SEMANTIC LABELING AND REFINEMENT OF LIDAR POINT CLOUDS USING DEEP NEURAL NETWORK IN URBAN AREAS

R. Huang[1], Z. Ye[1], D. Hong[2], Y. Xu[1,*], U. Stilla[1]

[1] Photogrammetry and Remote Sensing, Technical University of Munich (TUM), Germany
- (rong.huang, z.ye, yusheng.xu, stilla)@tum.de
[2] Remote Sensing Technology Institute, German Aerospace Center, Weisseling, Germany - danfeng.hong@dlr.de

ICWG II/III: Pattern Analysis in Remote Sensing

KEY WORDS: Point clouds, MLS, semantic labeling, deep learning, optimization

ABSTRACT:

In this paper, we propose a framework for obtaining semantic labels of LiDAR point clouds and refining the classification results by combining a deep neural network with a graph-structured smoothing technique. In general, the goal of the semantic scene analysis is to assign a semantic label to each point in the point cloud. Although various related researches have been reported, due to the complexity of urban areas, the semantic labeling of point clouds in urban areas is still a challenging task. In this paper, we address the issues of how to effectively extract features from each point and its local surrounding and how to refine the initial soft labels by considering contextual information in the spatial domain. Specifically, we improve the effectiveness of classification of point cloud in two aspects. Firstly, instead of utilizing handcrafted features as input for classification and refinement, the local context of a point is embedded into deep dimensional space and classified via a deep neural network (PointNet++), and simultaneously soft labels are obtained as initial results for next refinement. Secondly, the initial label probability set is improved by taking the context both in the spatial domain into consideration by constructing a graph structure, and the final labels are optimized by a graph cuts algorithm. To evaluate the performance of our proposed framework, experiments are conducted on a mobile laser scanning (MLS) point cloud dataset. We demonstrate that our approach can achieve higher accuracy in comparison to several commonly-used state-of-the-art baselines. The overall accuracy of our proposed method on TUM dataset can reach 85.38% for labeling eight semantic classes.

1. INTRODUCTION

In behalf of the popularity of LiDAR Detection and Ranging (LiDAR) techniques, 3D spatial information can be easily acquired, which is represented as 3D point clouds. However, a set of unstructured points can not provide semantic information describing objects in the real world directly and explicitly. To be specific, there is a semantic gap between the real applications of point cloud and the original representation of this 3D raw data. Hence, the semantic representation of point clouds serves as basis for many applications, such as 3D scene modeling (Moussa, El-Sheimy, 2010, Lafarge, Mallet, 2012), object detection (Jochem et al., 2009, Xu et al., 2018) and automatic navigation (Biswas, Veloso, 2012).

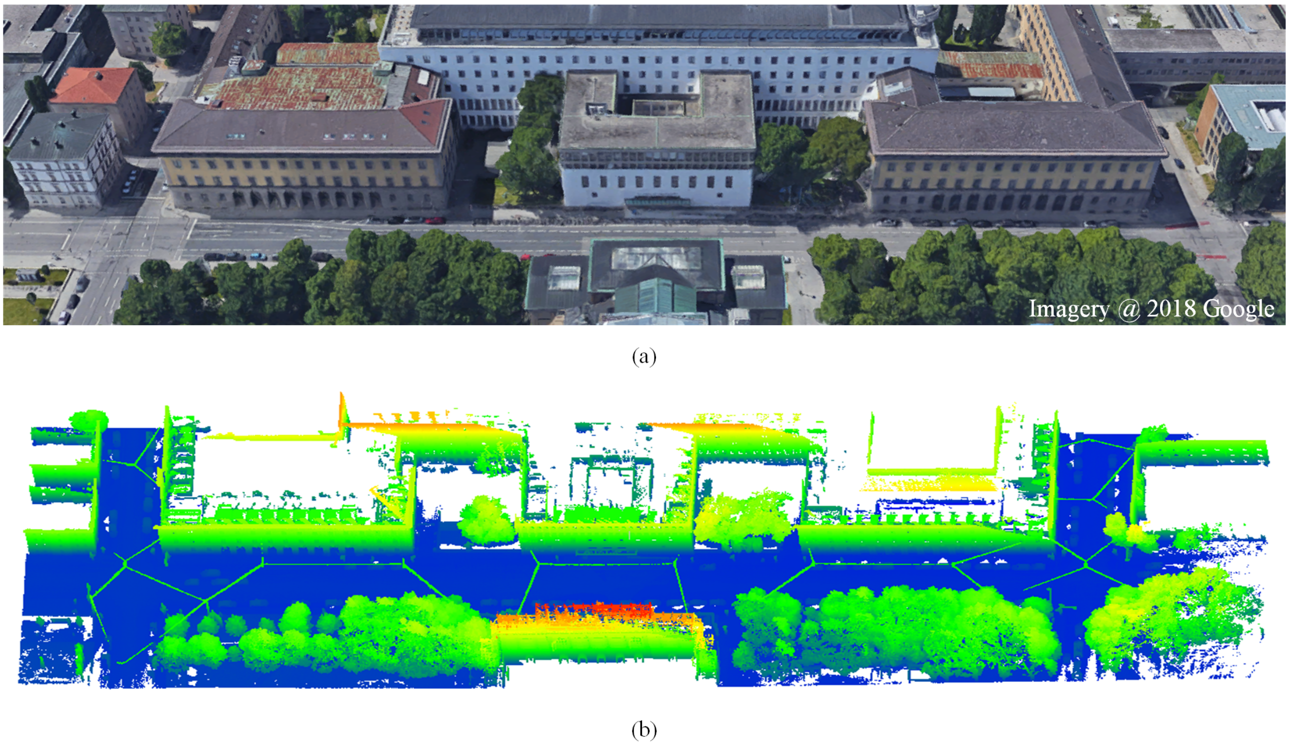
Similar to the other remotely sensed data, due to the complexity of urban environments (Hong et al., 2018), the quality of acquired point clouds could be influenced from many aspects, such as noises and outliers resulting from errors of scanning, uneven point cloud density caused by varying measuring distances of scanners, occlusions caused by the strained observation positions, and disturbances caused by dynamic objects. Owing to these problems, the semantic scene analysis of point clouds is still a challenging task in urban areas.

