IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received September 22, 2020, accepted October 3, 2020, date of publication October 13, 2020, date of current version October 27, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3030717

An Optimized Point Cloud Classification and Object Extraction Method Using Graph Cuts

BO GUO AND XIAOHAN ZUO
School of Civil and Transportation Engineering, Guangdong University of Technology, Guangzhou 510006, China
Corresponding author: Bo Guo (guobo@gdut.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 41704019 and Grant 41501499.

ABSTRACT The requirement for 3D scene classification and understanding has dramatically increased with the widespread use of airborne Light Detection And Ranging (LiDAR). This paper focuses on precise classification and object extraction based on point cloud data in complex scenes. There are usually gross errors in the initial classification based on locally independent classifiers, due to the over- and under-clustering in the feature extraction. We introduce a graph-cut based method to improve the classification precision and eliminate errors by using contextual information. The intuition behind our algorithm is based on the fact that nearby points have a high probability of belonging to the same object, while the distances of points belonging to different objects will be large. Based on this assumption, objects of interest can be precisely extracted, and we can thus optimize the classification results. The experiments undertaken in this study proved that the classification method we propose can be effectively used for point cloud classification in complex scenes.

INDEX TERMS LiDAR, point cloud classification, object extraction, graph cuts.

I. INTRODUCTION

LiDAR is an important and popular technology to acquire 3D dense and accurate information. The basic and most critical step is classifying the point clouds into categorical object instances (e.g. building, ground and vegetation) for further data processing [1]–[3].

A powerful mathematical tool for accurate perception of environments is machine learning that can be used to classify pixel-based image and point-based data [4]. Recently, with the development of deep learning, few works using deep learning on classifying point clouds have been studied [5], [6]. Various locally independent classifiers exist for classifying point clouds [7]–[9]. In these machine learning algorithms, such as AdaBoost and Support Vector Machine (SVM), feature vectors and corresponding training data labels are used to automatically learn the classification rules. Points, which are represented by features in locally classifiers, are merely a consequence of the discrete representation of objects. When the extracted features discriminative, good results can be produced. However, when the extracted features are noisy, this approach can produce noisy classifications. In areas of lower point density, feature vectors are unstable. Thus, the local features containing a limited amount of information, are often not sufficient to determine the correct object label. Feature vectors might be unstable at the boundaries. Thus, the classification results obtained using local classifiers are prone to many misclassification problems, most notably to ''leaking'' through weak spots in object boundaries [10]. For further object modeling, we need to improve the precision of the rough classification results by post-processing [11]. Recently, by treating point set as unordered sets, some deep learning methods also use local based features for classification [5].

In addition to local features which are mainly geometric primitive cues [12]–[14], high-level contextual information can be used to extend classification and object extraction. Regions, boundaries, etc., are effective high-level contextual information [15], [16]. By combing local and contextual information, Random Field (RF) based model is built for generating rational classification results [17]. But, global optimization of the RF models combining local features and contextual information is usually NP-hard [18]. It only can be solved efficiently in some special cases [19], [20]. Additionally, the classical Potts model used in Conditional Random Field (CRF) makes it favors smooth object boundaries [21]. Thus, it may fail at object boundaries [22]. Recently, deep learning based methods used multi-scale and hierarchical

The associate editor coordinating the review of this manuscript and approving it for publication was Qiangqiang Yuan.

information to build contextual models [6]. However, the spatial relations and high-level information used is not readily comprehensible.

In this paper, for improving classification accuracy, we introduce noticeable high-level relation that neighboring points of an object are more reasonable to be segmented together (within the same region), and different objects should be divided with gaps (boundaries). By combining region and boundary information, we can optimize the classification results and extract objects precisely by adopting the minimization of an energy function using graph cuts [23]–[25]. We introduce an algorithm for which the intuition behind it is that a good foreground segmentation consists of points that are well connected to each other (belonging to an object), but poorly connected to the background. This is similar to the segmentation methods. However, the segmentation studies that do exist mostly focus on the extraction of geometric primitives of parts rather than entire objects [26]. Thus, based on this approach, we describe a fairly general process using region and boundary properties, which are combined with the rough classification results to generate optimized classification results.

Contextual information, such as the co-occurrence and geometric relationships between objects [10], [27], is usually used in Markov random field (MRF) or CRF based methods. However, processing all this kind of contextual information together is a tough task. The inference task of CRF is NP-hard [28]. It usually assigns a high potential value for point cloud classification if all points are labeled the same, and zero otherwise with Potts model and based on approximate inference using α-expansion or $\alpha - \beta$ swap through graph-cuts. However, in many applications, it is difficult to obtain a group of sites (clique) that contain homogeneous labels. Thus, the contextual information used in CRF or MRF is restricted. In this way, a rounding procedure is performed that achieves a rough approximation, rather than a robust one. In the discussion section (Section IV), we compare the results of CRF with those of our proposed method.

In the proposed method, we combine three types of information (point, region, and boundary) to generate robust point classification and object extraction results by the use of graph cuts to find the nearly global optimization of an energy function.

This optimized classification and object extraction method can generate precise results, which can therefore be used for further modeling. The main contribution of this paper are:

(1) An efficient energy cost function is constructed for the optimization. By combining region and boundary information, we introduce a well-defined energy function. By minimizing this energy function using graph cuts, nearly global optimization of the classification can be precisely generated.

(2) Hard constraint setting simplifies the optimization process. We introduce a highly specialized process to automatically set hard constraints of the foreground and background. In complex examples, it is necessary to control the optimization properties by setting up appropriate hard constraints (seeds) to further constrain the search space of the possible solutions.

II. METHODOLOGY

The rough classification obtained using the classifiers based on local features usually contain errors [1]. For each object of interest, its neighbor may contain many misclassified points in the rough classification results. Classification results can be optimized by using object-level information (region and boundary). As described in the introduction, within a region, neighboring points belong to a same object with a strong possibility, and points belonging to different objects are usually separate from each other (divided by boundaries). Thus, the combined region and boundary information can be used to optimize the classification results and precisely extract objects. We impose an energy function based on the region and boundary properties:

$$E(A) = \lambda R(A) + B(A) \tag{1}$$

where A represents set of labels of the set of data elements P, $R(A)$ is the regional term, and $B(A)$ is the boundary term. We formulate the optimized point cloud classification and object extraction as minimizing an energy function problem using graph cuts by enforcing spatial coherence while preserving appropriate sharp discontinuities.

A. OPTIMIZATION MODEL

There are many objects in a typical scenario, and processing all the objects together is a tough task. Usually, approximate approaches [18] based on α-expansion or $\alpha - \beta$ swap was used, but cannot find global solution for multi-dimension optimization. For simplification, we first process each object individually. Then, by combining the results of all the objects, the final classification results are obtained.

