Classification of LiDAR Point Clouds Using Supervoxel-Based Detrended Feature and Perception-Weighted Graphical Model

Yusheng Xu, *Member, IEEE*, Zhen Ye, Wei Yao, Rong Huang, *Student Member, IEEE*, Xiaohua Tong, *Senior Member, IEEE*, Ludwig Hoegner, and Uwe Stilla, *Senior Member, IEEE*

***Abstract*—Interpretation of 3-D scene through LiDAR point clouds has been a hot research topic for decades. To utilize measured points in the scene, assigning unique tags to the points of the scene with labels linking to individual objects plays a crucial role in the analysis process. In this article, we present a supervised classification approach for the semantic labeling of laser scanning points. A novel method for extracting geometric features is proposed, removing redundant and insignificant information in the local neighborhood of the supervoxels. The proposed feature extraction method uses the supervoxel-based local neighborhood instead of points as basic elements, encapsulating the geometric features of local points. Based on the initial classification results, the graph-based optimization is used to spatially smooth the labeling results, based on the graphical model using the perception weighted edges. Benefiting from the graph-based optimization process, our supervised classification method required only a few training datasets. Experiments were carried out by comparing the semantic labeling results with manually generated ground truth datasets. The performance of the proposed methods with different characteristics was analyzed. By using our testing datasets, we have achieved an overall accuracy of better than 0.8 for assigning the measured points to eight semantic classes.**

***Index Terms*—Classification, detrended geometric features, graphical model, LiDAR, optimization, supervoxel context.**

Manuscript received May 11, 2019; revised August 23, 2019 and October 4, 2019; accepted October 28, 2019. Date of publication November 17, 2019; date of current version February 12, 2020. This work was supported in part by the National Key Research and Development Project of China under Grants 2018YFB0505400 and 2017YFB0502700, and in part by the National Natural Science Foundation of China under Grant 41631178. The work of W. Yao was supported by Bavarian Excellence program based on the Bavarian Elite Aid Act. *(Corresponding author: Zhen Ye.)*

Y. Xu, R. Huang, L. Hoegner, and U. Stilla are with the Department of Photogrammetry and Remote Sensing, Technische Universität München, 80333 Munich, Germany (e-mail: yusheng.xu@tum.de; rong.huang@tum.de; ludwig.hoegner@tum.de; stilla@tum.de).

Z. Ye is with the Department of Photogrammetry and Remote Sensing, Technische Universität München, 80333 Munich, Germany, and also with the College of Surveying and Geo-Informatics, Tongji University, Shanghai 200092, China (e-mail: yezhen0402@126.com).

W. Yao is with the Department of Land Surveying and Geo-Informatics, The Hong Kong Polytechnic University, Hung Hom, Hong Kong (e-mail: wei.hn.yao@polyu.edu.hk).

X. Tong is with the College of Surveying and Geo-Informatics, Tongji University, Shanghai 200092, China (e-mail: xhtong@tongji.edu.cn).

Digital Object Identifier 10.1109/JSTARS.2019.2951293

I. Introduction

In recent decades, automated interpretation of 3-D scenes with LiDAR has been a popular research topic in fields of photogrammetry [1], remote sensing [2], computer vision [3], architecture [4], civil engineering [5], cadastral investigation [6], and robotics [7]. As a flexible and portable measurement technique, LiDAR can obtain point clouds via different acquisition platforms and systems, such as airborne laser scanning (ALS) using aerial platform, like airplanes or UAVs, terrestrial laser scanning (TLS) using fixed tripod platform, and mobile laser scanning (MLS) using moveable platforms, like vehicles or boats. ALS measures point clouds with a far observation distance and a relatively low density, which is usually used for mapping and monitoring a large area since the flying aerial platform can easily cover a large investigation area. While, for accurate analysis and interpretation of 3-D urban scenes, TLS and MLS, providing higher scanning density with close observation distance and more precise points with static carrier platform or stations, are more competent.

However, only having applicable platforms and systems is far from enough to accomplish the analysis of 3-D scenes. To fully understand and extract 3-D information from the scene, unique labels of given categories should be used to mark the acquired 3-D points in the scene, which reveals the semantic meanings of the objects represented by the points. Therefore, assigning semantic tags to points belonging to objects of the different categories is crucial in the overall 3-D scenario analysis workflow [8]. Supervised classification is one of the solutions to accomplish this labeling task. Although a lot of related research has been presented, this work is still a challenge. For the majority of supervised classification methods, a well-trained classifier is required, with a large amount of training dataset needed [9]. Nevertheless, the conventional way of generating training data of point clouds still depends on manual work, which is a more challenging endeavor than manually labeling the 2-D image due to the complex 3-D structures. Fortunately, 3-D point clouds encode more precise topological and geometric information of the real 3-D scene than those of 2-D images, if we can provide additional constraints or assumptions based on these topological relations between structures and geometric characters of points, we could achieve a supervised classification without using a large percentage of the training dataset. Especially, point clouds

of urban scenes encapsulate some form of regularity with specific structures, which can be exploited to improve the accuracy of semantic labels [10].

To this end, we present a supervised semantic labeling method designed for classifying 3-D points, with supervoxel-based detrended features calculated in the local neighborhood and graph-based optimization. The effort spent on the reduction of the required training dataset considers two aspects. On one hand, we conduct a preclustering of points with geometric homogeneity, which provides constraint at the local level. Points of the cluster are forced to share the same label. In our work, this preclustering is achieved by supervoxels with refined boundaries. For the estimated geometric features, we proposed a novel detrending algorithm removing redundant and insignificant information in the local environment of the points. On the other hand, a regularization of labeled points is also applied to improve the accuracy of initial labels estimated by the classifier. In our work, this is achieved by the global graph optimization of labels. Here, we have an assumption that the perception-weighted graphical model has a natural relation to the representation of the 3-D scene, which reflects both the geometric and topological information. Therefore, by constructing and optimizing the graphical model, wrong labels can be corrected.

The following are innovative contributions in our proposed approach. 1) A novel detrended geometric feature, removing the local tendency of the geometric characteristic of the local neighborhood, is proposed. Our novel feature extraction method is effective for delineating the local geometry in the 3-D scene. Different from our previous work [11], apart from the geometric and height features, in this work, we also took surface and contextual features into consideration, which are concatenated in various ways. Moreover, in this work, the intensity values of the points are not used. Instead, it is a purely geometric solution. 2) Without using points as fundamental elements, the supervoxel-based context is designed in the local vicinity to encode geometric attributes of points, achieving a preclustering of homogenous points. Conventional segment-based classification methods heavily rely on the quality of obtained segments, so that we compromise this issue by using the refined supervoxels. In this step, a boundary refined supervoxelization algorithm is developed, which is an improvement of the classic voxel cloud connectivity segmentation (VCCS) algorithm by refining the boundaries between supervoxels. 3) A perception-weighted graphical model is constructed and optimized to improve the results of the initial classification result, which can reassign those wrongly labeled fragments with correct semantic labels. The remainder of this article is organized as follows. A brief literature view of point clouds classification is given in Section II. The methodology of the proposed point cloud classification method is provided in Section III. Subsequently, the tested point cloud datasets, experimental results, and related discussions are given in Sections IV and V. Finally, Section VI concludes the article.

II. Related Work

Semantic labeling of 3-D point clouds, aiming at tagging a unique semantic label to an individual point, is a fundamental task for urban mapping and remote sensing. Currently, a well-designed point cloud classification method could involve three core steps: Extraction of features, classification using extracted features, and smoothing of labels. Based on the various derived features, initial labels can be assigned to points by applying classifiers. Then, benefiting from label-smoothing techniques, the initial label of every point could be further refined according to external information or constraints, with those wrong labels corrected.

A. Extraction of Features

The extraction of features is for abstracting local geometric information of the given point and encapsulating the information into feature vectors [12]. Generally, there are two key factors influencing the extraction of features: 1) selection of neighborhoods for the investigated element (e.g., point or segment), and 2) parameterization of geometrical characteristics with appropriate descriptors for generating discriminative feature vectors.

It is essential to select appropriate neighborhood describing details in the near of a given point [8]. For various purposes, it is necessary to rely on different objective details of all points within the selected neighborhood. Neighborhood definitions in common use are divided into different categories, namely, the single-scale neighborhood and the multiscale neighborhood. The formal one extracts features using a constant neighborhood, while the later one utilizes multiple neighborhoods with flexible sizes and forms. A common way to generate the local neighborhood, which is defined by a specific number of neighbors (e.g., k-nearest neighbors (KNN) or neighbors with a given distance) [13] or 2-D projective distance [14]. Besides, spherical [15] and cylindrical [16] neighborhoods are also representatives of the single-scale neighborhood. For multiscale neighborhoods, different features are separately extracted from various neighborhoods with different forms and sizes, and then combine them to encode the output feature vectors. In [8], a neighborhood selection approach based on multiple individually optimized neighborhoods is proposed. Similarly, in [17], multiscale neighborhoods for selecting features are introduced to enhance the performance of 3-D point clouds classification. Although different classifiers will definitely influence the performances of the entire classification workflow, the importance of features also matters a lot, especially when the classifier is identified and has insufficient training samples. In [18], for detecting vehicles from the scene, the authors implement a multilayer model for feature generation consisting of partitioned octree structures of several levels. Moreover, in [19], a Latent Dirichlet allocation is adopted to generate features, which are derived from point-based hierarchical clusters, in order to classify objects having varying sizes.

For parameterizing geometrical characteristics with appropriate descriptors, using 3-D shape descriptors to encode local or global geometric information around the point of interests is a commonly adopted solution. Based on the spatial distribution of 3-D points around the point of interest, both local and global features of this point can be quantized into a vector having a fixed number of bins by the use of the 3-D shape descriptor. Here, the 3-D shape descriptor is actually an algorithm transforming 3-D coordinates to features via mathematical formulations, which

can describe the geometry of the local area around the point of interests. In the recent decade, a number of representative feature descriptors have been developed [20], for instance, 3-D context shape [21], Fast Point Feature Histogram (FPFH) [7], and Signature of Histogram of Orientations (SHOT) descriptor as well as its variants [22]. However, since all these descriptors rely mainly on the detailed description of geometric properties, and they seldom focus on the generalized structural features. Thus, they cannot immune noise and lack a strong geometric and topological relationship. Therefore, for characterizing the general geometry of an object, novel methods are proposed. Instead of considering detailed geometric and texture distributions, they estimated eigenvalues calculated from 3-D coordinates of points, which can reflect and express essential geometric information and topological relationships between the point and its neighbors. In this respect, the geometry description based on eigenvalues [23], [24] is an example. The eigenvalues of the tensor of coordinates characterize the 3-D features of the shape. In addition to the original spatial coordinates, number of returns and intensity, it is also possible to enrich the point attributes used to describe the features. For instance, the RGB color [25] and the thermal information [26] are used for featuring the 3-D point cloud.

B. Classification Using Extracted Features

Point and segment based methods are typical strategies for classification using extracted features. Point-based classification will gain a label for each point when conducting classification [8], [27]. While the segment-based classification applies segmentation like preclustering to the point cloud for getting primitives having homogeneities [10], [28], [29] before the further classification. The segment-based strategy has the advantage that it can separate individual objects from the scene simultaneously. No matter which strategy is being used, classifiers also contribute a lot to the classification performance. Here, AdaBoost [30], support vector machines [31], [32], random forest (RF) [24], conditional random field [33], or its high-order variations [34], and deep neural networks [35], [36] are the representative ones.

The voxel-based data structure is a popular way of preparing point clouds. Unlike conventional point-based data structure, the voxel structure can easily cope with nonuniform point density and problems caused by varying observation distances of large-scale point clouds. Octree-based structure [37] is an example, simplifying the entire dataset and suppressing noise and outliers with rasterized 3-D grids. The octree can identify neighboring relations of created voxels and index the inside points simultaneously. The tree structure can facilitate the searching of neighbors as well. Voxels can also be aggregated into larger supervoxels by the use of methods, such as VCCS [38] algorithm, which can explore the potentialities of voxel structures. A supervoxel is generated by grouping neighboring voxels via a graphical model or k-means clustering. One of the major advantages of using supervoxel structures is that they can precisely discover boundaries between objects or different parts of an object. When supervoxels are utilized in classification task, the generation of supervoxels is actually a preclustering of voxels with common properties, like normal vectors, colors, or other geometric attributes [39]. After such a clustering, the boundary of voxels from different clusters likely corresponds to the edges of various objects. Since this clustering is an unsupervised process, the size and the shape of the cluster is only adjusted by the difference of the local geometry of those voxels belonging to different objects. Namely, in the same cluster, voxels have always similar local geometric characteristic, which can better reflect the geometry of the local area. If we use such clusters as the neighborhood for the extraction of features, it could be an optimized solution. Based on the over-segmentation idea, it is also possible for supervoxels to be further merged into larger local patches utilizing a predefined radius of vicinity [19], [29], to delineate geometric features in a local vicinity more completely.