In general, the goal of the semantic scene analysis is to assign a semantic label to each point in the point cloud (Hackel et al., 2016). Conventionally, the semantic labeling compromises extracting a variety of handcrafted features for each point in a point cloud, and concatenating them to a feature vector, and then feeding this feature vector into a classifier trained on representative training samples (Weinmann et al., 2015a). The classifiers, such as AdaBoost (Lodha et al., 2007), support vector machines (SVM), and random forest (Chehata et al., 2009, Sun et al., 2018) are widely used for this semantic labeling tasks. Owing to the ease of easy conduction and good performance, these supervised statistical methods are the most common way applied for this task. However, for these supervised point-wise classification, although good classification results could be produced with this straightforward procedure due to the high distinctness of hand-drafted features (Hong et al., 2015), the classification result may be in-homogeneous, especially in the areas with low point density and the boundaries of objects, due to the deficiency of the consideration of the local neighborhood of each point. To enhance the regional smoothness of the result of semantic labeling, some contextual classification methods have been proposed, such as Markov random fields (Munoz et al., 2009, Lu, Rasmussen, 2012) and conditional random fields (Niemeyer et al., 2014, Weinmann et al., 2015b, Yao et al., 2017). In this method, each point is classified considering not only the extracted features but also the features and the labels of its surrounding points. With the improvement of spatial smoothness, the classification results are indubitably refined, which is also accompanied by the high expense of computation.

With the availability of high-performance computation resources and the access to large-scale datasets, deep learning techniques have been blooming in recent years and appear to be compelling tools in many fields, such as image classification, segmentation, and object detection and tracking. Concurrently, many pieces of research have paid their attention to applying this technique to 3D point cloud classification. For the sake

[*] Corresponding author

This contribution has been peer-reviewed. The double-blind peer-review was conducted on the basis of the full paper.
https://doi.org/10.5194/isprs-annals-IV-2-W7-63-2019 | © Authors 2019. CC BY 4.0 License.



(a)

(b)

Figure 1. Illustration of the test area in experiments. (a) Image scene of Arcistrasse facing the TUM main entrance from Google maps, 2018. (b) MLS point cloud colored regarding to height.

of the limitation of the way of data representation of an exemplary convolutional neural network, initial deep learning based 3D classification techniques are usually conducted after a 3D-to-2D transformation, in which some spatial information gets lost. With the emergence of PointNet (Qi et al., 2017a) and its derivatives (Qi et al., 2017b, Qi et al., 2018), new possibilities arise for the classification of point clouds. For deep learning techniques derived from PointNet, 3D point sets can be used as input for the network directly, and the feature mentioned above extraction and supervised classification steps are incorporated to achieve an end-to-end classification strategy, which dramatically simplifies the procedure of semantic labeling. Simultaneously, in the streamline of PointNet, the local and global features are both learned, which improves the ability to consider the local context for each point. However, for the deep learning techniques like PointNet, to some extent, the results of classification rely on the methods in sampling and splitting on preprocessing, and the way of interpolation on postprocessing since the input number of samples should be fixed when fed into a network. Some classification errors and ineffectiveness in the boundary for each split point set will be induced in these steps.

To obtain smoother classification results, one commonly used approach is to add a post-processing step considering the contextual information for each point. Compared with the aforementioned contextual classification methods, this strategy is less computation-expensive and less-dependent on the interactive feature extraction step. In the process of smoothing of semantic labeling, neighborhood dependencies are considered during the processing by exploring the interaction of neighborhood labels. Based on the scale of neighborhood dependencies, the methods of spatial regularization can be classified into two categories: local neighborhood and global optimizations. Local labeling refinement techniques usually focus on the assignments of weights for each neighboring point. Based on weights calculated based on different strategies, local filters can be applied to improve the local smoothness and thus improve the spatial smoothness on the whole classification results (Lillesand et al., 2014, Bilgin et al., 2008, Kang et al., 2014). These approaches work well when only small regions are wrongly labeled in the initial classification results, and the effectiveness of the smoothing methods is profoundly affected by the definition of local neighborhood. Another strategy is global optimization, in which the initial label probabilities and the spatial correlation in local neighborhood are simultaneously considered by finding an optimal solution for a cost function (Landrieu et al., 2017, Li et al., 2016). This result relies on the way of constructing the graph and the selection of adjacency neighbors. The problem for this method is also the ineffectiveness in large wrongly classified areas.