By combining the region and boundary information with the initial classification results, we formulate such an optimization problem as a binary (foreground/background) labeling problem for each object by adjusting the labels of its neighbors' misclassified points (Fig. 1). Thus, A is the labels

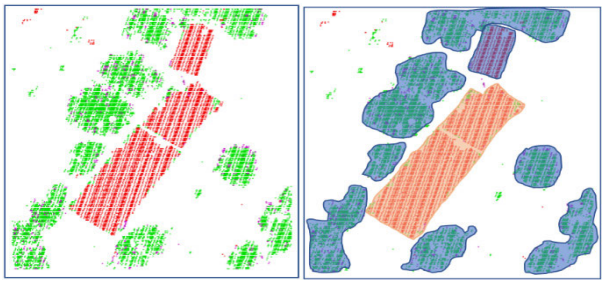

FIGURE 1. We optimize the initial classification (a) by formulating a binary (foreground/background) labeling problem (b); the foreground object is a building, and the other objects (mainly vegetation) are defined as background. After optimization, errors can be eliminated by adjusting the labels of the points.

of the foreground or background for the object of interest, which are not the labels of the point class.

Considering a point set P, we use all pairs $\{p, q\}$ of neighboring points in P to represent a neighborhood system. We set $(A_1, \ldots, A_p, \ldots, A_{|P|})$ to be a binary vector. each component A_p of A defines the assignment (foreground/background) to point p in P.

By using binary optimization to enforce spatial coherence while preserving appropriate sharp discontinuities, we formulate the optimized point cloud classification and object extraction as globally minimizing an energy function (1) through a graph-cut algorithm which is an efficient optimization tool [23], [29].

Finally, we adjust the optimized foreground points in P to their specific class and generate optimized classification results.

B. ENERGY FUNCTION

As seen in (1), by combining the region and boundary information, the energy function has two parts:

1) REGIONAL PART

As described before, when using a classifier based on local features, there may be many local classification errors, but no major errors. The region information extracted from the initial classification results thus can be used to assist designing highly specialized algorithm for contextual classification [30].

Objects by clustering points with the same labels are extracted first. Using the initial classification results, we first group nearby points with same labels by using region growing algorithm. The size of the points in the region is used to differentiate the objects as stable and unstable ones. Stable objects generally have large sizes, in which the main points are correctly classified. Otherwise, tiny objects are considered to be unstable (with classification errors).

The threshold for the stable objects is closely related to the size of its initial target classes. Only stable objects can be regarded as either foreground or background (shown in Fig. 2).

The regional part $R(A)$ is the combination of individual penalty for specifying each point p to a certain label A_p (''foreground'' or ''background'') with $R_p(A_p)$ using the regional information. Thus, the regional part is defined as $R(A) = \sum_{p \in P} R_p(A_p)$. R_p measures the penalty of how probable for point p in a region by evaluating the distance. Given the foreground and background region seed points (within stable objects), we use the following function to define the regional term:

$$\begin{cases} R_p(\text{"}bkg\text{"}) = 1 - e^{-\left(\frac{d_p^b}{\delta}\right)^2} \\ R_p(\text{"}obj\text{"}) = 1 - e^{-\left(\frac{d_p^o}{\delta}\right)^2} \end{cases} \tag{2}$$

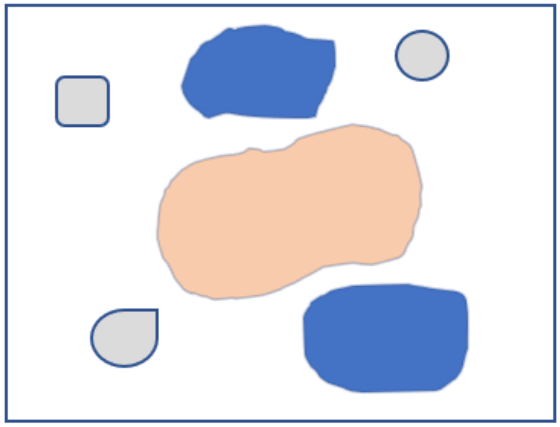

FIGURE 2. Sketch map of stable objects. The foreground object is shown in yellow, and the background objects are shown in blue. Unstable objects are shown in gray. Only one stable object is treated as foreground, and there may be many other stable objects are treated as background.

where d_p^b is the closet distance of point p to the background seed points, and d_p^o is the closet distance of point p to the foreground seed points. δ is the average distance of the points in the dataset. There may be multiple background objects, thus d_p^b is defined as the closet distance of point p to seed points of all background objects.

The marginal points of regions usually contain errors and entangle with points with other regions. Thus, a penalty P is added to strongly punishment misclassified points belongs to other regions to be labeled as foreground. We make a penalty function out of the distance r to center location of object.

$$P = \begin{cases} 1, & r < r_0 \\ \frac{r}{\mathrm{r}_0}, & r > \mathrm{r}_0 \text{ and } r < \boldsymbol{\alpha} \times \mathrm{r}_0 \\ \boldsymbol{\alpha}, & r \geq \boldsymbol{\alpha} \times \mathrm{r}_0 \end{cases} \tag{3}$$

where r_0 is set to be an initial radius which contain 90% points in the stable foreground object. Given a distance r, a soft foreground penalty P is created to punishment marginal misclassified points. If the radius r is less than r_0, the penalty P is set to 1. The penalty then increases linearly if the distance gradually enlarges when $r < \alpha \times \mathrm{r}_0$. Finally, when the distance extends to a certain level ($r \geq \alpha \times \mathrm{r}_0$), the penalty stays at a certain stable value α.

The total regional term with penalty is defined as:

$$\begin{cases} R_p^p(\text{"}obj\text{"}) = P \times R_p(\text{"}obj\text{"}) \\ R_p^p(\text{"}bkg\text{"}) = R_p(\text{"}bkg\text{"}) \end{cases} \tag{4}$$

The regional term of the background $R_p(\text{"}bkg\text{"})$ is not penalized.

2) BOUNDARY PART

The boundary term adds regularizing constraints that help to enforce the spatial coherence between neighboring points $\{p, q\} \in N$ by minimizing the discontinuities between them.

The boundary term is defined as:

$$B(A) = \sum_{\{p,q\} \in N} B_{p,q} \delta_{A_p \neq A_q} \tag{5}$$

where we define $\delta_{A_p \neq A_q} = \begin{cases} 1 \text{ if } A_p \neq A_q \\ 0 \text{ if } A_p = A_q \end{cases}$.

$B_{p,q}$ penalizes a lot for discontinuities between points ($\delta_{A_p \neq A_q}$) with a close distance. $B_{p,q}$ is large when points p and q are close, and $B_{p,q}$ is close to zero when the distance between the two points is very large. We do not use fine-scale geometrical cues such as curvature in discontinuities, but rather rely on the distance between points $d_{p,q}$ and the average distance δ of points in the dataset. We define $B_{p,q}$ as:

$$B_{p,q} = \exp\left(-\left(\frac{d_{p,q}}{\delta}\right)^2\right) \tag{6}$$

3) GRAPH CONSTRUCTION

As described in [31], [32], graph-cut based methods can be used to minimize energy function (1). We first have to construct a graph $G = \langle = V, E \rangle$ representing the spatial distribution of points in point cloud dataset. Neighbor points with closer distances are connected more strongly in graph G.

