A SUPER VOXEL-BASED RIEMANNIAN GRAPH FOR MULTI SCALE SEGMENTATION OF LIDAR POINT CLOUDS

Minglei Li[1],

[1] College of Electronic and Information Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing, China – minglei_li@126.com

Commission III, WG III/5

KEY WORDS: LiDAR Point Clouds, Multi Scale Segmentation, Geometry Processing, Riemannian Graph, Super Voxels.

ABSTRACT:

Automatically segmenting LiDAR points into respective independent partitions has become a topic of great importance in photogrammetry, remote sensing and computer vision. In this paper, we cast the problem of point cloud segmentation as a graph optimization problem by constructing a Riemannian graph. The scale space of the observed scene is explored by an octree-based over-segmentation with different depths. The over-segmentation produces many super voxels which restrict the structure of the scene and will be used as nodes of the graph. The Kruskal coordinates are used to compute edge weights that are proportional to the geodesic distance between nodes. Then we compute the edge-weight matrix in which the elements reflect the sectional curvatures associated with the geodesic paths between super voxel nodes on the scene surface. The final segmentation results are generated by clustering similar super voxels and cutting off the weak edges in the graph. The performance of this method was evaluated on LiDAR point clouds for both indoor and outdoor scenes. Additionally, extensive comparisons to state of the art techniques show that our algorithm outperforms on many metrics.

1. INTRODUCTION

Light detection and ranging (LiDAR) modules can obtain 3D point data of the scene with high efficiency and reliability. Exploiting such data could address a variety of tasks such as the urban modeling (Musialski et al., 2013; Rottensteiner et al., 2014; Li et al., 2016), object detection (Pu et al., 2011; Guo et al., 2014; Serna et al., 2014), robot navigation (Alismail et al., 2014) and different geographic information systems (Boyko and Funkhouser, 2011; Zhou and Vosselman, 2012). These tasks commonly require the segmentation of point clouds as a preprocess which may necessitate human intervention and can be quite time consuming.

The development of automatically segmentation pipeline is difficult by a number of factors. First of all, the elements of the scene usually exhibit in a variability of scales since the datasets are unstructured and massive. The scale problem make difficulties to bring control upon the segmentation sizes (Demantk et al., 2011). Secondly, the criteria for the definition of neighbor relation of point clouds are more complex compared with that in 2D image analysis, because the 3D point dataset has one more degree of freedom. Moreover, reliable features of points/regions are critical for the scene partition and clustering (Weinmann, et al., 2015; Vosselman et al., 2017), but under these conditions the feature description for local details are error-prone.

Many different approaches have been proposed in the literature for segmenting independent elements from the unstructured observation. However few efficient methods have been designed to solve all afore mentioned degenerate factors. Our goal is to investigate on the way to guide the unsupervised segmentation by the hierarchized quality criteria as geometric accuracy, structure complexity and semantics awareness. To explore the scale space of the structures, the proposed method investigates the observed structure as levels of detail from an adaptive octree algorithm. Another crucial procedure is to describe the local characteristics of point cloud with different complexities using a multi scale super voxel-based Riemannian graph, so the details of the segmentation are preserved among the different nodes.

The outline of this paper is as follows. In Section 2, we review the related literatures and briefly explain the proposed approach. Section 3 details our segmentation method. In Section 4, we provide illustrations and experiments to demonstrate the utility of our method. Finally, we offer some conclusions and suggests directions for future work.

2. RELATED WORK

2.1 A Classification of the Segmentation Methods

In the last two decades, there is a considerable body of work aimed at solving the problems of point cloud segmentation so as to further classify or modelling specific objects. Broadly speaking there are four categories in which the problem can be addressed.

The first one is related to the region growing-based methods. Many methods adopt the region growing strategy by modifying the seed selection or region extending criteria. For instance, Gorte (2002) investigated the TIN surface and used the angle and distance between the neighbouring triangles for the growing. Vieira and Shimada (2005) and Nurunnabi et al. (2012) removed points along sharp edges using a curvature threshold and the remaining points were considered as seed points. Vo et al. (2015) proposed using octree nodes to select the seed points followed by region growing to refine the coarse segments. The region growing-based methods are widely used because the methods are easy to implement and useful for extracting large dominated regions (Pu and Vosselman, 2006). However, when investigating

https://doi.org/10.5194/isprs-annals-IV-3-135-2018 | © Authors 2018. CC BY 4.0 License.