C. Smoothing of Labels

The spatial regularity is an elementary prior knowledge about acquired measuring data of the real world [40], [41]. To derive a regularized labeling, smoothing of labels can be designed to refine the initial tags obtained by the classification results, with the assumption that adjacent pixels or points are likely to share the same object label (i.e., class). Generally, the smoothing approaches can be conducted with two different strategies, i.e., local label smoothing and global label smoothing [34].

Local smoothing of labels focuses on the weight assignment of adjacent entities, which is mainly implemented via local filtering, local graph optimization, or preclustering of points. In [11], the initial labels of classification results are corrected by a voting-based filter process via a local first-order graph. In [42], the segmentation of local graphical model is also adapted to optimize initial labels. Moreover, in [10], instead of directly labeling individual points, a nonparametric segmentation is first conducted to aggregate points into segments sharing common labels, which could be regarded as a geometric constraint between labels and points. However, the performance of local label smoothing largely depends on the quality of the initial classification and definition of the neighborhood [34]. To be specific, the major assumption of applying the local smoothing is that the wrong labels are mainly surrounded by the corrected ones [11], but such a constraint is difficult to set for irregular shaped point clouds because large incorrectly labeled regions remain after the smoothing. Besides, how to design appropriate neighborhoods is still a challenging task.

As an alternative to the local label smoothing strategy, global label smoothing methods are also explored by a wide variety of studies, which consider the labels of points in the entire scene, simultaneously. The global label smoothing is mainly achieved via the optimization or regularization under graphical models or Markov networks [41]. In [43], the initial classification is globally optimized by adopting a multiclass graph cut algorithm, followed by a refinement with a local optimization using an object-oriented decision tree. In [36], a regularization framework using a global graph structure is proposed and employed for smoothing labels of point cloud segments, with impressive results achieved. Similarly, in [34], the initial labels with relaxed probabilities are optimized via graph-structured

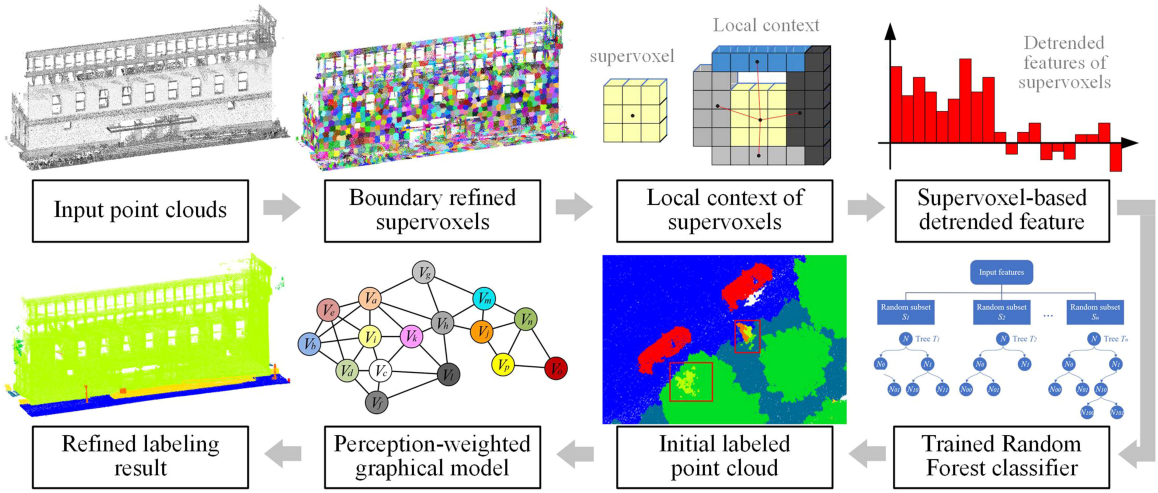


Fig. 1. Workflow of our point cloud classification.

regularization. The same as the local label smoothing strategy, the quality of initial labels matters, which usually serves as the prior knowledge for smoothing. Besides, the construction of graphical models or Markov networks also plays a vital role for the optimization or regularization, which should consider both the spatial relationship (e.g., topology) and defined weights (e.g., similarity or proximity) between points. The graph can be built not only based on a KNN structure [36], but also the one considering manifold structure, like Riemannian graph [44]. The optimization or regularization is achieved by solving the cost function formulated from the graphical models or Markov networks. However, the selection of appropriate solver of the cost function is still needed to be addressed when dealing with complex urban areas [41].

III. Methodology

In Fig. 1, a brief overview of the workflow is displayed, with key steps and representative results illustrated. The entire workflow includes four essential procedures: Supervoxelization and local neighborhood selection, extraction of detrended geometric features, supervised classification, and graph-based optimization. At first, an oversegmentation is implemented by the use of the VCCS method [38], but with boundaries refined. A local neighborhood is defined for every supervoxel, considering neighbors directly connected. Second, geometric features of every supervoxel and connected neighbors in the local neighborhood are calculated. Afterward, a local tendency of geometry is estimated for each supervoxel in the feature space. The estimated local tendency will detrend the geometric features of the center supervoxel. In the supervised classification step, RF classifier is used to distinguish points of objects utilizing the detrended features, with initial labels obtained. Finally, graph-based optimization is conducted to refine the initial labels with a perception-weighted graphical model.

A. Boundary Refined Supervoxelization

To generate the voxel structure for point clouds, the entire space is partitioned into small cubic grids employing octree, splitting each parent node into eight equal child nodes. Here, the octree structure is conducted through the approximate nearest neighbor [45] searching algorithm. The supervoxelization is conducted via VCCS algorithm. However, the supervoxels of VCCS always suffer from the "zig-zag" effect, namely, the edges are not smooth ones. Instead, they are twisted conforming the squared edges of basic voxels, because the fundamental element of VCCS is the cubic shaped voxel [11]. To be specific, the "zig-zag" effect that we want to remove is the one between/across two adjacent objects. For supervoxels generated within one single object (i.e., wall or ground), there would not have any effect for feature extraction. However, the edge between two objects at the meanwhile should also be the edge between two supervoxels, which means that such effect is originally caused by the "zig-zag" edges of each supervoxel generated by VCCS. Since the edges of a supervoxel are formed by cuboid-shaped voxels, the edges between/across two adjacent objects cannot reach a point-level accuracy. In such situations, supervoxel located at the edge of one object may be contaminated by points of other adjacent object, which may affect the feature extraction. To overcome this problem, we proposed a boundary refined supervoxel clustering algorithm for creating supervoxels with a point-level accuracy of their boundaries.

Our proposed boundary refined supervoxel is based on the original VCCS supervoxel, consisting of two major steps, namely, the detection of boundary points, and the refinement of boundary points. In the first step, all the points of one supervoxel will be measured by the distance from the point to the center of the supervoxel considering the local curvature [44] exploring the spatial proximity of adjacent supervoxels in geodetic space.

In the supervoxel V, from the point P_i to neighboring point P_j, the distance D_{proj} is calculated by its projected point $P_j{}'$ on the tangent plane of P_i defined by the normal vector N_i. If D_{proj} is larger than a given threshold of θ, the point is regarded as a boundary point. Empirically, the θ is set to $r_{\text{seed}}/2$, where r_{seed} is the seed resolution of supervoxels. The radius size of spherical neighborhoods for estimating the normal vector is equal to the size of the voxel. Then, in the second step, a local k-mean clustering is conducted between the boundary point and

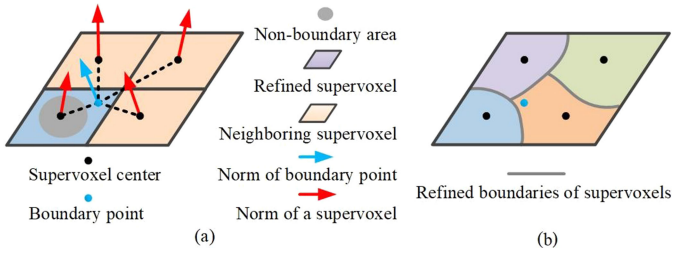


Fig. 2. Boundary refined VCCS supervoxelization. (a) Local k-means clustering of boundary points in neighboring supervoxels. (b) Refined boundaries between supervoxels.

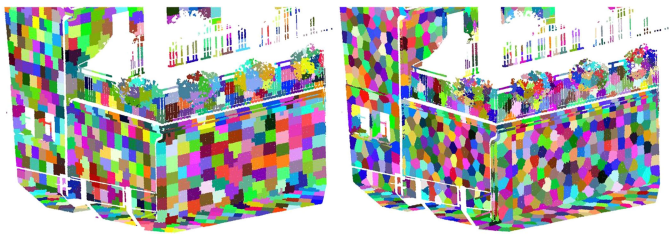


Fig. 3. Illustration of boundary refined VCCS supervoxelization.

the centers of neighboring supervoxels [see Fig. 2(a)]. Here, the clustering is governed by a distance measure calculated in a feature space, considering the normal vectors and spatial distance

$$D_{\text{proj}} = \sqrt{w_n \cdot ||N_i - N_b||_2 + w_d \cdot ||X_i - X_b||_2} \quad (1)$$

where N_i and N_b are the normal vectors of the center of one neighboring supervoxel and the boundary point, while X_i and X_b are their positions, respectively. w_n and w_d denote the weight factors for the angle between normal vectors and the distance between centroids, respectively. In the experiments, w_n and w_d are empirically set to one and the reciprocal value of the size of voxels, respectively. In our approach, merely spatial distance and normal vectors are used during supervoxelization (also for the VCCS step in our work), which shows better performance when finding real boundaries of objects than the implementation at the level of using merely voxels. In Fig. 3, we show the difference between supervoxels generated from the original VCCS supervoxelization and our boundary refined VCCS supervoxelization. As seen from the figure, the boundary between supervoxels from the original VCCS method suffer a significant "zig-zag" effects. By contrast, the boundary between supervoxels from our boundary refined VCCS method is greatly smoothed. For example, the boundary between the door of the garage and ground surface should be a straight line, but in the results of the original VCCS method, we can clearly find the twisted edges. On the contrary, the results from our method can fully recover this boundary with a straight line.

B. Detrended Feature Extraction

Considering a large number of 3-D points containing only spatial coordinates, we need to extract geometric features from the 3-D coordinates to describe the geometry of the object. Since the supervoxel and its local neighborhood are already known, it is necessary to properly use them to represent the local geometry. Therefore, geometric features based on eigenvalues [8], [24], as well as additional structural features, are introduced to delineate both the geometric and structural characteristics in the local area of the point of interests. The eigenvalue-based geometric features are used to represent the local geometry of the object (e.g., the size and shape of the local area), while additional structural features are added to describe the basic structure of the local neighborhood (e.g., height and topological information).

Eigenvalue-based features, including linearity L_λ, planarity P_λ, scattering S_λ, omnivariance O_λ, anisotropy A_λ, eigenentropy E_λ, local curvature C_λ as well as the sum of eigenvalues $\sum_\lambda$, can be derived following the work presented in [8]. The L_λ, P_λ, and S_λ describe the dimensionality of the points, while O_λ, A_λ, E_λ, C_λ, and $\sum_\lambda$ encode statistical features for the shape description. In addition to eigenvalue-based features derived from the 3-D structure tensor, height features, orientation features (i.e., normal vectors and verticality), and surface features (i.e., local point density D) are also introduced as additional information for the geometry description. Furthermore, considering the interaction between the supervoxel itself and the local context, we also utilize relative position R_p, relative direction R_d, and spatial distribution pattern R_s advocated in [46]. The relative position denotes the averaged distance d_{oi} between the center supervoxel V_o and its first-order neighbor V_i in the local context. For the relative direction, it relates to the averaged angle between the normal vector of the center supervoxel and those of its first-order neighbors in the local context. The spatial distribution pattern stands for the averaged angle of the orientation angle a_{oi} formed by the center supervoxel V_o and the first-order neighboring supervoxel V_i. The angle mentioned here is formed by the connection lines between the centers of the centering supervoxel and those of its neighbors. In detail, in Table I, we show the definition of each vector in the output feature vectors.