V is a set of nodes in graph G. When used for the point cloud classification optimization, the nodes V are represented by an arbitrary set of points P. Two special terminal nodes (source S and sink T) are also added to the node set $V = \{S, T\} \bigcup P$. In our point cloud graph construction, the special terminal nodes S and T are extracted automatically based on the rough classification results and region growing rule (described in Section II- B).

E is a set of undirected edges that interconnect the points and is constructed using a neighborhood system. Specifically, a k-nearest neighbor (KNN) graph is constructed on the input points to build an edge system. Because of the disarray and sparsity, an R-tree [33] is used to index points and search the neighboring points in the point cloud. Then, a set N of edges by pairs $\{p, q\}$ of the neighboring points are found. In the neighborhood system, each graph edge e is assigned with certain nonnegative weight w_e, the value of which is based on the length of the edge. Finally, A graph representing the structure of the point cloud is constructed, where the closer points are more strongly connected.

Terminal nodes S and T are absolutely partitioned by a s-t cut which is a subset of edges $C \subset E$. Then the complete graph G become an induced graph $G(C) = \langle V, E \backslash C \rangle$. The cost of a cut is defined as the sum of the costs of the edges that it severs $|C| = \sum_{e \in C} w_e$. By construct a graph from the input point cloud, the problem of minimizing an energy function can be equal to find a min-cut/max-flow using graph cuts [23].

There are two kinds of undirected edges E: n-links and t-links. Each n-link connects two neighboring points $\{p, q\}$ by using KNN system. t-links connect point p to special terminal nodes S and T to construct link $\{p, S\}$ and link $\{p, T\}$. The edges' weights are given as follows:

(1) The weight of the n-link:

$$w_{\{p,q\}} = B_{p,q} \tag{7}$$

(2) The weight of the t-links:

$$\begin{cases} w_{\{p,S\}} = \lambda R_p \text{ ("}bkg\text{")} \\ w_{\{p,T\}} = \lambda R_p \text{ ("}obj\text{")} \end{cases} \tag{8}$$

We treat λ as a hyperparameter. Typically, it is set to 1.

The weights are defined based on the regional and boundary terms. We compute the best cut that would give an ''optimal'' solution which precisely minimizes the energy function. Further details can be found in [23]. The object and boundary properties can thus be reinforced for an optimal solution.

4) OPTIMIZATION PROCESS

The previous section described how an energy function is constructed for optimized classification, and how the function can be efficiently solved using graph cuts. The object and boundary properties are used in the optimal classification to encourage separation from foreground to background.

This section describes the details of the optimization process (shown in Fig. 3). The optimization is processed for every stable object and its adjacent points (containing sufficient background seeds) in succession. For each foreground object, this optimization is a binary labelling method.

Based on the initial classification results, region growing is first used to gather point regions with the same label. For every stable object, the optimization includes the following steps.

A neighborhood is constructed for simplifying the graph construction. For a stable object being processed, a minimum 3D sphere is firstly set up to contain the points in its region. Then, the sphere is widened to contain the neighboring points of the stable object being processed. By temporally eliminating other points which have not been processed recently, this neighborhood is efficient for graph construction.

Hard constraints of the seed points have to be set for the optimization. The seed points of foreground are set to be the points in the stable object being optimized. Otherwise, the seed points of background are defined as points in other stable objects in the widened 3D sphere. The inaccuracy of hard constraints can be penalized by adding a penalty.

After establishing the above neighborhood and hard seed constraints, a graph for which the edges have weights that decrease with distance is generated. We then implement classification optimization using the graph-cut algorithm. Finally, the points of unstable objects that are segmented into foreground are optimized.

For one stable object, the precise extraction of its points is beneficial for the further modeling. By iteratively processing all the stable objects in a scene, we turn the binary labelling problem to a multiple labeling problem and the classification results can be optimized.

The above described process of optimization is shown in Fig. 3

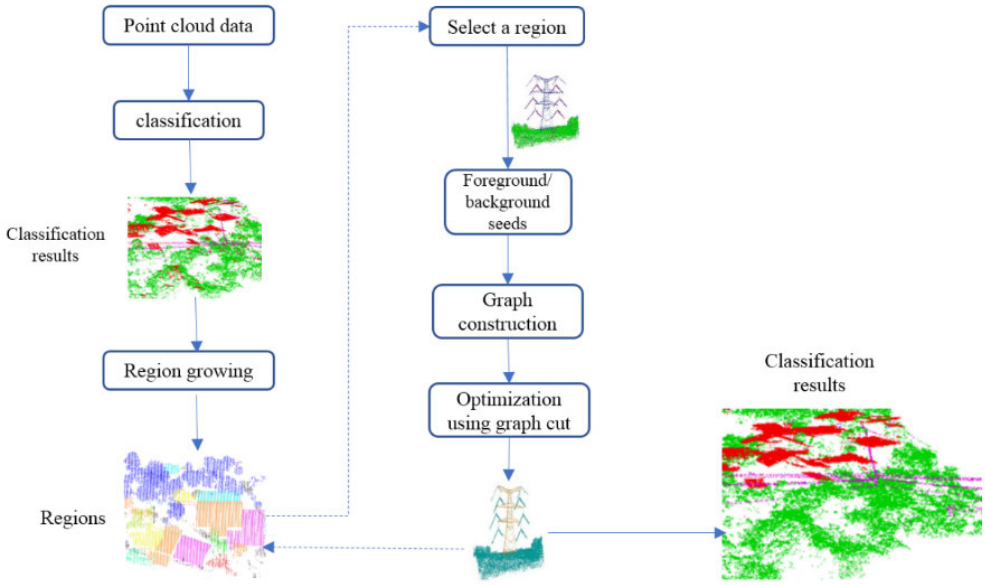

FIGURE 3. The optimization process.

III. EXPERIMENTAL RESULTS

Precise object extraction is important for further modeling, which can be badly disturbed by noise or misclassified points. We propose a method to optimize the classification results. In this section, we evaluate our optimized method on several examples. The results were obtained using two datasets.

The first dataset (Scene I) was provided by an airborne laser scanner Reigl Q560. This dataset was used for power-line management and acquired in Hubei province, China. There are many power facilities, such as pylons and power-lines, and vegetations in Scene I. The point density varies greatly and is approximately 10-30 points/m^2 due to the complex topography.

The second dataset (Scene II) was provided by Optech, Inc., and was used by the ISPRS Test Project on Urban Classification, 3-D Building Reconstruction and Semantic Labeling [34]. The point density is approximately 10 points/m^2. There are a small number of trees and many skyscrapers in Scene II where is a commercial district.

The rough classification results by using the local-based classifier are first shown. We then describe the results obtained by the method of automatic classification optimization proposed in this paper. Finally, some examples of object extraction and precise classification are provided.

A. CLASSIFICATION RESULTS BY USING LOCALLY CLASSIFIER

The JointBoost algorithm using local features [1] was used for classifying point cloud data. Before the classification process, we first filtered out the ground points in the dataset using Terrasolid software [35]. In order to avoid classification fault caused by imbalance of training samples, we randomly selected equal size of samples per class from the training dataset. After calculating the local features, we trained a JointBoost classifier [36] for classifying point cloud.