In this work, we aim at refining the initial classification results of PointNet++ by applying a graph-based optimization in the post-processing step. To be specific, the core concept of our method is to merge the advantages of the capabilities of deep learning in providing in-depth features and the high quality initial soft labels, and the benefits of graph cuts in finding the relation of each sample and its surroundings and optimizing the initial labels. Thus, we can achieve the useful refinement of semantic labeling in complex urban areas.

The contributions of this work are presented as follows:

- A general framework for semantic labeling of MLS point clouds and refinement of classification results is proposed.
- Instead of utilizing handcrafted features as input for classification and refinement, the local context of a point is embedded into deep dimensional space via an auto-encoder (PointNet++), and simultaneously soft labels are obtained as initial results for next refinement.

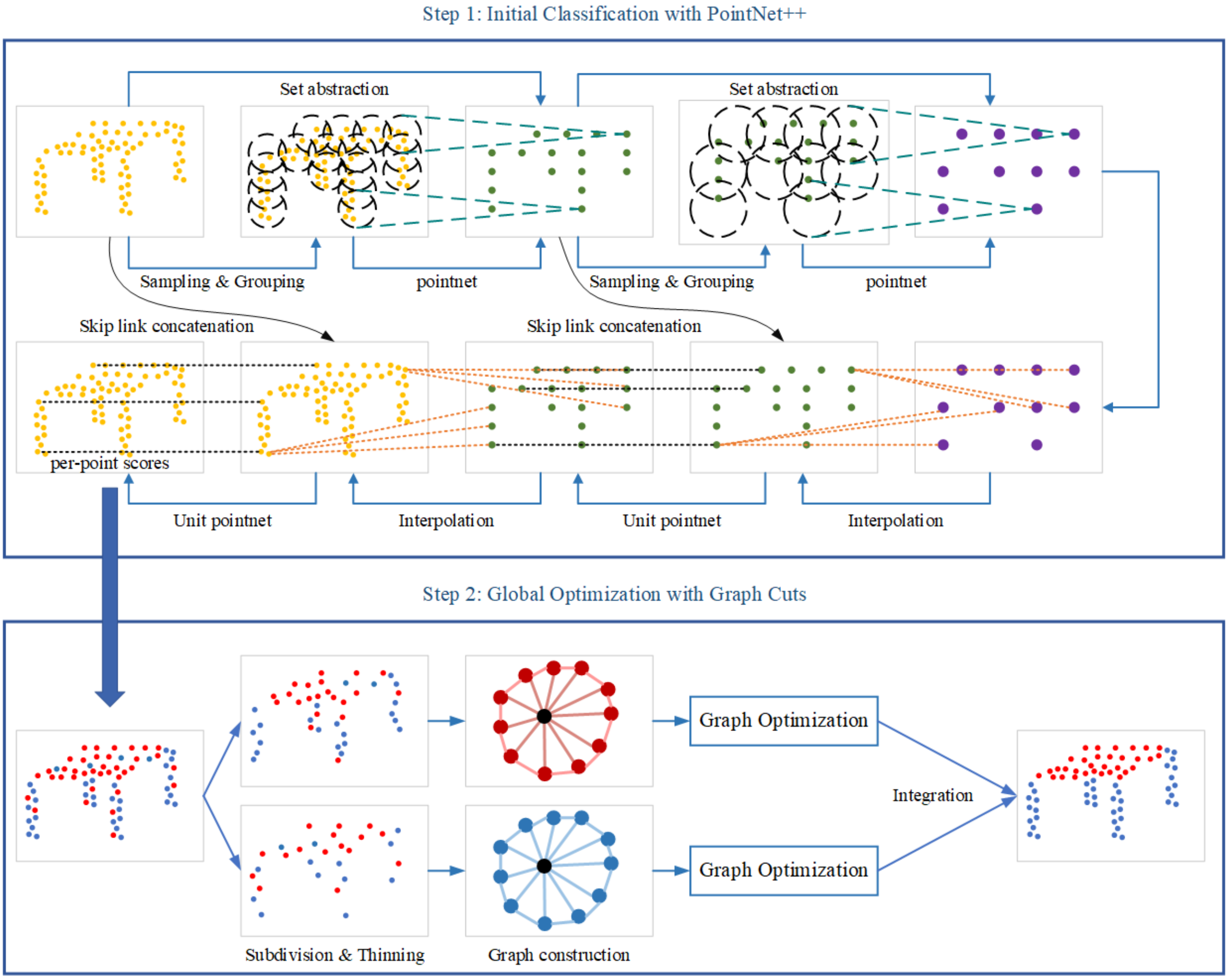


Figure 2. Overview of the proposed classification strategy.

- An optimization method based on regularization on a graph structure is conducted on initial labeling result of 3D point cloud to achieve spatial smoothness of semantic labeling.