1) Local Neighborhood of the Supervoxel: Although supervoxel structures have preclustered voxels at lower levels, supervoxels prefer to partition objects into small pieces, which leads to dissimilarities among different patches belonging to the same surface. Therefore, the decision tree cannot be trained perfectly. To address this issue, we use the idea given in [19], utilizing information from all supervoxels in the first-order graph of a given supervoxel. To be specific, for each supervoxel, we will define a local neighborhood to gain contextual information. In Fig. 4, we show the illustration of the defined local neighborhood for a given supervoxel.

2) Detrending Local Tendency of Supervoxel-Based Context: Regarding the interpretation of complex 3-D scenes, there are typically various objects, and it is necessary to identify the exact boundaries between the objects. Furthermore, according to the analysis performed in [10], the contribution of each vector from the local descriptor in the generated vector of features

TABLE I
LIST OF FEATURES USED IN THE FEATURE VECTORS

Features	Definition	Category
Linearity	$L_\lambda = \frac{e_1 - e_2}{e_1}$	Dimensionality features [8]
Planarity	$P_\lambda = \frac{e_2 - e_3}{e_1}$	
Scattering	$S_\lambda = \frac{e_3}{e_1}$	
Omnivariance	$O_\lambda = \sqrt[3]{e_1 \cdot e_2 \cdot e_3}$	
Anisotropy	$A_\lambda = (e_1 - e_3)/e_1$	
Eigenentropy	$E_\lambda = -\sum_{i=1}^{3} e_i \ln(e_i)$	Statistical features [24]
Local curvature	$C_\lambda = \frac{e_3}{e_1 + e_2 + e_3}$	
Sum of eigenvalues	$\sum_\lambda = e_1 + e_2 + e_3$	
Height mean	$\frac{1}{n}\sum_{i=1}^{n} Z_i$	Height features [47]
Height difference	$Z_{max} - Z_{min}$	
Normal vectors	N_x N_y N_z	Orientation features [48]
Verticality	$1 - N_z$	
Local density	$D = \frac{3n}{4\pi r_{seed}{}^3}$	Surface features
Relative position	$R_p = \frac{1}{n}\sum_{i=1,\ldots,n} d_{oi}$	
Relative direction	$R_d = \frac{1}{n}\sum_{i=1,\ldots,n} n_{oi}$	Contextual features
Distribution pattern	$R_p = \frac{1}{n}\sum_{i=1,\ldots,n} a_{oi}$	[46]

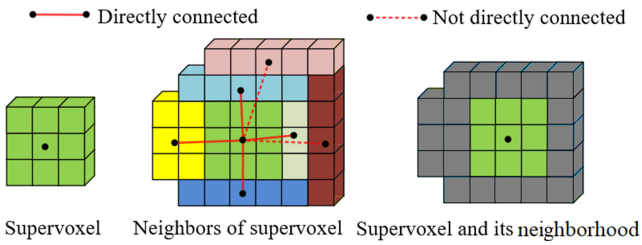


Fig. 4. Local neighborhood of the supervoxel.

(i.e., feature histogram) varies even for objects of the same type. This will result in the ambiguity when generating features for different types of objects. For instance, the geometric features of natural ground and artificial ground could be quite similar if we consider linearity, flatness, and orientation of normal vectors. In such a case, we need to use additional features, like the surface smoothness and roughness, to distinguish them. For the implemented vector of features, we carry out the process of enhancing the useful and salient feature vectors and weakening trivial feature vectors.

Enlightened by the edge detection operator (i.e., difference of Gaussian) of the domain of image processing, we used the idea given in [11] to estimate the local trend of the 3-D geometry of each supervoxel in the local environment, and then removed this local tendency to obtain salient information about objects that represent unique structures and detail elements. Additionally, the local trend of the supervoxel background also acts as a vital role when finding supervoxels near the real boundary of objects having different semantic labels. In Fig. 5, we show the 1-D outline, which illustrates the estimation of the local trend of the geometric surface of the object. It is obvious that after removing local trends, two curves having originally similar layouts will be distinguishable.

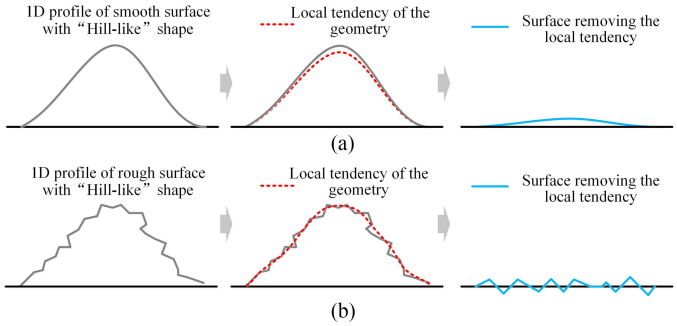


Fig. 5. Illustration of local tendency of geometric shapes. (a) For objects with smooth surface. (b) For objects with rough surface.

The geometric features based on eigenvalues basically reflect the general geometry relating to "low frequency" components of the geometry. In contrast, the detrended feature is also a kind of "high-pass" filter, removing background geometry information from nearby parts and leaving "high-frequency" components. Then, if we can describe the complete geometry of the object by integrating these two components, better uniqueness can be reached.

3) Detrended Geometric Features in the Local Neighborhood: For creating geometric features with local tendency detrended, we calculate dimensionality features, statistical features, height features, orientation features, and surface features from points of supervoxels in the local neighborhood. The dimensionality, statistical, and surface features mainly reflect the 3-D shape of the object, namely, those relatively detailed components. In contrast, the structural, height, and orientation features can provide contextual information of the object relating to fundamental components. In the meantime, contextual features encapsulate the interaction in the context. Thus, if we can combine these components together, better distinctiveness can be achieved for representing the geometry of objects.

The local tendency is expressed in the feature space and removed for each supervoxel. The vector of features of a given supervoxel V is denoted by H_v in (2). The feature vector of the local neighborhood of the supervoxel standing for the local tendency is noted by H_l in (2). The vector of contextual features is represented by H_r in (2). H_v is calculated according to the dimensionality, statistical, height, orientation, and surface features listed in Table I using points within the supervoxel itself. While H_l is also calculated according to the dimensionality, statistical, height, orientation, and surface features listed in Table I, but here, the points we used include points within all supervoxels in the local neighborhood. H_r is calculated according to the contextual feature listed in Table I. Then, the detrended geometric feature vector H_d is obtained by the following operation:

$$H_d = H_v - H_l. \tag{2}$$

The output vector of features H is achieved by a weighted concatenation of H_v, H_d, and H_r

$$H = [H_v^T, k \cdot {H_d}^T, {H_r}^T] \tag{3}$$

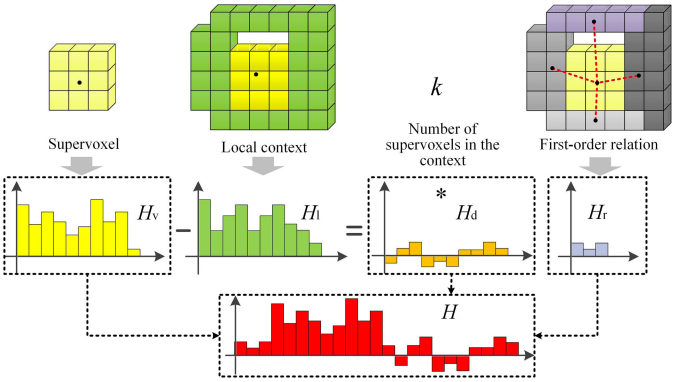


Fig. 6. Generation of the vector of features.

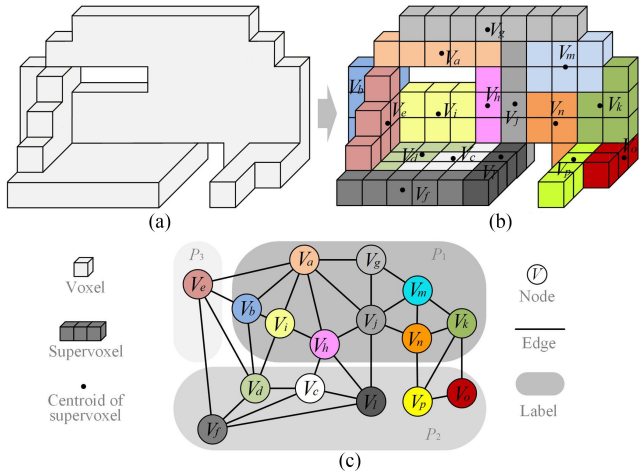


Fig. 7. Global graph structure. (a) 3-D scene. (b) Supervoxelized 3-D space. (c) Generated global graph of labeled supervoxels.

where k is the factor weighting the strength of the local tendency, estimated by the number of used supervoxels. Finally, a vector of features with 33 bins for supervised classification is achieved. We illustrate the concatenation of different geometric features in Fig. 6.

It is noteworthy that compared with the method given in [11], in our approach, we do not use radiometric features (e.g., RGB color or intensity), and only 3-D coordinates are utilized. To some extent, our proposed detrended geometric feature uses a similar strategy like the difference of normal feature presented in [49], which generate the difference of angles between normal vectors estimated from various sizes of neighborhoods. The difference is that what we used is more than normal vectors, instead, also get the difference of local geometries, height values, verticalities, and densities. Besides, we also consider the contextual features representing the interaction between the supervoxel and its context.

C. Initial Labeling With RF

The supervised classification using the classical RF algorithm [50] is implemented to distinguish supervoxels with different predefined labels. An RF classifier combines a number of decision trees that are created by randomized vectors that are sampled independently of the input vector of features. Each decision tree votes to elect the most probable label of the sample of the input vector [50]. Besides, the RF classifier will grow a tree at each node, which enforce it to be insensitive to overfitting following the strong law of large numbers. During training, the bagging method is applied to each combination of features, generating a training dataset by drawing N examples with random replacement. Here, N is the size of the original training set. After the supervised classification, each supervoxels V_i as well as all the points within it will be given a soft label P_i, where $P \in S$ and $S = \{p \in [0,1]^{\mathcal{K}} | \sum_{k \in \mathcal{K}} p_k = 1\}$. The probability of a supervoxel V_i having the label $k \in \mathcal{K}$ is calculated by

$$P_{i,k} = \frac{N_k}{N_t} \tag{4}$$

where N_k is the amount of decision trees voting for class k, while N_t is the number of all the trees.

D. Graph Structured Global Smoothing of Labels

To refine the results of RF, we adopt a global optimization for spatial smoothing based on the perception-weighted graphical model, as advocated in [51]. This optimization aims to find an improved labeling result $\hat{P}$, and the solution should provide the labeling of supervoxels with enhanced spatial smoothness and remain as close as possible to the input labeling P [41].

1) Construction of Perception-Weighted Graphical Model: The idea of using perception-weighted graphical model is that we assume that the graphical model could naturally represent the spatial space of 3-D scenes. Here, perceptual grouping, referring to the determination of regions and parts from visualization and perception that should belong to the same piece of higher-level perceptual elements [52], is adopted. This is because, for all the points of the same object, they are likely to form a smooth, continuous, and convex surface. Thus, to encode the spatial constraint from the aspect of perceptual grouping laws, we consider cues of proximity, continuation, and similarity, measuring the spatial distance, the difference angles of normal vectors, and the difference between feature vectors.