The points in Scene I were classified into pylon, building, vegetation, and power line. As shown in Fig. 4, there are no major errors in typical classification results.

There are a small number of trees and many skyscrapers in Scene II where is a commercial district. Thus, the points in Scene II were classified into building and vegetation. Typical results are shown in Fig. 5.

Using the classifier based on local features which cannot be precisely extracted, especially on the boundaries of objects, many typical misclassified points are exhibited in the initial classification results. Firstly, some points of building edges are misclassified as vegetation (Fig. 6(a)). Because points of building edges distribute scatteredly. Some misclassified points have vegetation characteristics due to the normal of these points in a neighborhood region varies greatly. Due to a scattered distribution, incorrect classification of vegetation points as building is also typical (Fig. 6(b)). When height difference is small and point density is high, some vegetation points may be misclassified as building. Some mislabeled points of power lines and vegetations appear widely in pylons (Fig. 6(c)).

B. IMPLEMENTATION DETAILS OF THE OPTIMIZATION

The workflow of the classification optimization process was described in Section II. We describe the implementation details in this section.

1) REGION INFORMATION EXTRACTION

Region information is important for energy function construction and classification optimization. Region growing is first used to detect and gather stable objects which contain

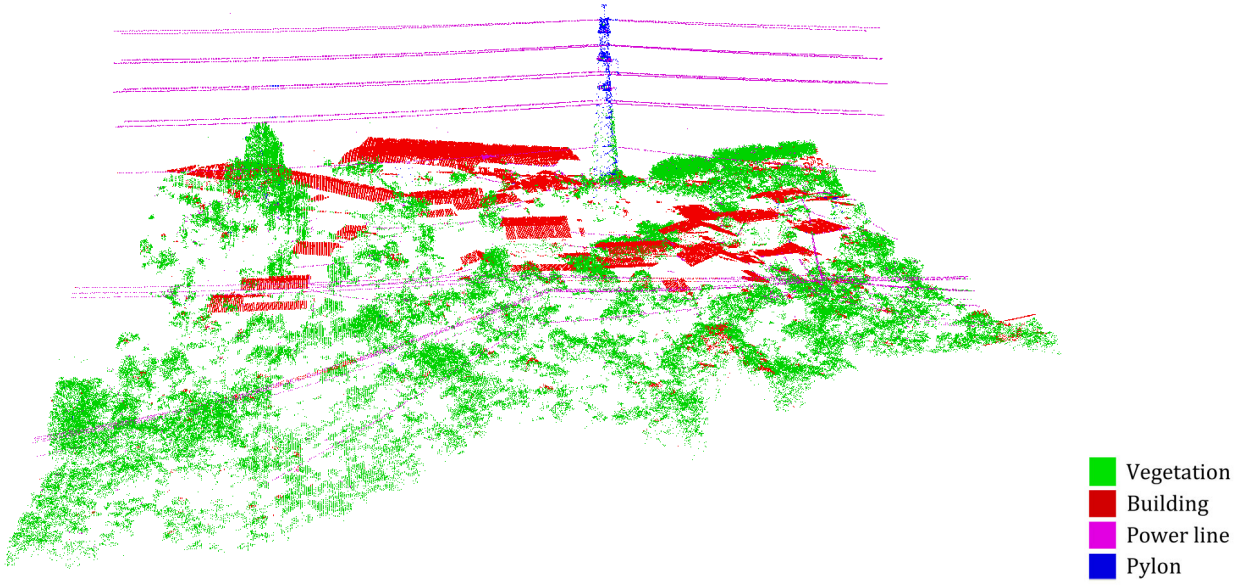

FIGURE 4. Classification results of Scene I (red: building, green: vegetation, violet: power-line, blue: pylon).

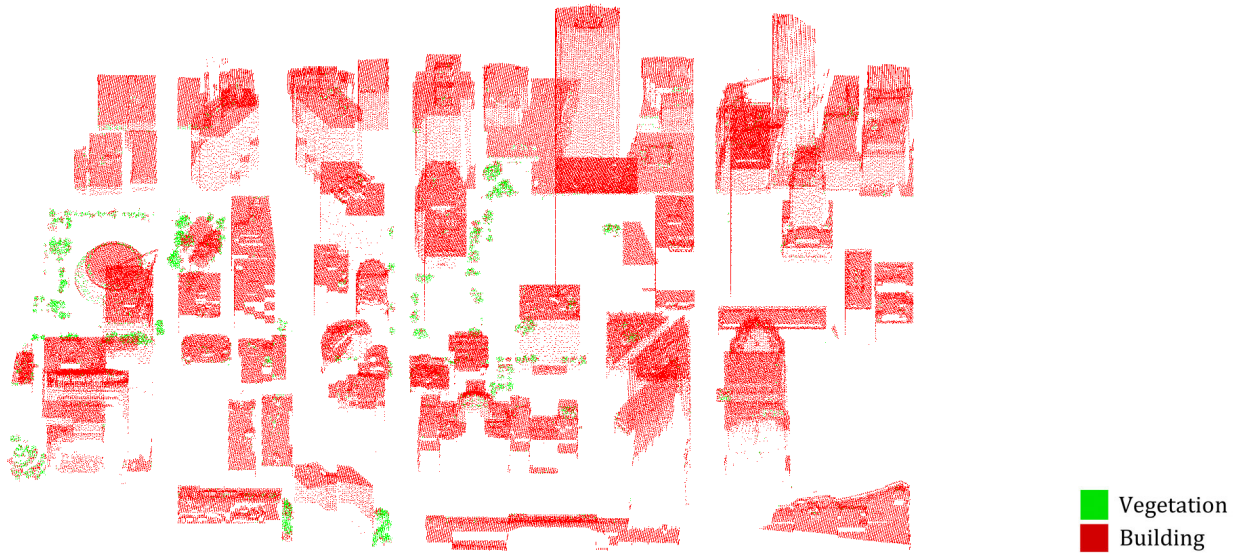

FIGURE 5. Classification results of Scene II (red: building, green: vegetation).

points with a high probability of correct class labels. Region growing is used to gather neighboring points by determining whether the neighboring points should be added to the region. As a general algorithm of data clustering, region growing process is iterated to searching which neighboring points should be merged.

Starting from selecting seed point set with same classification label. These seeds construct an initial region. The region then grows by adding adjacent points, following two criteria: 1) the distance of adjacent points should be within a certain range; and 2) the labels of adjacent points should be the same.

A 26-connected neighborhood is used to keep examining the adjacent points of seeds within a certain distance. The distance range used for the region growing is based on the average distance δ of points in the dataset ($1.5\times\delta$). If the points have the same classification label as the seed points, they are classified as new seed points. This is an iterative process until there is no change in two successive iterative stages.