2. METHODOLOGY

The complete methodology that we apply for point cloud classification is composed of two major steps illustrated in Figure 2: pre-classification with PointNet++ and classification refinement with global spatial regularization. In the first step, the initial classification results with soft labels are obtained by feeding subdivided point clouds into PointNet++. Subsequently, the initial labels are refined through constructing a weighted graph for global regularization which considers both the spatial correlation and the initial labels of the points in the neighborhood.

2.1 Introduction to PointNet++

Since there is no rigid structure for raw point clouds, it is impossible to throw the original point clouds directly into a convolutional neural network like the processing way for images, when we try to apply deep learning techniques to point clouds. PointNet, as a pioneer work on processing point set directly, provides a solution for spatial encoding ignoring the permutation of points and decentralizing the point sets, by learning the global features and integrating them with individual point features. Unlike other deep learning based techniques, PointNet deals with point sets without loss of spatial information and allows for simplified preprocessing step without rasterization of point sets.

In the base of PointNet, PointNet++ was induced to be a hierarchical strategy to improve the ability to capture local neighborhood and the effectiveness in complex urban areas. The general idea of PointNet++ is to add sampling and grouping layers to learn hierarchical point set features. The original point sets are firstly subdivided the point sets into several overlapping sub-pointsets, and local features are learned from local partitions to provide low-level local features. The local features are then grouped to learn high-level features. This procedure is repeated until the global features of the whole point sets are learned. In each step of hierarchical feature learning, the pointnet layer follows the sampling and grouping layer due to its superiority in feature learning from point sets. Due to the sampling step in hierarchical feature learning step, when dealing with the segmentation task, the features for the original points are propagated from subsampled points with interpolation and skip links with layers in set abstraction step. Compared with PointNet, PointNet++ is not only able to recognize fine-grained patterns and work better under the high complex scenes, but also more robust to varying point densities. Thus,

https://doi.org/10.5194/isprs-annals-IV-2-W7-63-2019 | © Authors 2019. CC BY 4.0 License.

PointNet++ is chosen to produce initial classification results in our proposed framework.

2.2 Initial classification of point clouds with PointNet++

In our work, the segmentation part of PointNet++ is applied to obtain initial point-based classification results. Simultaneously, due to the distinction between the urban scenes with the object-based input in ShapeNet dataset (Yi et al., 2016), the urban scene is subdivided into sub-point sets as input for the network using the strategy of voxelization for the whole scene. The details will be detailed in the preprocessing step in the section of experiments. In this step, soft labels are produced for subsequent regularization in the form of classification probabilities for each class.

2.3 Refining soft labels by regularization

Considering the classification probabilities as input in this step, the refinement of initial labels is conducted by searching for the optimal labels with the improvement of spatial regularity. This step can be divided into three sub-steps, namely subdivision and thinning of the original pre-classified point cloud, construction of a weighted graph and global optimization with Graph Cuts.

2.3.1 Subdivision and thinning Due to the high density of MLS point cloud and the large data amount of complex urban scenes, it is almost impossible to conduct graph-based regularization on the condition that more than ten millions of points with labels are fed. Many research proposed voxel- or supervoxel-based regularization method to downsample the points and decrease the number of points fed into regularization methods. Inspired by this downsampling strategy and to keep the spatial resolution, we proposed a thinning-based way which subdivides the pre-classified points into several sub-point sets. Owing to the manner of random-sampling for each subset, the geometric context for each point is not significantly changed, and the main structure is maintained. Thus the graph-based regularization can be further conducted on each subset. After the optimization step, these subsets will be merged to produce a classified point cloud with the same points as the original point cloud and the optimized labels.

2.3.2 Graph construction Graph cuts is a powerful and popular algorithm which is widely used in the field of computer vision for energy optimization, which is an optimization algorithm based on energy function minimization.

In general, a graphical model is composed of vertices and edges. Specifically, a graph $\mathcal{G} = (V, E)$ is used to represent the data (e.g., image) to be classified, and V and E are respectively a set of vertex and edge. If the edges have directions, such graphs are called directed graphs. Otherwise, they are undirected graphs. Each edge is entitled to values, and the value of edges varies according to different weights which are related to different physical attributes. In the Graph Cuts algorithm, the graphical model is slightly different from the normal graphical model. The Graph Cuts graph has two more vertices based on the normal graph. These two vertices s and t are represented by the symbols “S” and “T” (see Figure 3b), which are collectively referred to as terminal vertices. All other vertices should be joined and connected to the two vertices to build part of the edge set. To be specific, there are two kinds of vertices and two types of edges defined in Graph Cuts. Elements of the data structure build the first type of vertices and edges. The vertex corresponds to each pixel in the image, and in the case of point cloud classification, the vertex corresponds to each point in the point cloud, while the connection of every two adjacent vertices is an edge, the set of which is also called n-links. Alternatively, the second type of vertices and edges is built between two terminal vertices, called the source point S meaning the source of the flow and the sink point T meaning the convergence of the flow, and vertices of the first type. In other words, we need to construct a connection between common vertices of the data structure and the two terminal vertices to form a second side, called t-links.