More specifically, to structure the objective functional of this optimization, we first construct a weighted graphical model $G = (V, E, W)$, in which nodes denote supervoxels, and the edges are assigned with weights W. Regarding each supervoxel $V_i \in V$ as a node, all supervoxels of its KNN in Euclidean space will be connected. Then, a global graph is composed of all the connected nodes. An illustration of the generated global graph can be found in Fig. 7. Here, $\{P_1, P_2, \ldots, P_n\}$ is given initial labels of n supervoxels, and the weight $W(i,j)$ of edge $e(i,j)$ between V_i and V_j is assessed by the proximity relating to the spatial distance ΔX_{ij}, the continuation relating to the difference of normal vector angles ΔA_{ij}, and the similarity relating to the difference of feature vectors ΔH_{ij}. The weight value $W(i,j)$ ranges from 0 to 1. The spatial distance represents the Euclidean distance between V_i and V_j

$$\Delta X_{ij} = ||\vec{X}_i - \vec{X}_j||_2 \tag{5}$$

while the difference angles of normal vectors ΔA_{ij} between planes formed by points within supervoxels V_i and V_j

$$\Delta A_{ij} = \angle(\vec{N}_i, \vec{N}_j). \tag{6}$$

For measuring the similarity C^{s}_{ij} between V_i and V_j, the difference ΔH_{ij} between H_i and H_j is used

$$\Delta H_{ij} = \sum_{k=1}^{8} \left(\frac{h_i(k) - h_j(k)}{h_i(k) + h_j(k)} \right)^2. \tag{7}$$

Here, the smaller the value ΔH_{ij}, the more similar the 3-D shapes between two objects. Finally, the weight of edges is computed with the following:

$$W(i,j) = \exp\left(-\frac{\delta_x \Delta X_{ij} + \delta_a \Delta A_{ij} + \delta_h \Delta H_{ij}}{2\theta^2} \right) \tag{8}$$

where δ_x, δ_a, and δ_h are weight factors, and θ is the bandwidth of the Gaussian kernel.

2) Optimization of Graphical Model: The above-mentioned graphical model can be formulated to an optimization problem. P^* is the solution of the structured optimization problem

$$P^* \in \arg\min_{Q \in \Omega} \sum_{i \in V} \phi(P_i, Q_i) + \sum_{(i,j) \in E} \lambda \cdot \psi(Q_i - Q_j) \tag{9}$$

where ϕ and ψ stand for the fidelity term and the regularizer, respectively. The strength of regularization λ is a positive value denoting the strength for regularization and Ω represents the search space. The fidelity term $\phi(P, Q)$ influencing the initial labeling P will decrease if Q is closer to P. By contrast, the regularizer $\psi(Q_i, Q_j)$ guarantees that optimized labels of V_i and V_j are spatially smooth. λ balances the influence of the regularization regarding the fidelity term [41]. Here, the penalizer $\psi(a, b)$ influences the relation between adjacent nodes V_a and V_b, and is thus determined by Potts model [53]

$$\psi(a,b) = \begin{cases} 0 & \text{if } P_a = P_b \\ 1 & \text{if } P_a \neq P_b \end{cases} \tag{10}$$

where l_a and l_b are the labels of V_a and V_b. The regularization strength λ is estimated as follows:

$$\lambda = \exp\left(-\frac{(d_{ij})^2}{\delta^2} \right) \tag{11}$$

where d_{ij} is the weight (i.e., $W(i,i)$) between two supervoxels, and δ is the expectation of all neighboring weights. While the fidelity term $\phi(p, q)$ is a smooth and convex function, which is calculated following a linear-logarithmic function of the observed probability, which tends to induce discrete hard labels [41]:

$$\phi(p,q) = -\sum_{k \in \mathcal{K}} q_k \log\left(\frac{\alpha}{k} + \alpha p_k \right) \tag{12}$$

where $\alpha \in [0, 1]$, and the entrywise logarithm can make the observed probability to be smoothed to prevent numerical issues [41].

The minimization problem is solved by a graph-cut strategy using the alpha expansion, which can quickly find an approximate solution with a few graph-cut iterations. The implementation of the alpha expansion is achieved by the use of GCO-V3.0 library [54]–[56]. Here, the labeling cost is not considered since we assume that labels of all the objects are independent so that all elements in the labeling cost matrix are set to one, except the diagonal ones setting to zero. The optimization results automatically adapt to the underlying scene without the need for predefined features of certain potential objects.

IV. Experiments

A. Test Datasets

Experiments are conducted on two different datasets in urban scenes, including TUM city campus dataset [57] and Semantic3D dataset [58]. For the TUM MLS dataset, the testing site is in the area of the city campus of the Technical University of Munich, covering around 80 000 m^2. Fraunhofer Institute of Optronics, System Technologies and Image Exploitation (IOSB) [57] originally acquires this dataset. Two Velodyne HDL-64E, used for acquiring point clouds, are mounted on the top of the vehicle [57].

For the evaluation process, the ground truth is generated by manually labeling of point clouds. As a consequence, a highly precise reference of the entire city campus is made. In Fig. 8, the labeled scene with eight semantic classes is rendered by eight different colors, including building, high vegetation, low vegetation, vehicles, human-made terrain, natural terrain, hardscape, and scanning artefacts [27].

While the TLS dataset is used to further test the versatility of our classification method on point clouds with varying point density. Here, we used the popular Semantic3D dataset published by ETH Zürich [27]. This dataset is manually labeled into eight different classes, namely building, hight vegetation, low vegetation, vehicles, human-made terrain, natural terrain, hardscape, and scanning artefacts. In this experiment, the scans we used are Bildstein and Untermaederbrunnen. Each scene has three scans, and we use two of them as training dataset and the rest one as test dataset. In Fig. 8, we show the manually labeled reference of these two scenes, with labels provided by www.semantic3d.net.

B. Evaluation Metric

For the evaluation of the classification, we follow the Pascal VOC challenges [59] and use Intersection over Union (IoU) averaged over all classes. The evaluation measure for class i is defined as

$$\mathrm{IoU}_i = \frac{\mathrm{TP}_i}{\mathrm{TP}_i + \mathrm{FP}_i + \mathrm{FN}_i}. \tag{13}$$

The main evaluation measure is the $\overline{\mathrm{IoU}}$, which is the average summation of IoU_i for each class i. Moreover, the overall accuracy (OA) are also calculated

$$\overline{\mathrm{IoU}} = \frac{1}{N} \sum_{i=1}^{N} \mathrm{IoU}_i \tag{14}$$

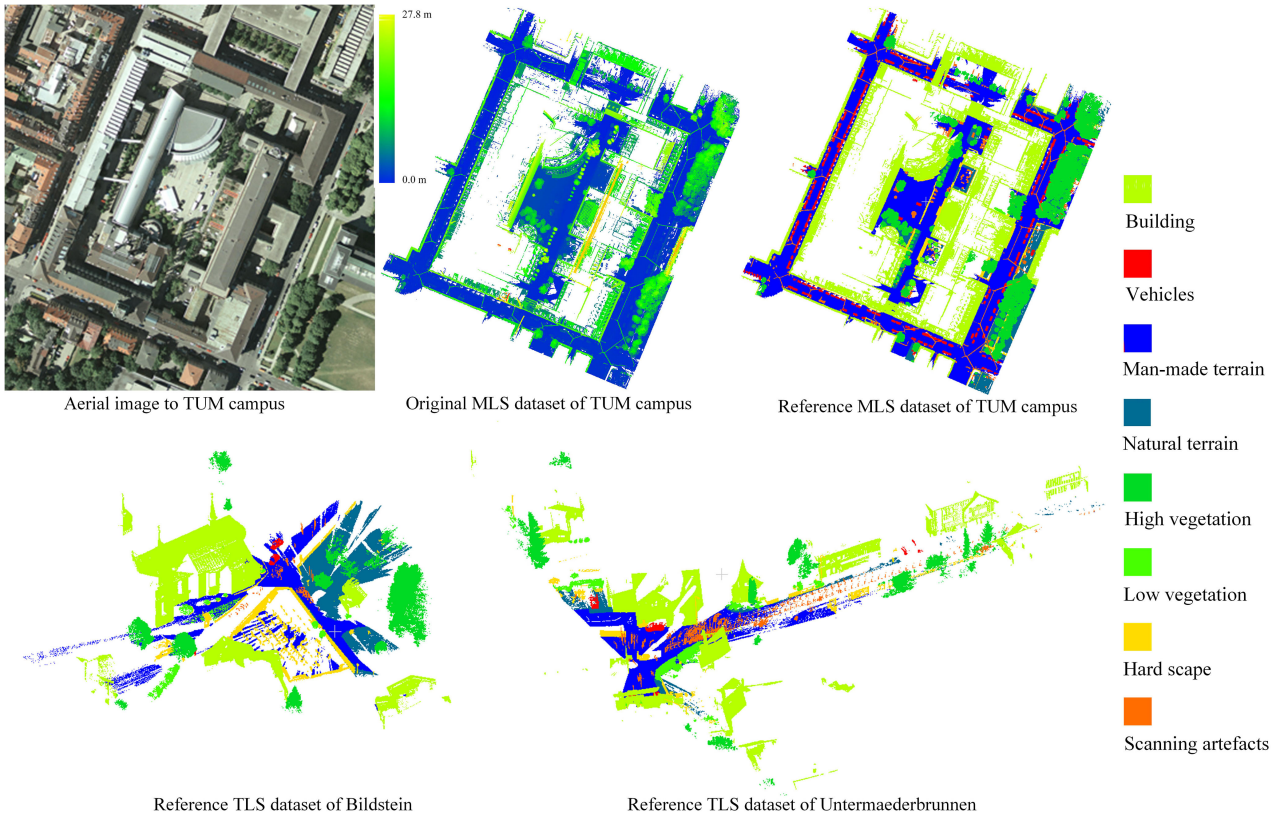


Fig. 8. Testing dataset with manually labeled reference.

Here, for the labeled result of each class, TP denotes the true positive, which is the number of points correctly labeled as this class, namely the points with correct labels. FP stands for the false positive, which means the number of points with incorrect labels. FN is the false negative, which is the number of points which should be labeled as other classes, but incorrectly labeled as this class. Besides, the precision (Pre.) and recall (Rec.) values are also given for assessing the performance, and finally, the OA will be calculated as well.

$$\mathrm{OA} = \sum_{i=1}^{N} \left(\frac{\mathrm{TP}_i}{\mathrm{TP}_i + \mathrm{TN}_i + \mathrm{FP}_i + \mathrm{FN}_i} \right). \tag{15}$$

V. Results and Discussion

A. Preprocessing

Although the dataset suits well for the work, it has some drawbacks. For instance, the original dataset is exceptionally dense, including more than one billion points only for the Arcisstrasse, which could not be handled efficiently. Besides, the alignment between scans is not so accurate, that the objects are usually with variable thickness. Therefore, appropriate processing for later work is necessary. The original raw point cloud is preprocessed first by the statistical outlier removal filter, and then down sampled. Preprocessed points have been reduced to around 50 million points, namely about only 5% of the original point cloud. In Fig. 9, a comparison between a subscene of the raw and preprocessed point clouds is illustrated. It is shown that apparent noise and outliers are removed, while the structures of objects are well preserved.

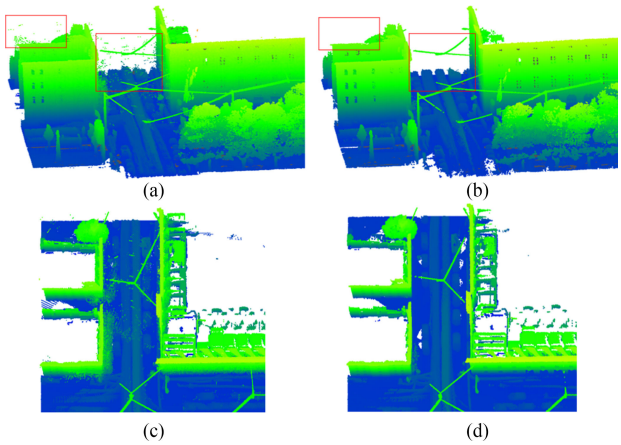


Fig. 9. Preprocessing of point clouds. (a) and (d) Original point clouds. (b) and (c) Preprocessed point clouds.

B. Experimental Results

In semantic labeling experiments, our algorithms used for feature extraction and initial classification are implemented via C++ which is run on an Intel i7-6700 CPU @ 3.4 GHz and with 32.0 GB RAM.