Stable regions are defined as objects with the major points having correct classification labels. While tiny objects are unstable, we generally consider stable objects have large sizes. Only stable objects can be regarded as foreground or

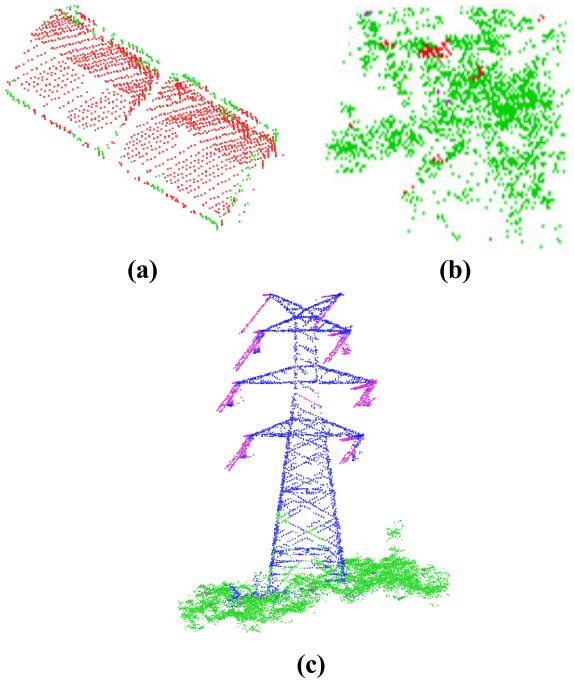

FIGURE 6. Typical misclassification errors. (a) misclassified points in building margins (mainly misclassified as vegetation); (b) misclassified vegetation points (mainly misclassified as buildings); (c) misclassified pylon points (mainly misclassified as power-lines and vegetations).

TABLE 1. The distance threshold for stable objects/m.

	Trees	Buildings	Power lines	Pylons
Scene I	5.2	8.1	10.0	6.2
Scene II	5.4	12.2	--	--

background. Because the size threshold is difficult to calculate, we use the distance threshold to determine stable objects. The distance threshold (Table 1) is defined as the largest distance of the points in a region. Training dataset is used to estimate the distance threshold for each class (Tab. 1).

The region growing results are shown in Fig. 7. Stable objects are shown in various colors, while unstable objects are shown in black.

After the region growing, each stable object is iteratively selected to be optimized.

2) HARD CONSTRAINT SETTING

Seeds are hard constraints. B and O denote the seed subsets which are *a priori* of a part of "background" and "foreground". For precise object extraction, it is beneficial to obtain the seed of foreground/background as accurately as possible.

Boundary points usually contain errors. In order to eliminate wrong points in the hard constraints, only points with correct class labels in stable objects should be selected.

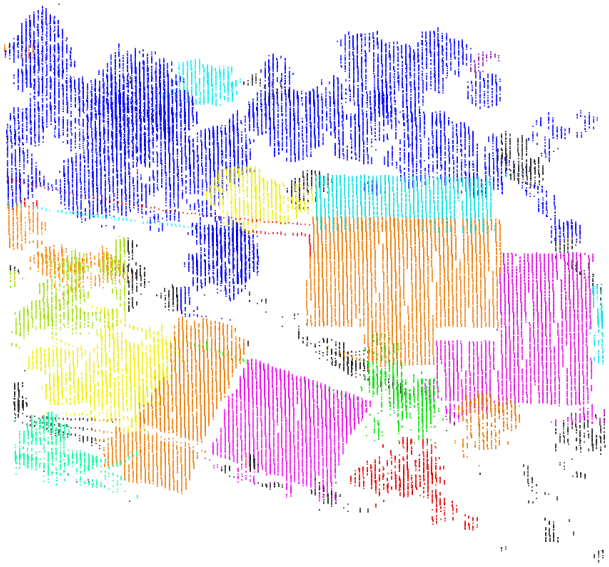

FIGURE 7. Region growing results. Stable objects are shown in different colors, while otherwise unstable objects are shown in black.

However, this strategy is difficult to implement. In the proposed method, a penalty is added to strongly encourage points in the object margin with errors not to be labeled as foreground. In order to select the optimal penalty function (3), we make the penalty as a function of the distance r to the central location of the object.

The distance threshold r_0 is set to the radius of stable objects. The setting of parameter α is described in Section IV. In this paper, α is set to 1.1.

3) OPTIMIZATION

By setting the hard constraints of an object, we have to optimize the labels of its neighboring points. The main steps are:

a: GRAPH CONSTRUCTION

The neighborhood should be first set for the graph construction. For a processing stable object, a minimum 3D sphere is set up to contain the points of selected object. Then, we expand the sphere by 3.0 times to hold all neighboring points.

We construct a KNN graph on the input points to construct the edge system. The value of k is important. If k is too small, the neighborhood system cannot represent the structure of the point cloud. Otherwise, an overlarge k value will greatly influence the computational efficiency. We use the 26-neighborhood system for graph construction.

b: OPTIMIZATION USING GRAPH CUTS

After constructing the energy function and setting the graph structure of the dataset, and given the constraints of the foreground and background seed points, the optimization problem is formulated as a binary labeling problem to be

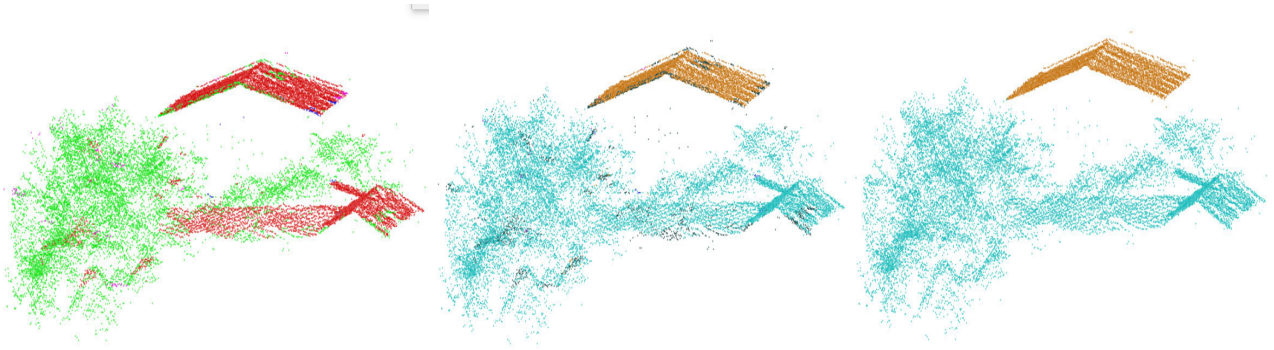

FIGURE 8. Object extraction for a building. (a) Initial classification based on JointBoost (red: building, green: vegetation, violet: power-line, blue: pylon); (b) Seed points: foreground/yellow background/blue; (c) Object extraction results: foreground/yellow background/blue.

solved using graph cuts [23]. After implementing min-cut C, if $\{p, T\} \in C$, we set A_p= "obj". Otherwise, if $\{p, S\} \in C$, we set A_p= "bkg".

C. OPTIMIZED CLASSIFICATION AND OBJECT EXTRACTION RESULTS

1) OBJECT EXTRACTION

For a single object, the further precise modeling [37], [38] is dependent on the accuracy of the classification. Using the method described previously, we first examine the extraction results of some typical objects.

The initial classification usually contains errors, which should be removed for further processing. For a stable object, the precise extraction of its points is beneficial for the modeling.