As illustrated in Figures 3a-b, an s-t diagram corresponding to the data structure of an image is provided. Each pixel corresponds to a corresponding vertex in the graph, and two additional vertices are representing s and t. The edge of the solid black line represents the edge n-links of the normal vertex connection of each two neighborhoods, and the edge t-links of each normal vertex connected with s and t are represented by the edge of the dotted line represents. In the context of segmentation, s generally represents the foreground target, while t generally represents the background. A “cut” for the graphical model is a subset P of the edge set E, and the cost $|P|$ of the “cut” is the sum of the weights of all edges of the subset P. The disconnection of edges in the set will result in the separation of the graphical model, corresponding to two disjoint subsets “S” and “T”, which is term as “cutting” (see Figure 3c). If a “cut” has the smallest sum of the weights for all of its edges, it is regarded as the minimum cut. According to the Ford-Fulkerson theorem, the minimum cut problem of the graphical model is equal to the maximum flow problem of the network, so it could be solved by the optimized solution using Goldberg-Tarjan algorithms (Boykov, Kolmogorov, 2004). Namely, we set the label of vertices of “S” to one, and the label of vertices of “T” to zero. Then, the “cut” of the graph can be achieved by minimizing the energy function:

$$\mathcal{E}(L) = \mu \cdot \mathcal{R}(L) + \mathcal{B}(L) \tag{1}$$

where $L = \{l_1, l_2, ..., l_u\}$ is the set of labels (0 or 1) given to all the vertices. Here, μ is the important factor determining the balance of influence between $\mathcal{R}(L)$ and $\mathcal{B}(L)$ on the energy. $\mathcal{R}(L) = \sum \mathcal{R}_u(l_u)$ is the regional term relating to the weight, and $\mathcal{R}_u(l_u)$ represents the penalty for assigning the label l_u to the vertex v. While $\mathcal{B}(L) = \sum \mathcal{B}_{<u,v>} \cdot \delta(l_u, l_v)$ is the boundary term. $\mathcal{B}_{<u,v>}$ can be resolved into a discontinuous penalty between vertices u and v according to following:

$$\mathcal{B}_{<u,v>} = e^{-\frac{(I_u - I_v)^2}{2\delta^2}} \tag{2}$$

where I_u and I_v stand for the general values of vertices u and v. The term $\delta(l_u, l_v)$ is the key to the boundary term, and on this term there are three constraints:

$$\delta(l_\alpha, l_\beta) = \begin{cases} 0 & \text{if } \; l_\alpha = l_\beta \\ 1 & \text{else} \end{cases} \tag{3}$$

$$\delta(l_\alpha, l_\beta) = \delta(l_\beta, l_\alpha)\delta(l_\alpha, l_\beta) > 0 \tag{4}$$

$$\delta(l_\alpha, l_\beta) \leq \delta(l_\alpha, l_\rho) + \delta(l_\rho, l_\beta) \tag{5}$$

Here, l_ρ is the label of a third vertex. The first two constraints tell that an energy between two different labels l_α and l_β should be non-zero. If it is zero, that means the two labels are the same. Generally, if u and v have similar properties, then $\mathcal{B}_{<u,v>}$ will be larger, and if they are totally different, then $\mathcal{B}_{<u,v>}$ is close

https://doi.org/10.5194/isprs-annals-IV-2-W7-63-2019 | © Authors 2019. CC BY 4.0 License.

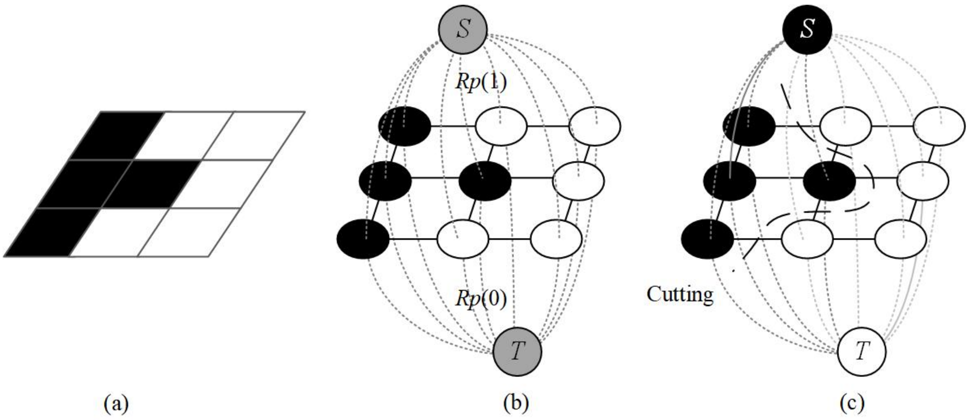


Figure 3. Graph Cuts for optimization. (a) Original data structure. (b) Constructed graph. (c) Cut of the graph.

to zero. While the last constraint defines the triangle rule, limiting that a shortcut of edges has always lower or similar energy than taking the whole path of edges. Only if the last constraint is satisfied, we can say the boundary term $\mathcal{B}_{<u,v>}$ is metric.