TABLE II
COMPARISON FOR CLASSIFICATION RESULTS USING TUM DATASET WITH DIFFERENT FEATURES

Method	SV			CX			RL			DE		
Class	*Pre.*	*Rec.*	*IoU*	*Pre.*	*Rec.*	*IoU*	*Pre.*	*Rec.*	*IoU*	*Pre.*	*Rec.*	*IoU*
Buildings	0.897	0.742	0.684	0.822	0.754	0.648	0.767	0.604	0.510	0.912	0.826	0.765
Vehicles	0.272	0.364	0.184	0.273	0.202	0.131	0.218	0.301	0.145	0.571	0.595	0.411
Man-made terrain	0.895	0.676	0.626	0.841	0.664	0.590	0.552	0.884	0.515	0.920	0.691	0.652
Natural terrain	0.117	0.375	0.098	0.129	0.330	0.102	0.091	0.002	0.002	0.159	0.515	0.138
High vegetation	0.343	0.797	0.316	0.305	0.451	0.222	0.863	0.757	0.676	0.415	0.794	0.375
Low vegetation	0.150	0.052	0.040	0.051	0.005	0.004	0.402	0.076	0.068	0.250	0.071	0.059
Hard scape	0.150	0.047	0.037	0.100	0.016	0.014	0.730	0.190	0.177	0.373	0.045	0.042
Scanning artefacts	0.462	0.079	0.072	0.038	0.113	0.029	0.469	0.092	0.083	0.682	0.133	0.125
MEAN	0.411	0.391	0.257	0.320	0.317	0.218	0.512	0.363	0.272	0.535	0.459	0.321
OA		0.695			0.66			0.706			0.760	

1) Results of Using TUM Datasets: When using the TUM datasets for conducting a supervised classification, we use only the area along the Acissstrasse as the training set, nearly 30% of the entire dataset, while the rest of the dataset (around 70%) is used as the testing set. For assessing the effectiveness of our proposed detrended features and the graph-structured optimization, we compared the method (termed as SV, i.e., using H_v) using merely features from points of the supervoxel without the local neighborhood information, the method (termed as CX, i.e., using H_l) using features from the local neighborhood information, the method (termed as RL, i.e., using H_d and H_r) using features removing the local tendency and the local contextual features, the method (termed as DE, i.e., using H) using our proposed detrended features without graph-structured optimization, and our proposed method (termed as DEGO, i.e., using H and graph optimization) with using our proposed detrended features and our graph-structured optimization. For the setting of key parameters, the size of the voxel is 0.3 m, while the seed resolution of supervoxels is 1.0 m. For the weight factors in the boundary refined process, w_n is set to one, while w_d is set to the reciprocal value of the size of voxels. The number of trees used in our RF classifier is 200. The default threshold for the graph cut is set to 0.5. Finally, we reach an OA of better than 0.86, for labeling eight semantic classes (see the legend in Fig. 8). In Table II, the comparison of classification results of using different features is shown.

With respect to the OA, methods using our detrended geometric features can overweight other methods, having an improvement of around 1% and 7%, respectively. It means that using the features removing the local tendency can get equivalent or even better performance as the one of using features from single supervoxel, with merely 1% improvement. However, when combining these two vectors of features, the accuracy can be drastically improved with 7%. This is because that in the combined situation, the original geometric features are kept, and at the meanwhile the details are enhanced, which facilitates the feature expression. For the IoU measures, our approach outperforms others as well. In particular, for scanning artefacts and vehicles, which are likely to be labeled as the building facades and low vegetation, our detrended feature can gain better IoU measures. The effectiveness of our detrended features can be backed by the analysis of features as well. In Fig. 10, we also provide a summary of feature importance of different vectors

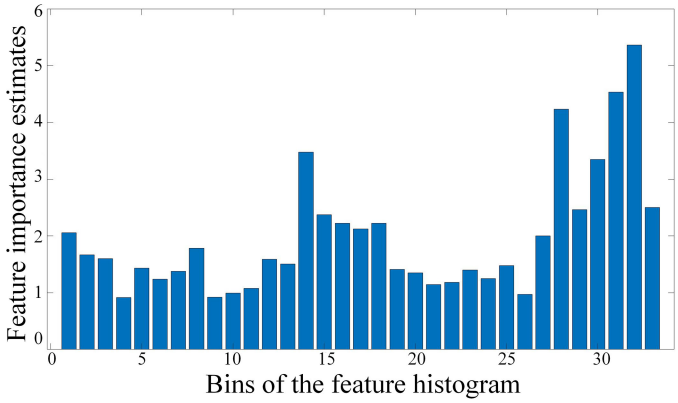


Fig. 10. Importance of feature vectors used in RF classifier.

in the RF process. As shown in the figure, we can observe that the last three vectors representing the contextual features play an important role in decision trees. Besides, the vectors of detrended features from the bins 15 to 30 in the vector of features are also essential to the creation of decision trees, with a higher averaged value of importance compared with those of the features from the supervoxel itself.

While the comparisons of classification results of using different graph cut thresholds τ are given in Table III, and it seems that with an increasing threshold of τ, we can achieve a better OA. However, what stands out is the low vegetation, which has been totally categorized into wrong classes after the optimization process when using a high threshold for graph optimization. This is because a high threshold of τ will result in large cliques in the graphical model after the optimization and vice versa. The generation of such large cliques will force small cliques or nearby nodes to merge into a large clique so that for small objects with different initial labels, they will be wrongly optimized.

Benefiting from the preclustering, one advantage of segmentation-based classification methods over classical point-based classification methods is the in-sensitiveness to outliers and noise. Meanwhile, the supervoxel structure also has some disadvantages. For instance, choosing the appropriate voxel size is a compromise between noise suppression and detail preservation and unify uneven density. The larger the voxel, the smoother the details. Obviously, for traditional boundaries, such as the right-angled corners of the corners formed by smooth walls,

TABLE III
COMPARISON FOR CLASSIFICATION RESULTS USING TUM DATASET WITH DIFFERENT GRAPH CUT THRESHOLDS

Method	DEGO, τ=0.2			DEGO, τ=0.5			DEGO, τ=0.8		
Class	*Pre.*	*Rec.*	*IoU*	*Pre.*	*Rec.*	*IoU*	*Pre.*	*Rec.*	*IoU*
Buildings	0.928	0.924	0.862	0.937	0.970	0.911	0.943	0.980	0.926
Vehicles	0.703	0.667	0.520	0.697	0.544	0.440	0.697	0.264	0.237
Man-made terrain	0.916	0.712	0.668	0.896	0.709	0.655	0.841	0.709	0.625
Natural terrain	0.170	0.522	0.147	0.169	0.514	0.146	0.161	0.493	0.138
High vegetation	0.610	0.859	0.555	0.768	0.908	0.713	0.816	0.931	0.769
Low vegetation	0.417	0.043	0.041	0.000	0.000	0.000	0.000	0.000	0.000
Hard scape	0.714	0.011	0.011	0.833	0.011	0.011	0.833	0.011	0.011
Scanning artefacts	0.919	0.106	0.105	0.945	0.081	0.081	0.951	0.071	0.071
MEAN	0.672	0.480	0.364	0.656	0.467	0.370	0.655	0.433	0.347
OA		0.835			0.866			0.870	

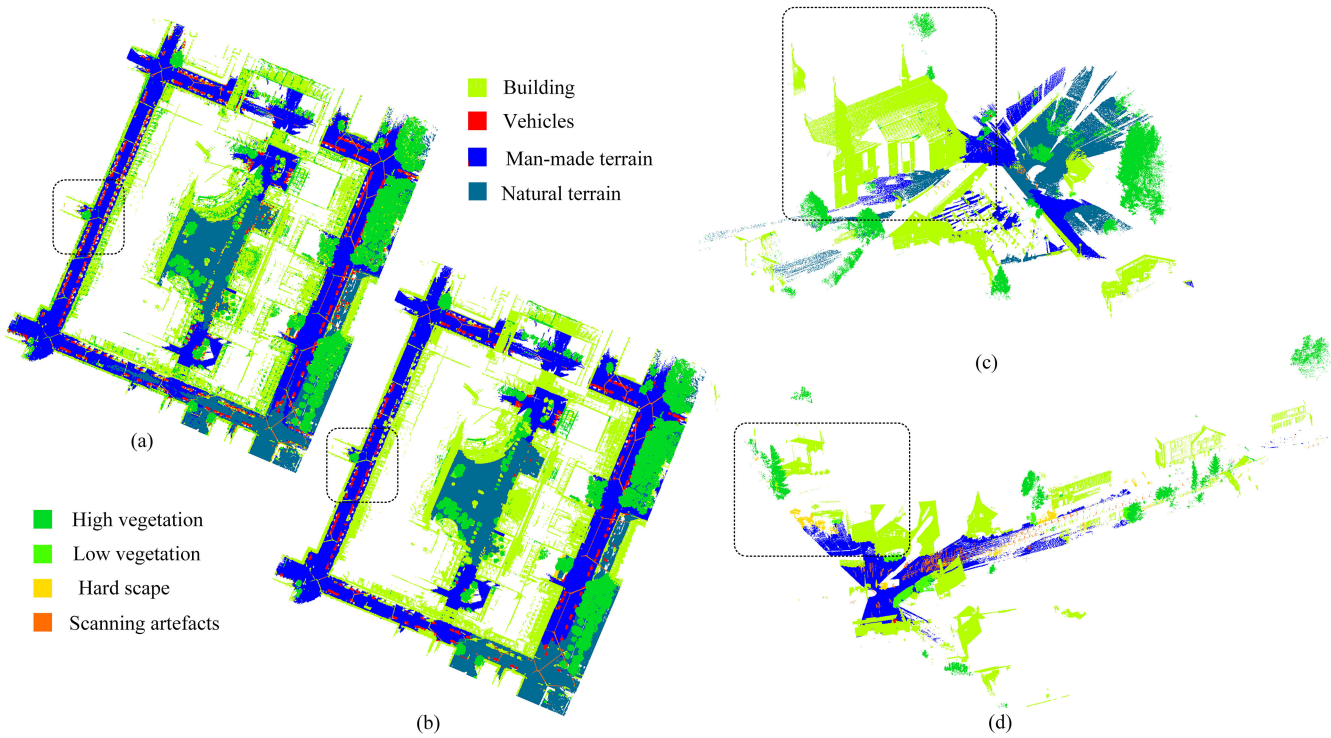


Fig. 11. Classification results. (a) Initial classification result of TUM campus. (b) Optimized classification result of TUM campus. (c) Optimized classification result of Bildstein. (d) Optimized classification result of Untermaederbrunnen.

our fine-grained boundary supervoxels can accurately find out these boundaries. However, when irregular edges are involved, for example, due to the presence of French windows, the edges between the walls and the ground, the boundaries found by supervoxels are biased. Besides, for small objects like minor scanning artefacts, if the size of the supervoxel is too large, they are easily blurred by their background and cannot be described correctly. This can be seen from the initial result that large objects like buildings and high vegetation can always achieve a good IoU value.