Given a stable region, we first calculate a 3D sphere to hold all points. The neighborhood is then widened to contain neighboring points of other stable objects. Hard constraints of seed points have to be set for optimization. The foreground seeds are set to be the points of processing stable object. While, the other stable objects' points in the neighborhood are regarded as background seeds. A graph is formed in which the edges have weights that decrease with the distances. By using a min-cut algorithm, the optimization of energy function (1) can be solved.

The object extraction results are shown in Fig. 8. It can be seen many local errors have been eliminated and the object extraction are smooth in the final results (c). The initial classification (a) ausing local features and JointBoost [1] usually contain many noise points. Part (b) shows the segmentation seeds setting. The building is stable object being processed. Thus, the foreground seed points are selected by using building points, while the background seed points are using other stable objects (vegetation, other building).

2) OPTIMIZED CLASSIFICATION

By optimizing the classification of all the stable objects as foreground in a scene, the points that are adjusted to the foreground are reclassified to the labels of the corresponding stable objects. By adjusting the error points of all the stable objects in the test region, the initial classification results are optimized.

The proposed optimization method generates smoother results than the initial classification. After the optimization, some error points caused by classification using local features will have been rectified.

Fig. 9 shows the optimized classification results of Scene I. The optimized classification results are smooth, and many tiny local errors have been eliminated. The typical objects, such as pylons and buildings, are precisely delineated.

It is shown in Fig. 10 the optimized classification results of Scene II. It can be seen that the typical buildings are mainly correctly classified. The digital city modeling thus benefits from the precise classification.

IV. DISCUSSION

In the following, we discuss the comparation of the results obtained with the proposed method with those obtained using other methods. Method I [1] was point cloud classification method using JointBoost which is a locally classifier, based on carefully hand-desighed features. Contextual information, such as the co-occurrence and geometric relationships between objects, is used in CRF. CRF can generate more accurate classification results than classifiers only using local features. However, the contextual information used in CRF (Method II) is restricted [39], and it is difficult to obtain a group of points (clique) that contains homogeneous labels in many applications. Thus, there are also many local errors.

Classification quality is usually represented by precision/recall. If more relevant results are returned than irrelevant results, it is means high precision is obtained. High recall means that a classification method returned mostly relevant results. The precision/recall, together with accuracy of the different methods is shown in Table 2. The highest precision and recall have been achieved by using our proposed method in Scene I and II. The classification precision for the power lines in Scene I obtained using the proposed method is much

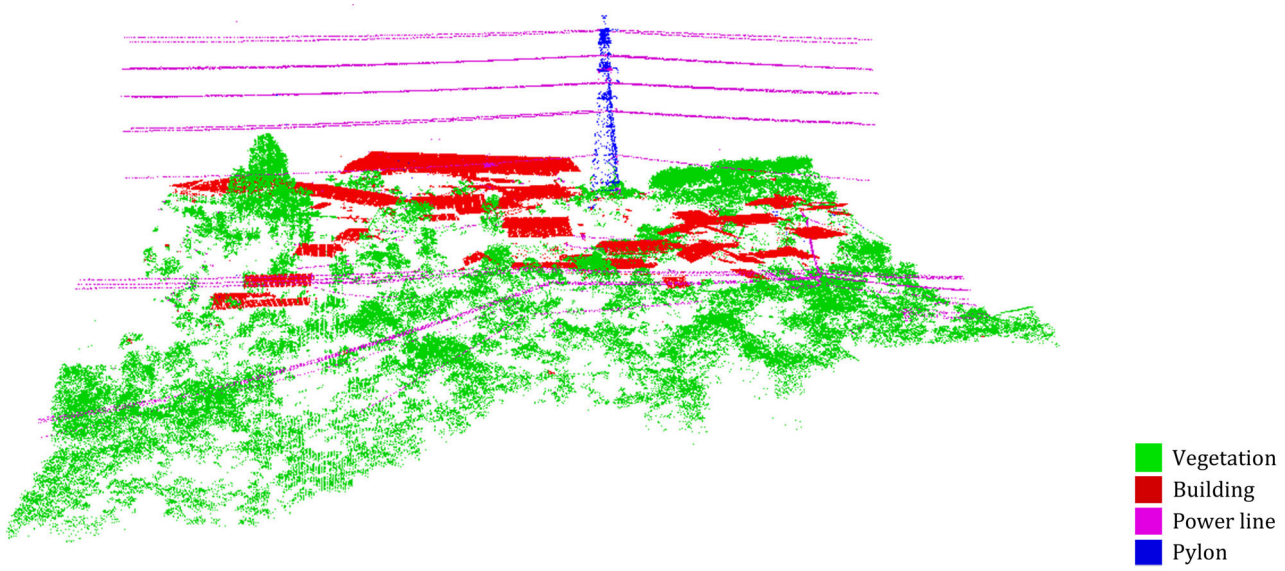

FIGURE 9. Optimized classification results for the test dataset of Scene I. (red: building, green: vegetation, violet: power-line, blue: pylon).

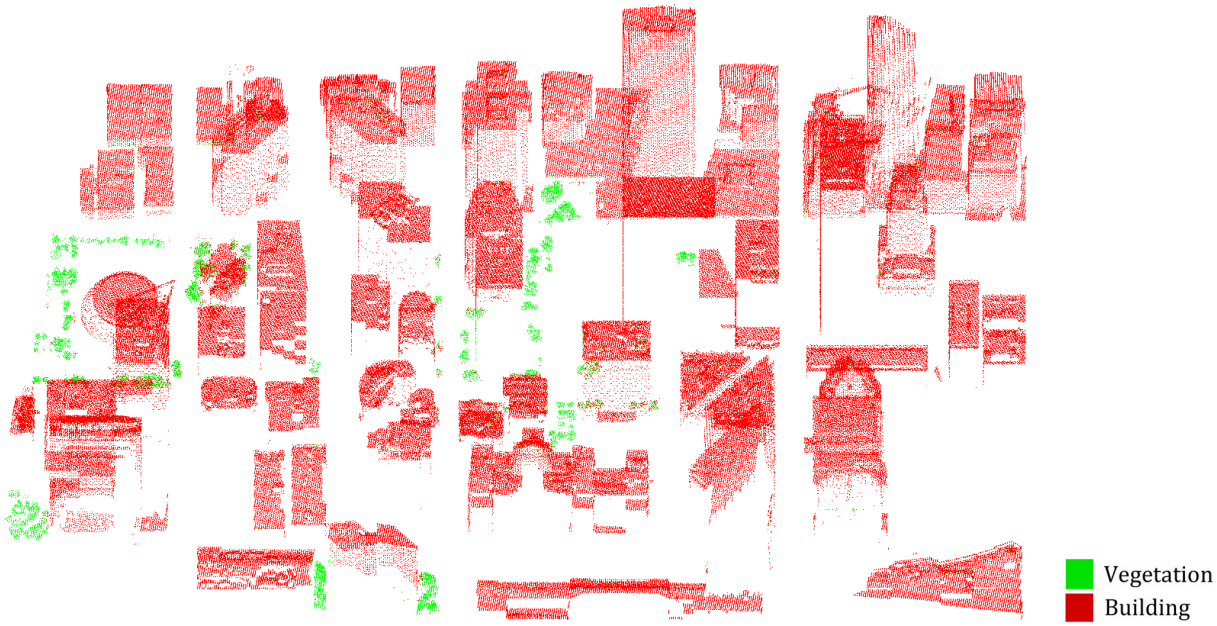

FIGURE 10. Optimized classification results for the test dataset of Scene II (red: building, green: vegetation).

higher than that obtained by the other methods. Therefore, combining region and boundary features can better describe the characteristics of the objects.