2.3.3 Global optimization of graph structures The solution of the energy function as mentioned above can be achieved by two algorithms: alpha-expansion and alpha-beta-swap (Delong et al., 2012). Here, the alpha-expansion algorithm can only be applied when the boundary term is metric. Otherwise, the alpha-beta swap algorithm will be used. The general idea of the alpha-expansion algorithm is to separate all l_α labeled and non-l_α labeled nodes with "cutting" and the algorithm will change the label of l_α at each iteration. At each iteration, the region $\mathcal{R}_\alpha$ near the node with label l_α is expanded, with the graph weights reset. During the iteration, if two neighboring nodes do not share the same label, an intermediate node is inserted with weighted linking to the distance to the node with label l_α. The algorithm will iterate through each possible label for l_α until it converges. In contrast, the alpha-beta swap algorithm is to successively partition all nodes with label l_α from nodes with label l_β with "cutting" and the algorithm will change the label combination $l_\alpha - l_\beta$ at each iteration. During each iteration, the graph is constructed in a reasonable way which can segment between the region $\mathcal{R}_\alpha$ and the region $\mathcal{R}_\beta$ efficiently. In other words, for a node, the terminal link weight should be added with the sum of all links to neighbors which are neither within the region $\mathcal{R}_\alpha$ nor in the region $\mathcal{R}_\beta$. Similarly, the algorithm will iterate through each possible combination $l_\alpha - l_\beta$ until it converges.

3. EXPERIMENTS

3.1 Experimental dataset

This dataset is originally acquired by Fraunhofer Institute of Optronics, System Technologies and Image Exploitation (IOSB) (Gehrung et al., 2017). The used point clouds are acquired by two Velodyne HDL-64E mounted at an angle of 35° on the front roof of the vehicle. The original raw point clouds are also preprocessed by a statistical outlier removal for downsampling and noise suppressing. The number of points after preprocessing is around 50 million.

With thousands of scans acquired by the laser scanners along the Arcisstrasse, a scene containing various kinds of objects are obtained through the combination of the point cloud of all the scans. For the evaluation process, we also generate an accurate manually labeled point cloud for the experimental dataset as ground truth.

3.2 Preprocessing

In order to fulfill the requirement for input in PointNet++, the whole point cloud is subdivided into thousands of sub-point chips, in which 65536 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++. Each point in the chip is represented by a 3D-vector, containing the coordinates (x, y, z).

The total number of points for training is 19,974,050, and the training dataset is subdivided into ten folds. Nine folds are used for training, and one fold is used for validation when training PointNet++ for semantic segmentation. As for the test part, the number of points is 21,205,244.