To have a complete view of the classification performance, we apply the trained classifier mentioned above to the entire TUM dataset, which is nearly four times larger than the training data of the Arcisstrasse. The classification result of our method is given in Fig. 11. As seen from the figure, corresponding to the performance shown in Table III, the semantic labeling result of the entire experimental area is given. In the result, we can find that majority of buildings, man-made terrain, and high vegetation are labeled correctly. Besides, we can find that our purposed method reveals excellent potential in positioning stationary vehicles, although this can only be achieved after the optimization. In Fig. 12, we give a detailed view of the optimization of vehicles and buildings. The initial classification result of vehicles is questionable because in fact, parts of points of vehicles are occluded due to the view direction of observation. These occluded observations of vehicles make the supervoxel of cars and buses look like part of hardscapes. After the optimization, those wrongly labeled supervoxels can be corrected. However, if the initial label of the local area is biased, as the lower-left corner of the campus scene in Fig. 12,

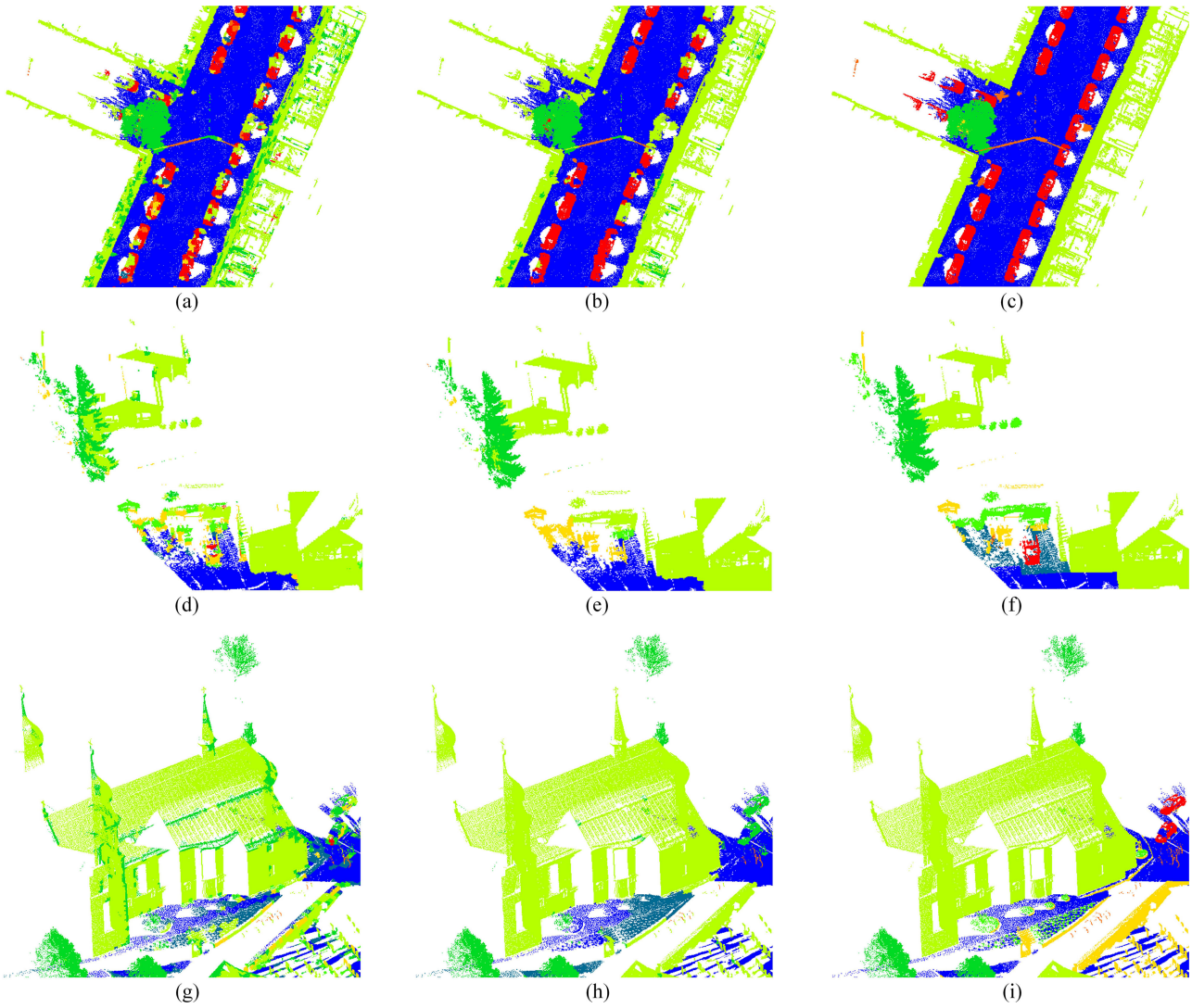


Fig. 12. Comparison of results before and after the optimization (the area in the boxes of Fig. 12). (a) Initial classification result. (b) Optimized classification result. (c) Ground truth of TUM campus. (d) Initial classification result. (e) Optimized classification result. (f) Ground truth of Untermaederbrunnen. (g) Initial classification result. (h) Optimized classification result. (i) Ground truth of Bildstein.

the optimization may make the situation worse. In this area, large parts of the road surface are initially labeled as natural terrain, so that the optimization is failed because for nodes in the graphical model, all their neighbors have wrong labels. Besides, due to the complexity of the large testing area, there are misclassifications, such as "sparkling effects" [11], where the area within the facade of the building is incompletely measured and too sophisticated for classification. Thus, certain parts of the inner building structure are considered to be disturbing objects. Although part of the fragmentation error label is normalized, there are still a large number of areas with incorrect labels.

For a further evaluation of the performance of our proposed method, we also compared our proposed method with other two baseline methods PointNet [60] and PointNet++ [61], which are both renown deep learning based classification methods for 3-D points. In Table IV, we illustrate the comparison of the classification results. Here, the training and testing datasets we used for PointNet and PointNet++ are the same as those used in our method. Here, to fulfill the requirement for input in PointNet and PointNet++, the entire point cloud is subdivided into thousands of subpoint chips, in which 10 000 points are contained. These chips are downsampled to 8192 points, which represent the main structure of each chip, and the downsampled chips serve as the input for PointNet and PointNet++. Each point in the chip is represented by a 3-D vector, containing the coordinates (x, y, z). For the training process, each training batch contained in a total of 16 chips. The stochastic gradient descent algorithm with a learning rate $\eta = 0.001$ and a momentum value of $p = 0.9$ was utilized. For adjusting the learning rate, we decayed its value by the factor of 0.7 in every 40 training chips. The training process lasts for a total of 500 epochs. We monitor the progress of the validation loss and save the weights if the loss improves. Both of these methods were implemented via Tensorflow and carried out by a NVIDIA TITAN X (Pascal) 12 GB GPU. As seen from the results, our proposed method has significant advantages

TABLE IV
COMPARISION OF CLASSIFICATION RESULTS USING TUM DATASET WITH DIFFERENT METHODS

Method	PointNet [60]			PointNet++ [61]			Ours, DEGO, τ = 0.5		
Class	*Pre.*	*Rec.*	*IoU*	*Pre.*	*Rec.*	*IoU*	*Pre.*	*Rec.*	*IoU*
Buildings	0.8707	0.915	0.7642	0.9223	0.9098	0.7924	0.937	0.97	0.911
Vehicles	0.4637	0.5692	0.7606	0.7079	0.9255	0.7881	0.697	0.544	0.44
Man-made terrain	0.8782	0.5907	0.5459	0.9228	0.4587	0.4418	0.896	0.709	0.655
Natural terrain	0.1456	0.4056	0.4594	0.1494	0.6512	0.3358	0.169	0.514	0.146
High vegetation	0.6428	0.5703	0.4459	0.7248	0.6954	0.4351	0.768	0.908	0.713
Low vegetation	0.1549	0.0147	0.4426	0.458	0.1509	0.4324	0.000	0.000	0.000
Hardscape	0.0353	0.0199	0.7635	0.3643	0.1633	0.7919	0.833	0.011	0.011
Scanning artefacts	0.1444	0.2192	0.7614	0.1685	0.384	0.7882	0.945	0.081	0.081
MEAN	0.4169	0.4131	0.618	0.5523	0.5424	0.6007	0.656	0.467	0.370
OA		0.758			0.773			0.866	

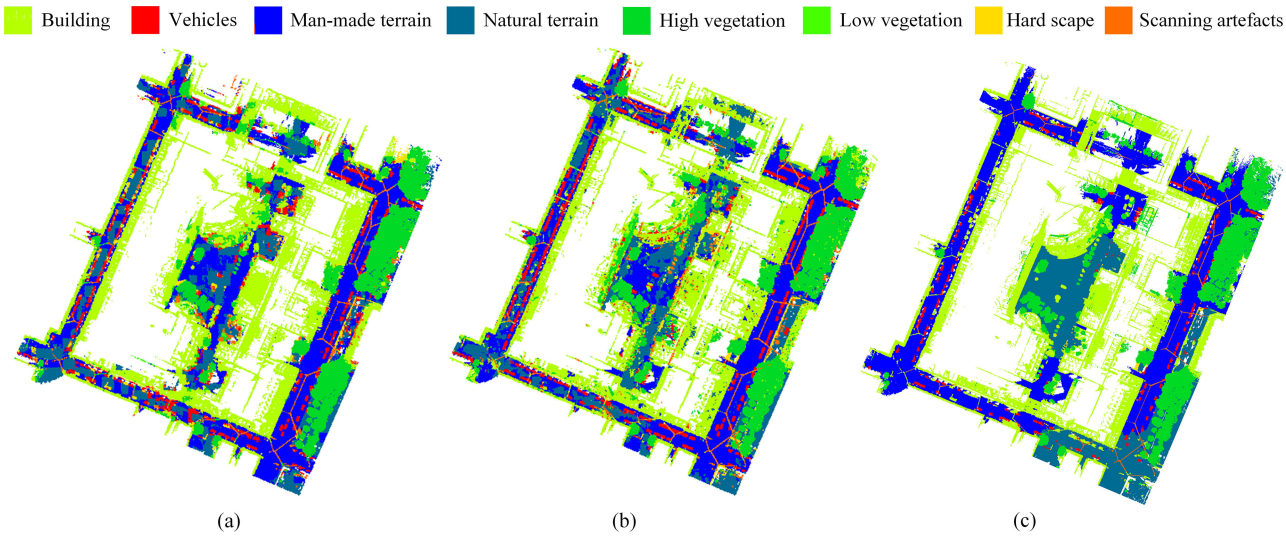


Fig. 13. Comparison of classification results using different methods. (a) Classification result using PointNet. (b) Classification result using PointNet++. (c) Classification result using our proposed method.

over these two baseline methods, when checking the overall accuracies. Especially, our proposed method outperforms the others when labeling buildings, man-made terrain, and high vegetation. The possible reason could be that the these three kinds of objects have generally isotropic geometric characteristics, which can facilitate the graph-based optimization process considering the local contextual information. However, for the vehicles and low vegetation, the deep learning based methods show better results. One of the explanations could be that the features used in the deep learning based methods are not generated from feature engineering. Instead, they are generated by supervised learning, so, theoretically, they should perform better when dealing with irregular shaped objects. However, due to the lack of training samples, they cannot fully perform the advantages of neural networks. On the other hand, this can also be regarded as the merits of our proposed method. Similarly, we apply the trained models of PointNet and PointNet++ to the entire TUM dataset. The results of different methods are shown in Fig. 13. As seen from the figure, the visualization results can also support the quantitative evaluation results that our proposed method performs better when discriminating buildings and high vegetation, but is weak at classifying vehicles and low vegetation.

2) Results of Using ETH Datasets: We tested our method using the semantic3D dataset provided by ETH Zürich [27], in order to explore the full potentiality of our feature extraction method. It is worth noting that the semantics 3-D dataset is measured by TLS that causes the density of the points to vary with the distance between the station and the object. Here, the voxel size is 0.3 m and the supervoxel seed resolution is 1.0 m. For the weight factors in the boundary refined process, w_n and w_d are set to one and the reciprocal value of the size of voxels, respectively. The number of trees used in our RF classifier is 200. The default threshold for the graph cut is set to 0.75. In Fig. 11, we provide results for the experimental scenario. As illustrated in the results, for our main concerns like buildings, roads, and tall trees, the results of our method reveal the enormous potential when using our detrended features. Our optimization is suitable to the points of the building and the high vegetation, but for areas with significant changes in density, the artificial terrain is easily considered to be a natural terrain.

