boulch, B. Le Saux, and N. Audebert, "Unstructured point cloud semantic labeling using deep segmentation networks," in *Eurographics Workshop on 3D Object Retrieval (3DOR)* (2017).
34. S. Schmohl and U. Sörgel, "Submanifold sparse convolutional networks for semantic segmentation of large-scale ALS point clouds," ISPRS Ann. Photogramm. Remote Sens. Spatial Inf. Sci. **IV-2/W5**, 77–84 (2019).
35. C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "PointNet: Deep learning on point sets for 3D classification and segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2017), pp. 652–660.
36. C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "PointNet++: Deep hierarchical feature learning on point sets in a metric space," in *Advances in Neural Information Processing Systems* (2017), pp. 5099–5108.
37. S. Jin, Y. Su, X. Zhao, T. Hu, and Q. Guo, "A point-based fully convolutional neural network for airborne lidar ground point filtering in forested environments," IEEE J. Sel. Top. Appl. Earth Observ. Remote Sensing **13**, 3958–3974 (2020).
38. X. Ye, J. Li, H. Huang, L. Du, and X. Zhang, "3D recurrent neural networks with context fusion for point cloud semantic segmentation," in *Proceedings of the European Conference on Computer Vision (ECCV)* (2018), pp. 403–417.
39. L. Landrieu and M. Simonovsky, "Large-scale point cloud semantic segmentation with superpoint graphs," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2018), pp. 4558–4567.
40. M. Cramer, "The DGPF-test on digital airborne camera evaluation—overview and test design," PFG–J. Photogramm. Remote Sens. Geoinf. **2010**, 73–82 (2010).
41. T. Hackel, J.-D. Wegner, and K. Schindler, "Fast semantic segmentation of 3D point clouds with strongly varying density," ISPRS Ann. Photogramm. Rem. Sens. Spatial Inf. Sci. **III-3**, 177–184 (2016).
42. M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," Commun. ACM **24**, 381–395 (1981).
43. A. P. Brooks, J. Spencer, N. Doriean, T. J. Pietsch, and J. Hacker, "A comparison of methods for measuring water quality improvements from gully rehabilitation in great barrier reef catchments," in *Proceedings of the 9th Australian Stream Management Conference* (2018), pp. 567–574.
44. A. P. Brooks, J. Shellberg, J. Knight, and J. Spencer, "Alluvial gully erosion: an example from the Mitchell fluvial megafan, Queensland, Australia," Earth Surf. Processes Landforms **34**, 1951–1969 (2009).
45. https://www.rapidlasso.com.
46. G. Häufel, D. Bulatov, and P. Solbrig, "Sensor data fusion for textured reconstruction and virtual representation of alpine scenes," Proc. SPIE **10428**, 1042805 (2017).
47. D. Bulatov, G. Häufel, J. Meidow, M. Pohl, P. Solbrig, and P. Wernerus, "Context-based automatic reconstruction and texturing of 3D urban terrain for quick-response tasks," ISPRS J. Photogramm. Remote Sens. **93**, 157–170 (2014).
48. F. Carvalho, A. Lopes, A. Curulli, T. da Silva, M. Lima, G. Montesperelli, S. Ronca, G. Padeletti, and J. Veiga, "The case study of the medieval town walls of Gubbio in Italy: first results on the characterization of mortars and binders," MDPI Herit. **1**, 468–478 (2018).
49. https://www.agisoft.com.
50. T. Rabbani, F. Van Den Heuvel, and G. Vosselman, "Segmentation of point clouds using smoothness constraints," in *ISPRS Symposium: Image Engineering and Vision Metrology, International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences* (2006), Vol. **36**, pp. 248–253.
51. O. Chum and J. Matаš, "Randomized RANSAC with $T_{d,d}$ test," in *Proceedings British Machine Vision Conference* (2002), Vol. **2**, pp. 448–457.
52. L. Breiman, "Random forests," Mach. Learn. **45**, 5–32 (2001).
53. R. Blomley and M. Weinmann, "Using multi-scale features for the 3D semantic labeling of airborne laser scanning data," ISPRS Ann. Photogramm. Rem. Sens. Spat. Inf. Sci. **IV-2/W4**, 43–50 (2017).
54. M. Weinmann and M. Weinmann, "Geospatial computer vision based on multi-modal data—How valuable is shape information for the extraction of semantic information?" MDPI Rem. Sens. **10**, 2 (2018).
55. H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle, "Surface reconstruction from unorganized points," ACM SIGGRAPH Comput. Graph. **26**, 71–78 (1992).
56. F. Rottensteiner, G. Sohn, J. Jung, M. Gerke, C. Baillard, S. Benitez, and U. Breitkopf, "The ISPRS benchmark on urban object classification and 3D building reconstruction," ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci. **I-3**, 293–298 (2012).
57. Y. Li, B. Yong, H. Wu, R. An, and H. Xu, "An improved top-hat filter with sloped brim for extracting ground points from airborne lidar point clouds," MDPI Rem. Sens. **6**, 12885–12908 (2014).
58. P. Cheng, Z. Hui, Y. Xia, Y. Y. Ziggah, Y. Hu, and J. Wu, "An improved skewness balancing filtering algorithm based on thin plate spline interpolation," Appl. Sci. **9**, 203 (2019).
59. X. Meng, Y. Lin, L. Yan, X. Gao, Y. Yao, C. Wang, and S. Luo, "Airborne LiDAR point cloud filtering by a multilevel adaptive filter based on morphological reconstruction and thin plate spline interpolation," Electronics **8**, 1153 (2019).
60. G. Sithole, *Segmentation and Classification of Airborne Laser Scanner Data*, Vol. **59** of Publications on Geodesy (2005).

61. R. B. Rusu and S. Cousins, "3D is here: Point cloud library (PCL)," in *IEEE International Conference on Robotics and Automation* (2011), pp. 1–4.
62. R. Raguram, J.-M. Frahm, and M. Pollefeys, "A comparative analysis of RANSAC techniques leading to adaptive real-time random sample consensus," in *Proceedings of the European Conference on Computer Vision* (Springer, 2008), pp. 500–513.
63. R. Schnabel, R. Wahl, and R. Klein, "Efficient RANSAC for point-cloud shape detection," Comput. Graph. Forum **26**, 214–226 (2007).