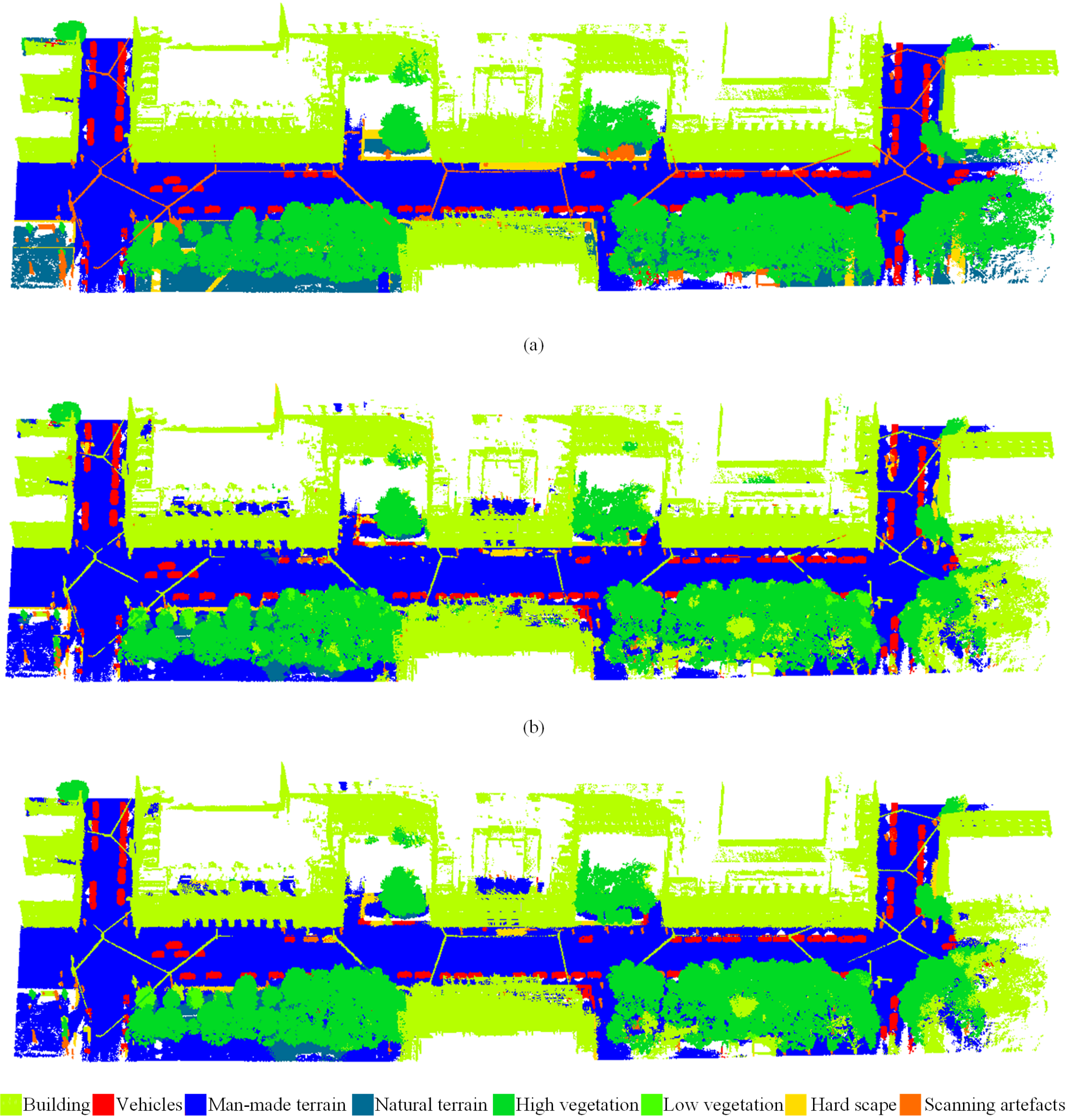
3.3 Traning parameters

By the way, 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 employed for training. To adjust the learning rate, we decayed its value by the factor of 0.7 in every 40 training chips. The training process proceeds for a total of 200 epochs. We monitor the progress of the validation loss and save the weights if the loss improves. All experiments were implemented with Tensorflow and carried out by NVIDIA TITAN X (Pascal) 12GB GPU.

3.4 Experimental results

In general, with our proposed method, the overall accuracy for eight semantic labels can reach 85.38%. The detailed statistical results are shown in Table 1.

To evaluate the performance of our proposed method against others, we compare our method with the results of traditional point-based classification with handcrafted features (HF) and random forest (Sun et al., 2018) as the classifier (single scale features are used, and the neighborhood size is set as 100 (formed by KNN) regarding the point density, and the number of trees in Random Forest is set as 50, detemined by cross validation), and the original results of PointNet++. As illustrated in Table 1, our proposed method outperforms the other methods concerning overall accuracy and accuracy for most categories.



(a)

(b)

(c)

Figure 4. Illustration of the classification results on the TUM dataset. (a) benchmark with manual labels. (b) the classification results with PointNet++. (c) the optimized classification results with our proposed method.

Table 1. Classification accuracy for different classification methods using TUM dataset.

Class	HF + Random Forest	PointNet2	PointNet2 + Global Regularization
Man-made terrain	0.8196	0.7200	0.7029
Natural terrain	0.3516	0.7488	0.7947
High vegetation	0.1760	0.9684	0.9717
Low vegetation	0.1442	0.0392	0.0566
Buildings	0.4246	0.8818	0.8937
Hard scape	0.0852	0.2335	0.2802
Scanning artefacts	0.1029	0.3067	0.3757
Vehicles	0.4397	0.8522	0.8899
OA	0.4721	0.8354	0.8538
AA	0.3180	0.5938	0.6207
Kappa	0.2226	0.7725	0.7964

The baseline method of point-wise classification is not admirable in some categories because of the unbalance in training samples for each class. Compared with the original results of PointNet++, our proposed method works better, especially for hard scape, scanning artefacts and vehicles. When the whole point cloud is subdivided into chips as input for PointNet++, there is a high possibility that these small objects (compared to large objects like building in urban scenes) are broken into pieces with the loss the main structure. With global optimization, these pieces can be grouped to share the same labels and appear to have a complete structure, which is illustrated in the classification accuracy. However, as illustrated in Figure 4, for large wrongly classified areas, such as the right-bottom corner, the global optimization can do nothing about it.

4. CONCLUSION

In this work, we propose a framework for point cloud classification. The experiments are conducted using TUM lidar dataset, which proves the effectiveness and efficiency of our method. Specifically, the proposed method is a combination of the advantages of the capabilities of deep learning in providing deep features and the high quality initial soft labels, and the advantages of global optimization in finding the relation of each sample and its surroundings, and the performance has been proved by quantitative and qualitative comparison with other classic classification methods.

ACKNOWLEDGEMENTS

This work is supported by the China Scholarship Council. The authors would like to thank Dr. Markus Hebel, Joachim Gehrung, and Zhenghao Sun for their valuable help and support to this work. This work was carried out within the frame of Leonhard Obermeyer Center (LOC) at Technische Universität München (TUM) [www.loc.tum.de].

REFERENCES

Bilgin, G., Erturk, S., Yildirim, T., 2008. Unsupervised classification of hyperspectral-image data using fuzzy approaches that spatially exploit membership relations. *IEEE Geoscience and Remote Sensing Letters*, 5(4), 673–677.

Biswas, J., Veloso, M., 2012. Depth camera based indoor mobile robot localization and navigation. *2012 IEEE International Conference on Robotics and Automation*, IEEE, 1697–1702.

Boykov, Y., Kolmogorov, V., 2004. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 1124–1137.

Chehata, N., Guo, L., Mallet, C., 2009. Airborne lidar feature selection for urban classification using random forests. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 38(Part 3), W8.

Delong, A., Osokin, A., Isack, H.N., Boykov, Y., 2012. Fast approximate energy minimization with label costs. *International Journal of Computer Vision*, 96(1), 1–27.