The evaluation of using two datasets is given in Tables V and VI. In the experiments of Bildstein scene, we used the scans of Bildstein 3,5 for training. Then, the point cloud of Bildstein 1 is used for validation, classifying points into eight

TABLE V
EVALUATION FOR CLASSIFICATION RESULTS USING BILDSTEIN DATASET

Method		DE			DEGO, τ=0.5	
Class	$Pre.$	$Rec.$	IoU	$Pre.$	$Rec.$	IoU
Buildings	0.804	0.770	0.648	0.803	0.949	0.770
Vehicles	0.083	0.016	0.014	0.000	0.000	0.000
Man-made terrain	0.861	0.655	0.593	0.850	0.644	0.578
Natural terrain	0.627	0.875	0.576	0.632	0.943	0.609
High vegetation	0.759	0.983	0.749	0.883	0.999	0.882
Low vegetation	0.050	0.030	0.019	1.000	0.015	0.015
Hard scape	0.838	0.067	0.066	1.000	0.001	0.001
Scanning artefacts	0.941	0.314	0.308	0.962	0.245	0.243
MEAN	0.620	0.464	0.371	0.766	0.474	0.387
OA		0.754			0.811	

TABLE VI
EVALUATION FOR CLASSIFICATION RESULTS USING UNTERMAEDERBRUNNEN DATASET

Method		DE			DEGO, τ=0.5	
Class	$Pre.$	$Rec.$	IoU	$Pre.$	$Rec.$	IoU
Buildings	0.799	0.891	0.728	0.830	0.979	0.815
Vehicles	0.412	0.253	0.186	0.889	0.096	0.095
Man-made terrain	0.805	0.966	0.783	0.750	0.979	0.738
Natural terrain	0.667	0.045	0.044	0.500	0.003	0.003
High vegetation	0.743	0.702	0.565	0.909	0.854	0.787
Low vegetation	0.374	0.397	0.239	0.532	0.361	0.274
Hard scape	0.400	0.389	0.246	0.452	0.266	0.201
Scanning artefacts	0.886	0.607	0.563	0.942	0.722	0.692
MEAN	0.636	0.531	0.419	0.725	0.533	0.451
OA		0.747			0.804	

classes. As given in the table, we can observe that our method can still get an outcome having an OA of 0.811. Interestingly, the classification of vehicles, low vegetation, and hardscapes are almost failed in the test. The primary reason is due to the insufficient number of training samples, which is a frequent problem for segment-based point cloud classification, namely, for objects without enough training samples, like hardscapes and vehicles, the labeling is always failed. In contrast, for those objects having enough training samples, for instance, buildings and artificial terrain [see Fig. 12(e)], satisfying results can be achieved.

In the experiments of the Untermaederbrunnen scene, the scan we used for training is Untermaederbrunnen 1. For evaluation, the scan of Untermaederbrunnen 3 is used, involving all eight classes. As shown in the table, our proposed method can achieve an OA of 0.804. Similar to the result of the previous scene, buildings, man-made terrain, high vegetation (see Fig. 12) can obtain good OA, but for the natural terrain and vehicles both the initial labeling and optimization failed due to the insufficient training samples. It is noted that for vehicles, TLS laser scanning may cause more severe occlusions so that the scanning points of vehicles have confusing geometry like that of hardscapes.

Indeed, the result of using the ETH dataset cannot compare with the methods, which have higher OA reaching 0.9, but considering that our proposed method is supervised and requires much less training dataset, the results are satisfying. Besides, for the object of our primary concern (i.e., buildings), both our proposed detrended features and graph-structured optimization perform well, and the voxel-based data structure significantly accelerates the processing speed. For practical applications, LiDAR points are always textured with reliable intensity or RGB color, the performance of our method can be further improved by this radiometric information like in [11]. Moreover, in our proposed method, we only use the RF classifier for obtaining the initial labels, but in fact, any classifier providing soft labels can be used in our methods. An excellent initial labeling result can significantly improve the effectiveness of graph-structured optimization. In future work, a better initial labeling method could be utilized.

VI. CONCLUSION

In this work, we proposed a supervised point cloud classification method using a novel detrended geometric features, removing redundant and insignificant information in the local neighborhood of the supervoxels. The proposed feature extraction method uses the supervoxel-based local neighborhood instead of points as basic elements, encapsulating the geometric features of local points. Based on the initial classification results, the graph-based optimization is used to spatially smooth the labeling results, based on the graphical model using the perception weighted edges. The optimization of labeling can be naturally represented in terms of energy minimization [62]. We minimize the energy function constructed with a data term and piecewise smoothness term via a graph cut algorithm, i.e., alpha-expansion proposed in [54], which finds a good approximate solution by iteratively running min-cut/max-flow algorithms on an appropriate graph. This move making algorithm iteratively selects a label and considers moves increasing the clique of pixels that are given this label if the movement has lower energy. Besides, the constructed energy function can be justified in the context of maximum *a posteriori* estimation of Markov Random Fields. The discrete multilabel MRF are solved by applying min-cut/max-flow algorithms iteratively to binary-labeled piecewise MRF [63]. At the moment, our result of using the Semantic3D dataset cannot be compared with the state-of-the-art methods with OA reaching 0.9 yet, but for the object of our primary concern (i.e., buildings), both our proposed detrended features and graph-structured optimization perform well, and the voxel-based data structure significantly reduce the number of elements needed to be processed, which is similar to a downsampling procedure. However, we should also admit that the handcrafted feature has naturally drawbacks compared with those generated from supervised learning, because it is always difficult to manually design the feature expression algorithm and tune the parameters, although they may perform in some specific situations. In the future, features generated by the learning based methods should be more promising and of wide use.

ACKNOWLEDGMENT

This work was carried out within the frame of Leonhard Obermeyer Center (LOC) at Technische Universität München (TUM).

REFERENCES

[1] G. Vosselman and H.-G. Maas, *Airborne and Terrestrial Laser Scanning*. Boca Raton, FL, USA: CRC Press, 2010.

[2] M. A. Lefsky, W. Cohen, S. Acker, G. G. Parker, T. Spies, and D. Harding, "Lidar remote sensing of the canopy structure and biophysical properties of douglas-fir western hemlock forests," *Remote Sens. Environ.*, vol. 70, no. 3, pp. 339–361, 1999.

[3] N. Buch, S. A. Velastin, and J. Orwell, "A review of computer vision techniques for the analysis of urban traffic," *IEEE Trans. Intell. Transp. Syst.*, vol. 12, no. 3, pp. 920–939, Sep. 2011.

[4] F. Remondino and S. Campana, *3D Recording and Modelling in Archaeology and Cultural Heritage: Theory and Best Practices*. Oxford, U.K.: Archaeopress, 2014.

[5] F. Bosché, M. Ahmed, Y. Turkan, C. T. Haas, and R. Haas, "The value of integrating scan-to-BIM and scan-vs-BIM techniques for construction monitoring using laser scanning and BIM: The case of cylindrical MEP components," *Autom. Construction*, vol. 49, pp. 201–213, Jan. 2015.

[6] J. Du *et al.*, "A novel framework for 2.5-D building contouring from large-scale residential scenes," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 6, pp. 4121–4145, Jun. 2019.

[7] R. B. Rusu, N. Blodow, and M. Beetz, "Fast point feature histograms (FPFH) for 3D registration," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2009, pp. 3212–3217.

[8] M. Weinmann, B. Jutzi, S. Hinz, and C. Mallet, "Semantic point cloud interpretation based on optimal neighborhoods, relevant features and efficient classifiers," *ISPRS J. Photogrammetry Remote Sens.*, vol. 105, pp. 286–304, 2015.

[9] Z. Li, L. Zhang, R. Zhong, T. Fang, L. Zhang, and Z. Zhang, "Classification of urban point clouds: A robust supervised approach with automatically generating training data," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 10, no. 3, pp. 1207–1220, Mar. 2017.

[10] S. Guinard and L. Landrieu, "Weakly supervised segmentation-aided classification of urban scenes from 3d lidar point clouds." *Int. Arch. Photogrammetry, Remote Sens. Spatial Inf. Sci.*, vol. 42, 2017.

[11] Z. Sun, Y. Xu, L. Hoegner, and U. Stilla, "Classification of mls point clouds in urban scenes using detrended geometric features from supervoxel-based local context," *ISPRS Ann. Photogrammetry, Remote Sens. Spatial Inf. Sci.*, vol. IV-2, pp. 271–278, May 2018.

[12] Y. Guo, M. Bennamoun, F. Sohel, M. Lu, and J. Wan, "3D object recognition in cluttered scenes with local surface features: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 11, pp. 2270–2287, Nov. 2014.

[13] L. Linsen and H. Prautzsch, "Local versus global triangulations," in *Eurographics—Short Presentations*. Geneva, Switzerland: Eurographics Association, 2001.

[14] J. Niemeyer, F. Rottensteiner, and U. Soergel, "Contextual classification of lidar data and building object detection in urban areas," *ISPRS J. Photogrammetry Remote Sens.*, vol. 87, pp. 152–165, 2014.

[15] I. Lee and T. Schenk, "Perceptual organization of 3d surface points," *Int. Arch. Photogrammetry, Remote Sens. Spatial Inf. Sci.*, vol. 34, no. 3/A, pp. 193–198, 2002.

[16] S. Filin and N. Pfeifer, "Segmentation of airborne laser scanning data using a slope adaptive neighborhood," *ISPRS J. Photogrammetry Remote Sens.*, vol. 60, no. 2, pp. 71–80, 2006.

[17] W. Dong, J. Lan, S. Liang, W. Yao, and Z. Zhan, "Selection of lidar geometric features with adaptive neighborhood size for urban land cover classification," *Int. J. Appl. Earth Observ. Geoinformation*, vol. 60, pp. 99–110, 2017.

[18] Y. Yu, J. Li, C. Wen, H. Guan, H. Luo, and C. Wang, "Bag-of-visual-phrases and hierarchical deep models for traffic sign detection and recognition in mobile laser scanning data," *ISPRS J. Photogrammetry Remote Sens.*, vol. 113, pp. 106–123, 2016.

[19] Z. Wang *et al.*, "A multiscale and hierarchical feature extraction method for terrestrial laser scanning point cloud classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 5, pp. 2409–2425, May 2015.

[20] X. Yu, X. Liang, J. Hyyppä, V. Kankare, M. Vastaranta, and M. Holopainen, "Stem biomass estimation based on stem reconstruction from terrestrial laser scanning point clouds," *Remote Sens. Lett.*, vol. 4, no. 4, pp. 344–353, 2013.

[21] A. Frome, D. Huber, R. Kolluri, T. Bülow, and J. Malik, "Recognizing objects in range data using regional point descriptors," in *Proc. Comput. vision-Eur. Conf. Comput. Vision*, pp. 224–237, 2004.

[22] S. Salti, F. Tombari, and L. Di Stefano, "Shot: Unique signatures of histograms for surface and texture description," *Comput. Vision Image Understanding*, vol. 125, pp. 251–264, 2014.

[23] B. Jutzi and H. Gross, "Nearest neighbour classification on laser point clouds to gain object structures from buildings," *Int. Archives Photogrammetry, Remote Sens. Spatial Inf. Sci.*, vol. 38, pp. 4–7, 2009.

[24] N. Chehata, L. Guo, and C. Mallet, "Airborne lidar feature selection for urban classification using random forests," *Int. Archives Photogrammetry, Remote Sens. Spatial Inf. Sci.*, vol. 38, p. W8, 2009.

[25] K. Al-Manasir and C. S. Fraser, "Registration of terrestrial laser scanner data using imagery," *Photogrammetric Rec.*, vol. 21, no. 115, pp. 255–268, 2006.

[26] M. Weinmann, B. Jutzi, and C. Mallet, "Feature relevance assessment for the semantic interpretation of 3D point cloud data," *ISPRS Anna. Photogrammetry, Remote Sens. Spatial Inf. Sci.*, vol. 5, p. W2, 2013.

[27] T. Hackel, J. D. Wegner, and K. Schindler, "Fast semantic segmentation of 3d point clouds with strongly varying density," *ISPRS Ann. Photogrammetry, Remote Sens. Spatial Inf. Sci.*, vol. 3, no. 3, pp. 177–184, 2016.

[28] 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. Photogrammetry Remote Sens.*, vol. 66, no. 3, pp. 260–271, 2011.

[29] Y. Yu, J. Li, H. Guan, F. Jia, and C. Wang, "Learning hierarchical features for automated extraction of road markings from 3-D mobile lidar point clouds," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 8, no. 2, pp. 709–726, Feb. 2015.

[30] S. K. Lodha, D. M. Fitzpatrick, and D. P. Helmbold, "Aerial lidar data classification using adaboost," in *Proc. 6th Int. Conf. 3-D Digit. Imag. Model.*, 2007, pp. 435–442.

[31] S. K. Lodha, E. J. Kreps, D. P. Helmbold, and D. N. Fitzpatrick, "Aerial lidar data classification using support vector machines (SVM)." in *Proc. 3rd Int. Symp. 3D Data Process., Visualization, Transmiss.*, 2006, pp. 567–574.

[32] J. Secord and A. Zakhor, "Tree detection in aerial lidar and image data," in *Proc. IEEE Int. Conf. Image Process.*, 2006, pp. 2317–2320.