The proposed algorithm runs in an automatic regime. In order to achieve optimized classification results, some hard constraints for the foreground/background seeds need to be made. In this paper, we only consider stable objects for optimized classification. Thus, some tiny objects and connected objects cannot be well processed.

Parameter sensitivity of our proposed model should be Analyzed.

F_1 measure (9) is a compound metric considering both the precision and the recall to represent the classification accuracy:

$$F_1 = 2\frac{Rc \times Pr}{Rc + Pr} \tag{9}$$

where Rc is recall and Pr is precision.

The value of α for penalty term in function (3) is crucial for classification results. By setting different value, F_1 measure is used to analyze the impact of the parameter (shown in Fig.11).

TABLE 2. The precision/recall and accuracy of the classification results.

Scene I	Trees (%)	Buildings (%)	Power lines (%)	Pylons (%)	Accuracy (%)
Our proposed method	94.1/94.5	95.3/94.3	90.5/87.2	92.6/89.0	94.5
Method I	92.2/90.0	90.9/90.1	84.3/86.5	93.9/73.6	90.1
Method II	93.1/91.3	94.7/90.6	89.3/86.6	91.9/81.4	92.5
Scene II	Trees (%)	Buildings (%)		Accuracy (%)	
Our proposed method	94.6/94.1	96.4/95.8		95.9	
Method I	92.1/89.7	93.4/94.6		92.3	
Method II	93.6/92.5	95.1/93.6		93.7	

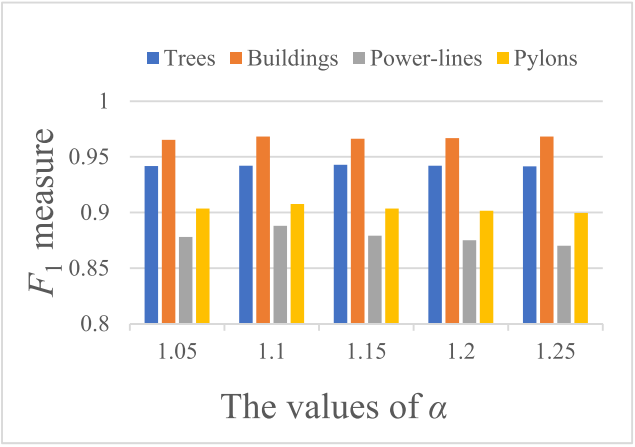

FIGURE 11. The influence of different values of α on the classification results.

To determine the proper value of α in penalty term, various values of α, i.e., 1.05, 1.1, 1.15, 1.20, and 1.25, are set to test the influence on the classification. We mainly tested it on Scene I. As illustrated in Fig. 11, the F_1 values of buildings and vegetation are both greater than 0.94. While those of power lines and pylons are small. Finally, power line has smallest F1 value. Even though, that of power lines is always larger than 0.87. If we set $\alpha = 1.1$, the F_1 value of power lines is the highest.

V. CONCLUSION

The automatic classification of point cloud data is of importance for scene understanding and mapping. However, if there are gross errors in the initial classification, further tasks such as modeling, etc., will be critically influenced. Optimized classification and precise object extraction should thus be achieved. A graph-cut based optimization technique has been introduced in this paper to improve the initial classification results generated by local classifiers. In addition to local features, effective high-level contextual information (such as regions, boundaries, etc.) is helpful for precise classification. By using the region and boundary properties, we describe a fairly general process combining the rough classification results to generate optimized classification results. Finally, precise object extraction and classification can thus be generated.

In practice, real-world data are noisy and some objects may be entangled with others. Only using boundary and region information may fail in some extreme cases. Thus, no algorithm can guarantee 100% accuracy for classification and object extraction. There are also some limitations to the method proposed in this paper:

(1) Some tiny objects may be ignored. The optimization uses reliable objects as seed points by the use of region growing. Thus, some tiny objects are not processed further. As a result, the rough classification errors in these tiny objects cannot be corrected.

(2) Connected objects, especially with similar classes, cannot be separated precisely. Some connected objects will be entangled with others, and thus these objects cannot be separated precisely using only region growing. The hard constraint setting for the connected objects should thus be improved.

There are also some shortcomings in the computational efficiency. Thus, this method may not be suitable for fast or real-time computing.

In the future, we will introduce multi-scale or stacked hierarchical labeling skills [40]–[42] to further improve the classification precision. Objects with different scales are sometimes difficult to precisely extract with information of only one level. Thus, combining low and higher-level information by using Deep learning is a reasonable way to further improve the classification results.

REFERENCES

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

[2] J. Jung, E. Che, M. J. Olsen, and C. Parrish, "Efficient and robust lane marking extraction from mobile lidar point clouds," *ISPRS J. Photogramm. Remote Sens.*, vol. 147, pp. 1–18, Jan. 2019.

[3] J. Zhang and X. Lin, "Filtering airborne LiDAR data by embedding smoothness-constrained segmentation in progressive TIN densification," *ISPRS J. Photogramm. Remote Sens.*, vol. 81, pp. 44–59, Jul. 2013.

[4] Y. Lin, F. Lv, S. Zhu, M. Yang, T. Cour, K. Yu, L. Cao, and T. Huang, "Large-scale image classification: Fast feature extraction and SVM training," in *Proc. CVPR*, Colorado Springs, CO, USA, Jun. 2011, pp. 1689–1696.

[5] R. Q. Charles, H. Su, M. Kaichun, and L. J. Guibas, "PointNet: Deep learning on point sets for 3D classification and segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 652–660.

[6] 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. Adv. Neural Inf. Process. Syst.*, 2017, pp. 5099–5108.

[7] S. K. Lodha, E. J. Kreps, D. P. Helmbold, and D. Fitzpatrick, "Aerial LiDAR data classification using support vector machines (SVM)," in *Proc. 3rd Int. Symp. 3D Data Process., Vis., Transmiss. (DPVT)*, Chapel Hill, NC, USA, Jun. 2006, pp. 567–574.

[8] H. B. Kim and G. Sohn, "Random forests based multiple classifier system for power-line scene classification," in *Laser Scanning*. Calgary, AB, Canada: ISPRS, 2011.

[9] S. Xu, G. Vosselman, and S. Oude Elberink, "Multiple-entity based classification of airborne laser scanning data in urban areas," *ISPRS J. Photogramm. Remote Sens.*, vol. 88, pp. 1–15, Feb. 2014.