Gehrung, J., Hebel, M., Arens, M., Stilla, U., 2017. An approach to extract moving objects from MLS data using a volumetric background representation. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 4, 107.

Hackel, T., Wegner, J.D., Schindler, K., 2016. Fast semantic segmentation of 3D point clouds with strongly varying density. *ISPRS Annals of Photogrammetry, Remote Sensing & Spatial Information Sciences*, 3(3).

Hong, D., Liu, W., Su, J., Pan, Z., Wang, G., 2015. A novel hierarchical approach for multispectral palmprint recognition. *Neurocomputing*, 151, 511–521.

Hong, D., Yokoya, N., Chanussot, J., Zhu, X.X., 2018. An augmented linear mixing model to address spectral variability for hyperspectral unmixing. *IEEE Transactions on Image Processing*, 28(4), 1923–1938.

Jochem, A., Höfle, B., Hollaus, M., Rutzinger, M., 2009. Object detection in airborne LIDAR data for improved solar radiation modeling in urban areas. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences. Paris*, 38(part 3), W8.

Kang, X., Li, S., Benediktsson, J.A., 2014. Spectral–spatial hyperspectral image classification with edge-preserving filtering. *IEEE Transactions on Geoscience and Remote Sensing*, 52(5), 2666–2677.

Lafarge, F., Mallet, C., 2012. Creating large-scale city models from 3D-point clouds: a robust approach with hybrid representation. *International Journal of Computer Vision*, 99(1), 69–85.

Landrieu, L., Raguet, H., Vallet, B., Mallet, C., Weinmann, M., 2017. A structured regularization framework for spatially smoothing semantic labelings of 3D point clouds. *ISPRS Journal of Photogrammetry and Remote Sensing*, 132, 102–118.

Li, Z., Zhang, L., Tong, X., Du, B., Wang, Y., Zhang, L., Zhang, Z., Liu, H., Mei, J., Xing, X. et al., 2016. A three-step approach for TLS point cloud classification. *IEEE Transactions on Geoscience and Remote Sensing*, 54(9), 5412–5424.

Lillesand, T., Kiefer, R.W., Chipman, J., 2014. *Remote sensing and image interpretation*. John Wiley & Sons.

Lodha, S.K., Fitzpatrick, D.M., Helmbold, D.P., 2007. Aerial lidar data classification using adaboost. *Sixth International Conference on 3-D Digital Imaging and Modeling (3DIM 2007)*, IEEE, 435–442.

Lu, Y., Rasmussen, C., 2012. Simplified markov random fields for efficient semantic labeling of 3d point clouds. *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE, 2690–2697.

Moussa, A.M., El-Sheimy, N., 2010. Automatic classification and 3d modeling of lidar data. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences, France*.

Munoz, D., Bagnell, J.A., Vandapel, N., Hebert, M., 2009. Contextual classification with functional max-margin markov networks. *2009 IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, 975–982.

Niemeyer, J., Rottensteiner, F., Soergel, U., 2014. Contextual classification of lidar data and building object detection in urban areas. *ISPRS Journal of Photogrammetry and Remote Sensing*, 87, 152–165.

Qi, C.R., Liu, W., Wu, C., Su, H., Guibas, L. J., 2018. Frustum pointnets for 3d object detection from rgb-d data. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 918–927.

Qi, C.R., Su, H., Mo, K., Guibas, L. J., 2017a. Pointnet: Deep learning on point sets for 3d classification and segmentation. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 652–660.

Qi, C.R., Yi, L., Su, H., Guibas, L. J., 2017b. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in Neural Information Processing Systems*, 5099–5108.

Sun, Z., Xu, Y., Hoegner, L., Stilla, U., 2018. Classification of MLS point cloud in urban scenes using detrended geometric features from supervoxel-based local contexts. *ISPRS Annals of Photogrammetry, Remote Sensing & Spatial Information Sciences*, 4(2).

Weinmann, M., Jutzi, B., Hinz, S., Mallet, C., 2015a. Semantic point cloud interpretation based on optimal neighborhoods, relevant features and efficient classifiers. *ISPRS Journal of Photogrammetry and Remote Sensing*, 105, 286–304.

Weinmann, M., Schmidt, A., Mallet, C., Hinz, S., Rottensteiner, F., Jutzi, B., 2015b. Contextual classification of point cloud data by exploiting individual 3D neigbourhoods. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences II-3 (2015), Nr. W4*, 2(W4), 271–278.

Xu, Y., Tuttas, S., Hoegner, L., Stilla, U., 2018. Reconstruction of scaffolds from a photogrammetric point cloud of construction sites using a novel 3D local feature descriptor. *Automation in Construction*, 85, 76–95.

Yao, W., Polewski, P., Krzystek, P., 2017. Semantic labeling of ultra dense MLS point clouds in urban road corridors based on fusing CRF with shape priors. *International Archives of the Photogrammetry, Remote Sensing & Spatial Information Sciences*, 42.

Yi, L., Kim, V.G., Ceylan, D., Shen, I.-C., Yan, M., Su, H., Lu, C., Huang, Q., Sheffer, A., Guibas, L., 2016. A Scalable Active Framework for Region Annotation in 3D Shape Collections. *SIGGRAPH Asia*.