[33] E. H. Lim and D. Suter, "3D terrestrial lidar classifications with super-voxels and multi-scale conditional random fields," *Comput.-Aided Des.*, vol. 41, no. 10, pp. 701–710, 2009.

[34] N. Li, C. Liu, and N. Pfeifer, "Improving lidar classification accuracy by contextual label smoothing in post-processing," *ISPRS J. Photogrammetry Remote Sens.*, vol. 148, pp. 13–31, 2019.

[35] A. Vetrivel, M. Gerke, N. Kerle, F. Nex, and G. Vosselman, "Disaster damage detection through synergistic use of deep learning and 3d point cloud features derived from very high resolution oblique aerial images, and multiple-kernel-learning," *ISPRS J. Photogrammetry Remote Sens.*, vol. 140, pp. 45–59, 2017.

[36] L. Landrieu and M. Simonovsky, "Large-scale point cloud semantic segmentation with superpoint graphs," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2018, pp. 4558–4567.

[37] A.-V. Vo, L. Truong-Hong, D. F. Laefer, and M. Bertolotto, "Octree-based region growing for point cloud segmentation," *ISPRS J. Photogrammetry Remote Sens.*, vol. 104, pp. 88–100, 2015.

[38] J. Papon, A. Abramov, M. Schoeler, and F. Worgotter, "Voxel cloud connectivity segmentation-supervoxels for point clouds," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2013, pp. 2027–2034.

[39] D. Chen, J. Peethambaran, and Z. Zhang, "A supervoxel-based vegetation classification via decomposition and modelling of full-waveform airborne laser scanning data," *Int. J. Remote Sens.*, vol. 39, no. 9, pp. 2937–2968, 2018.

[40] K. Schindler, "An overview and comparison of smooth labeling methods for land-cover classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 50, no. 11, pp. 4534–4545, Nov. 2012.

[41] L. Landrieu, H. Raguet, B. Vallet, C. Mallet, and M. Weinmann, "A structured regularization framework for spatially smoothing semantic labelings of 3d point clouds," *ISPRS J. Photogrammetry Remote Sens.*, vol. 132, pp. 102–118, 2017.

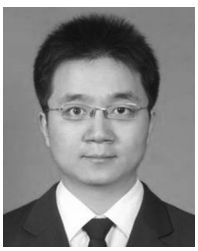
[42] B. Guo, X. Huang, F. Zhang, and G. Sohn, "Classification of airborne laser scanning data using jointboost," *ISPRS J. Photogrammetry Remote Sens.*, vol. 100, pp. 71–83, 2015.

[43] Z. Li *et al.*, "A three-step approach for TLS point cloud classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 9, pp. 5412–5424, Sep. 2016.

[44] M. Li, "A super voxel-based riemannian graph for multi scale segmentation of lidar point clouds," *ISPRS Ann. Photogrammetry, Remote Sens. Spatial Inf. Sci.*, vol. 4, no. 3, pp. 135–141, 2018.

[45] M. Muja and D. G. Lowe, "Fast approximate nearest neighbors with automatic algorithm configuration," in *Proc. 4th Int. Conf. Comput. Vision Theory Appl.*, vol. 2, pp. 331–340, 2009.

[46] B. Yang, Z. Dong, Y. Liu, F. Liang, and Y. Wang, "Computing multiple aggregation levels and contextual features for road facilities recognition using mobile laser scanning data," *ISPRS J. Photogrammetry Remote Sens.*, vol. 126, pp. 180–194, 2017.
[47] H.-G. Maas, "The potential of height texture measures for the segmentation of airborne laserscanner data," in *Proc. 4th Int. Airborne Remote Sens. Conf. Exhib./21st Can. Symp. Remote Sens.*, vol. 1, 1999, pp. 154–161.
[48] T. Rabbani, F. Van Den Heuvel, and G. Vosselmann, "Segmentation of point clouds using smoothness constraint," *Int. Arch. Photogrammetry, Remote Sens. Spatial Inf. Sci.*, vol. 36, no. 5, pp. 248–253, 2006.
[49] Y. Ioannou, B. Taati, R. Harrap, and M. Greenspan, "Difference of normals as a multi-scale operator in unorganized point clouds," in *Proc. Int. Conf. 3D Imag., Model., Process., Visualization Transmiss.*, Oct. 2012, pp. 501–508.
[50] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.
[51] Y. Xu, L. Hoegner, S. Tuttas, and U. Stilla, "Voxel-and graph-based point cloud segmentation of 3d scenes using perceptual grouping laws," *ISPRS Ann. Photogrammetry, Remote Sens. Spatial Inf. Sci.*, vol. 4, pp. 43–50, 2017.
[52] Y. Xu, W. Yao, S. Tuttas, L. Hoegner, and U. Stilla, "Unsupervised segmentation of point clouds from buildings using hierarchical clustering based on gestalt principles," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 11, no. 11, pp. 4270–4286, Nov. 2018.
[53] R. B. Potts, "Some generalized order-disorder transformations," in *Mathematical Proceedings of the Cambridge Philosophical Society*, vol. 48, no. 1. Cambridge, U.K.: Cambridge Univ. Press, 1952, pp. 106–109.
[54] Y. Boykov, O. Veksler, and R. Zabih, "Fast approximate energy minimization via graph cuts," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 11, pp. 1222–1239, Nov. 2001.
[55] V. Kolmogorov and R. Zabih, "What energy functions can be minimized via graph cuts?" *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 2, pp. 147–159, Feb. 2004.
[56] Y. Boykov and V. Kolmogorov, "An experimental comparison of min-cut/max- flow algorithms for energy minimization in vision," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 9, pp. 1124–1137, Sep. 2004.
[57] J. Gehrung, M. Hebel, M. Arens, and U. Stilla, "An approach to extract moving objects from MLS data using a volumetric background representation," *ISPRS Ann. Photogrammetry, Remote Sens. Spatial Inf. Sci.*, vol. 4, pp. 107–114, 2017.
[58] T. Hackel, N. Savinov, L. Ladicky, J. D. Wegner, K. Schindler, and M. Pollefeys, "SEMANTIC3D.NET: A new large-scale point cloud classification benchmark," *ISPRS Ann. Photogrammetry, Remote Sens. Spatial Inf. Sci.*, vol. IV-1-W1, 2017, pp. 91–98.
[59] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *Int. J. Comput. Vision*, vol. 88, no. 2, pp. 303–338, 2010.
[60] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2017, pp. 652–660.
[61] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," in *Proc. Advances Neural Inf. Process. Syst.*, 2017, pp. 5099–5108.
[62] R. Szeliski *et al.*, "A comparative study of energy minimization methods for Markov random fields with smoothness-based priors," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 6, pp. 1068–1080, Jun. 2008.
[63] V. Lempitsky and A. Zisserman, "Learning to count objects in images," in *Proc. Advances Neural Inf. Process. Syst.*, 2010, pp. 1324–1332.



Yusheng Xu (Member, IEEE) was born in 1989. He received the B.S and M.E. degrees from Tongji University, Shanghai, China, in 2011 and 2014, respectively, and the Ph.D. (Dr.-Ing.) degree from the Chair of Photogrammetry and Remote Sensing, Technical University of Munich (TUM), Munich, Germany, in 2019.

He is currently a Scientific Collaborator and Lecturer within the Chair of Photogrammetry and Remote Sensing of TUM. His research interests include point clouds processing, 3-D reconstruction, image processing, and photogrammetry.

Zhen Ye received the B.E. and the Ph.D. degrees from Tongji University, Shanghai, China, in 2011 and 2018, respectively.

He is currently a Postdoctoral Researcher with the Chair of Photogrammetry and Remote Sensing, Technical University of Munich, Munich, Germany. His research interests include photogrammetry and remote sensing, high-accuracy image registration, and high-resolution satellite image processing.

Wei Yao received the B.S. degree in photogrammetry and remote sensing from Wuhan University, Wuhan, China, in 2003, and the Dipl.-Ing. (Univ.) degree in geodesy and geoinformation and the Ph.D. degree in photogrammetry and remote sensing from the Technical University of Munich (TUM), Germany, in 2007 and 2010, respectively.

Since 2007, he has been a Scientific Collaborator and Lecturer within the Institute of Photogrammetry and Cartography of TUM. Since 2011, he has been working as Senior Scientist in the research cluster of computer vision, remote sensing, and navigation at Munich University of Applied Sciences, Munich, Germany. In 2017, he was selected for National Thousand Young Talents Program of China and joined the Hong Kong Polytechnic University as Assistant Professor. He has already authored or coauthored nearly 70 academic articles in refereed international journals and conferences. His research interests include active remote sensing technology toward reconstruction and analysis of spatial-temporal behaviors of objects, image processing and analysis, machine learning, and related environmental and industrial applications.

Dr. Yao was the recipient of Best Student Paper Award from 2009 IEEE/ISPRS Joint Event on Urban Remote Sensing. Meanwhile, he was named a winner of the Chinese Government Award for Outstanding Self-Financed Students Abroad, which is granted around the world across all disciplines. He was also the recipient of Best Presentation Award of the International Symposium on Mobile Mapping 2013 and best paper awards of several IEEE/ISPRS conferences. Since 2016, he has been serving as Co-Chair of ISPRS WG III/6.

Rong Huang (Student Member, IEEE) was born in 1994. She received the B.E. degree in surveying engineering from Tongji University, Shanghai, China, in 2015, and the M.E. degree in earth oriented space science and technology (ESPACE) from Technical University of Munich, Munich, Germany, in 2018. She is currently working toward the Ph.D. degree in photogrammetry and remote sensing at the Department of Photogrammetry and Remote Sensing, Technical University of Munich, Munich, Germany.

Her research interests include point clouds processing, 3-D registration, and construction monitoring.

Xiaohua Tong (Senior Member, IEEE) received the Ph.D. degree in traffic engineering from Tongji University, Shanghai, China, in 1999.

He was a Research Fellow with Hong Kong Polytechnic University, Hong Kong, in 2006, and a Visiting Scholar with the University of California, Santa Barbara, CA, USA, between 2008 and 2009. He is currently a Professor with the College of Surveying and Geoinformatics, Tongji University. His research interests include remote sensing, geographic information system, uncertainty and spatial data quality, and image processing for high-resolution and hyperspectral images.

Ludwig Hoegner received the diploma in computer science and the Ph.D. degree from Technical University of Munich (TUM), Munich, Germany, in 2005.

Since 2006, he has been with the Department of Photogrammetry and Remote Sensing at Technical University of Munich (TUM). He is currently working as Lecturer with TUM and Technical University of Darmstadt, Darmstadt, Germany. He has authored or coauthored more than 60 scientific papers. His research interests include building photogrammetry, 3-D reconstruction, and object extraction from terrestrial and airborne cameras and laser scanners.

Dr. Hoegner is the Secretary of the ISPRS Intercommission Working Group II/III Pattern Analysis in Remote Sensing.

Uwe Stilla (Senior Member, IEEE) was born in Cologne, Germany, in 1957. He received the Dipl.Ing. degree in electrical engineering from Gesamthochschule Paderborn, Paderborn, Germany, in 1980, the Dipl.Ing. in biomedical engineering, and the Ph.D. (Doctor of Engineering) degree in pattern recognition from the University of Karlsruhe, Karlsruhe, Germany, in 1987, and 1993, respectively.

From 1990 to 2004, he was with the Institute of Optronics and Pattern Recognition, Ettlingen, Germany. Since 2004, he has been a Professor with the Technical University of Munich (TUM), Munich, Germany, and the Head of the Department of Photogrammetry and Remote Sensing. He was the Vice Dean of the Faculty of civil, geo, and environmental engineering, and is currently the Dean of Studies of the bachelor's and master's Programs Geodesy and Geoinformation, Earth Oriented Space Science and Technology, and Cartography. He has authored or coauthored more than 350 entries. His research interests include image analysis in the field of photogrammetry and remote sensing.

Dr. Stilla is the Chair of the ISPRS Working Group II/III Pattern Analysis in Remote Sensing, a Principal Investigator of the International Graduate School of Science and Engineering, the President of the German Society of Photogrammetry, Remote Sensing and Geoinformation, a member of the Scientific Board of German Commission of Geodesy, and a member of the Commission for Geodesy and Glaciology, Bavarian Academy of Science and Humanities, Munich, Germany.