[10] D. Munoz, J. A. Bagnell, N. Vandapel, and M. Hebert, "Contextual classification with functional max-margin Markov networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 975–982.

[11] F. Lafarge, X. Descombes, J. Zerubia, and M. Pierrot-Deseilligny, "Structural approach for building reconstruction from a single DSM," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 1, pp. 135–147, Jan. 2010.

[12] H. Zheng, R. Wang, and S. Xu, "Recognizing street lighting poles from mobile LiDAR data," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 1, pp. 407–420, Jan. 2017.

[13] P. J. Besl and R. C. Jain, "Segmentation through variable-order surface fitting," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 10, no. 2, pp. 167–192, Mar. 1988.

[14] T. Rabbani, F. Van Den Heuvel, and G. Vosselmann, "Segmentation of point clouds using smoothness constraint," *Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci.*, vol. 36, no. 5, pp. 248–253, 2006.

[15] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, pp. 1137–1146, 2017.

[16] L. Ladicky, C. Russell, P. Kohli, and P. H. S. Torr, "Associative hierarchical CRFs for object class image segmentation," in *Proc. IEEE 12th Int. Conf. Comput. Vis.*, Sep. 2009, pp. 739–746.

[17] H. Luo, C. Wang, C. Wen, Z. Chen, D. Zai, Y. Yu, and J. Li, "Semantic labeling of mobile LiDAR point clouds via active learning and higher order MRF," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 7, pp. 3631–3644, Jul. 2018.

[18] 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.

[19] H. Ishikawa, "Exact optimization for Markov random fields with convex priors," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 10, pp. 1333–1336, Oct. 2003.

[20] V. Kolmogorov and C. Rother, "Minimizing nonsubmodular functions with graph cuts—A review," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 7, pp. 1274–1279, Jul. 2007.

[21] P. Kohli, L. Ladický, and P. H. S. Torr, "Robust higher order potentials for enforcing label consistency," *Int. J. Comput. Vis.*, vol. 82, no. 3, pp. 302–324, May 2009.

[22] J. Shotton, J. Winn, C. Rother, and A. Criminisi, "TextonBoost for image understanding: Multi-class object recognition and segmentation by jointly modeling texture, layout, and context," *Int. J. Comput. Vis.*, vol. 81, no. 1, pp. 2–23, Jan. 2009.

[23] Y. Boykov and G. Funka-Lea, "Graph cuts and efficient N-D image segmentation," *Int. J. Comput. Vis.*, vol. 70, no. 2, pp. 109–131, Nov. 2006.

[24] L. Wang, G. Hua, R. Sukthankar, J. Xue, Z. Niu, and N. Zheng, "Video object discovery and co-segmentation with extremely weak supervision," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 10, pp. 2074–2088, Oct. 2017.

[25] O. Veksler, "Efficient graph cut optimization for full CRFs with quantized edges," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 4, pp. 1005–1012, Apr. 2020.

[26] A. Golovinskiy and T. Funk, "Min-cut based segmentation of point clouds," in *Proc. IEEE 12th Int. Conf. Comput. Vis. Workshops (ICCV)*, Kyoto, Japan, Sep. 2009, pp. 39–46.

[27] D. Munoz, N. Vandapel, and M. Hebert, "Directional associative Markov network for 3-D point cloud classification," in *Proc. 4th Int. Symp. 3D Data Process., Vis. Transmiss.*, Atlanta, GA, USA, 2008, pp. 1–8.

[28] D. Munoz, J. A. Bagnell, and M. Hebert, "Stacked hierarchical labeling," in *Proc. Eur. Conf. Comput. Vis.*, 2010, pp. 57–70.

[29] S. Oesau, F. Lafarge, and P. Alliez, "Indoor scene reconstruction using feature sensitive primitive extraction and graph-cut," *ISPRS J. Photogramm. Remote Sens.*, vol. 90, pp. 68–82, Apr. 2014.

[30] A. Monszpart, N. Mellado, G. J. Brostow, and N. J. Mitra, "Rapter: Rebuilding man-made scenes with regular arrangements of planes," *ACM Trans. Graph.*, vol. 34, no. 4, p. 103, Aug. 2015.

[31] Y. Boykov and O. Veksler, "Graph cuts in vision and graphics: Theories and applications," in *Handbook of Mathematical Models in Computer Vision*. New York, NY, USA: Springer, 2006, pp. 79–96.

[32] 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.

[33] Y. Manolopoulos and Y. Theodoridis, *R-Trees: Theory and Applications*. New York, NY, USA: Springer, 2005.

[34] M. Cramer, "The DGPF-test on digital airborne camera evaluation–overview and test design," *Photogrammetrie Fernerkundung Geoinformation*, vol. 2010, no. 2, pp. 73–82, May 2010.

[35] 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," *Remote Sens.*, vol. 6, no. 12, pp. 12885–12908, Dec. 2014.

[36] A. Torralba, K. P. Murphy, and W. T. Freeman, "Sharing visual features for multiclass and multiview object detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 5, pp. 854–869, May 2007.

[37] B. Guo, X. Huang, Q. Li, F. Zhang, J. Zhu, and C. Wang, "A stochastic geometry method for pylon reconstruction from airborne LiDAR data," *Remote Sens.*, vol. 8, no. 3, p. 243, Mar. 2016, doi: 10.3390/rs8030243.

[38] B. Guo, Q. Li, X. Huang, and C. Wang, "An improved method for power-line reconstruction from point cloud data," *Remote Sens.*, vol. 8, no. 1, p. 36, Jan. 2016, doi: 10.3390/rs8010036.

[39] D. Munoz, N. Vandapel, and M. Hebert, "Onboard contextual classification of 3-D point clouds with learned high-order Markov random fields," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2009, pp. 4273–4280.

[40] Y. Zheng, J. Fan, J. Zhang, and X. Gao, "Hierarchical learning of multi-task sparse metrics for large-scale image classification," *Pattern Recognit.*, vol. 67, pp. 97–109, Jul. 2017.

[41] C. Kurtz, A. Stumpf, J.-P. Malet, P. Gançarski, A. Puissant, and N. Passat, "Hierarchical extraction of landslides from multiresolution remotely sensed optical images," *ISPRS J. Photogramm. Remote Sens.*, vol. 87, pp. 122–136, Jan. 2014.

[42] Z. Wang, L. Zhang, T. Fang, P. T. Mathiopoulos, X. Tong, H. Qu, Z. Xiao, F. Li, and D. Chen, "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.

BO GUO received the Ph.D. degree from Wuhan University, Wuhan, China, in 2014.

He is currently an Associate Professor with the School of Civil and Transportation Engineering, Guangdong University of Technology, Guangzhou, China.

His research interests include point cloud data processing and application of laser scanning on photogrammetry and computer vision.

XIAOHAN ZUO received the bachelor's degree in surveying and mapping engineering from Anhui Agricultural University, Hefei, China, in 2018. She is currently pursuing the master's degree in technology of surveying and mapping with the School of Civil and Transportation Engineering, Guangdong University of Technology, Guangzhou, China.

Her research interests include deep learning and 3D point cloud classification.