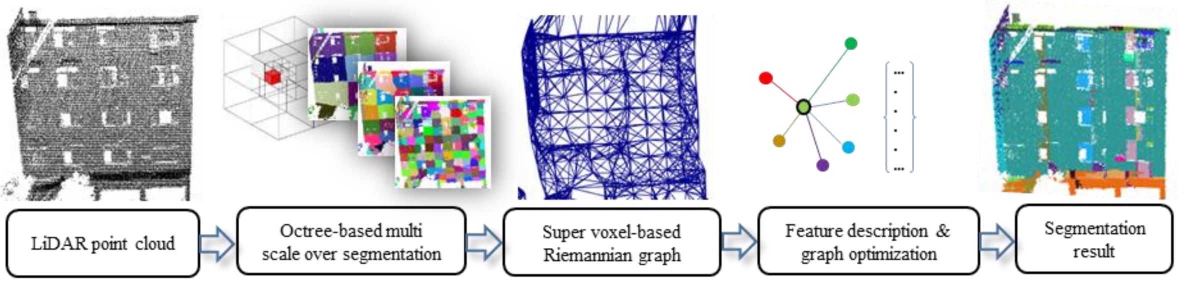

LiDAR point cloud
Octree-based multi scale over segmentation
Super voxel-based Riemannian graph
Feature description & graph optimization
Segmentation result

Figure 1. The schematic of the proposed approach

small regions in a large scene, the methods are sensitive to noise. Hence, they lack the analysis of the scale space. Moreover, the selection of seed points strongly affects the segmentation quality, while no universally valid criterion exists.

Secondly, the category is noticed as model fitting-based methods. These methods use parameterized models to fit the observed scenes. Two widely employed algorithms are the Hough Transform (HT) (Overby et al., 2004; Vosselman et al., 2004; Tarsha-Kurdi et al., 2007) and the random sample consensus (RANSAC) approach (Schnabel et al., 2007). Basically, within the dataset these methods iteratively detect potential geometry items that has some specific parameterized structures, such as cylinders, planes and spheres. The output represents the scene with many structured shapes which have compact geometry forms and are resistant to noise and outliers. However, these methods lack the ability to express complex undefined shapes. Besides, the requirements for the fitting thresholds hinder the model fitting-based methods to resolve scale problems. Another problem caused by the threshold is that the methods are sensitive to the density variety of the dataset.

The third category is graph-based segmentation. Graph-based segmentation has been successfully used in the field of image analysis. One of the most popular approach is graph cut method that is designed to minimize energy function in weighted graph where the energy function defines segmentation (Boykov and Jolly, 2001). For 3D point analysis, the main challenge to build graph is how to deal with differences in node and edge structure. For example, Golovinskiy and Funkhouser (2009) used *k*-nearest neighbors (KNN) to build the graph on the point cloud. Many works benefits from graph cuts for 3D data segmentation (Rusu et al., 2009; Ural and Shan, 2012; Guinard and Landrieu, 2017). However, previous methods build graph based on points, which lack stable features for various scenes. Point-based graph usually maintains massive volume that stunts the computation efficiency. In terms of graph cut mechanism, the methods tend to segment the target objects from background, hence they need to assume there are some presupposed targets such buildings or vegetation.

The last one is machine learning-based segmentation. It is also referred to as object retrieval which has received significant attention in recent years (Pang and Neumann, 2013; Yu et al., 2015; Boulch et al., 2017; Yang et al., 2017). The crucial part of these methods is how to extract salient geometric and topological characteristics from the object shapes. At the learning stage, a large set of features are extracted from every object and a training algorithm is applied to learn the classification function in the feature space. Yang and Dong (2013) used support vector machines (SVMs) to over segment the points, then these over-segmentations are merged based on topological connectivity into a meaningful geometrical abstraction. Yang et al. (2017) combined multiple aggregation levels of features and contextual features to recognize road facilities. Laga and Nakajima (2007) used AdaBoost to learn a classifier that relies on a small subset of the features with the mean of weak classifiers. The convolutional neural network (CNN) has been used for semantic labeling of point clouds by transforming the point dataset to a voxelization of the space (Wu et al., 2015) or many picked snapshots of the point cloud (Boulch et al., 2017). Although in point cloud data with noise, uneven density and occlusions machine learning techniques give better results for extracting specific objects, they are usually slow and rely on the result of feature extraction process. The definition of neighborhood strongly influences the estimated feature vectors.

2.2 Overview of the Proposed Approach

In spite of recent research effort, a satisfactory solution to the problem of point cloud segmentation has not been proposed yet. In this paper, we present a new, fully automated framework composed of three components: (i) investigating the scale space by an adaptive octree to generate multi scale super voxels; (ii) building voxel-based Riemannian graph; (iii) voxel clustering using a graph optimization method. An illustration of the schematic of the approach is shown in the Fig. 1. In contrast to previous methods, the proposed approach has a number of advantages.

Firstly, the scales of various shapes in the scene are studied by exploring through different depth of the octree. An over segmentation process produces multi scale super voxels based on octree nodes in different depth. These super voxels hold different sizes corresponding to the local details. In this way, the method avoids the typical segmentation errors caused by an inappropriate choice of seed points or growing thresholds.

Secondly, we use super voxels as the nodes to build the Riemannian graph which satisfies the manifold requirement. Our definition of the edge-weight is linked directly to the geodesic distance of the underlying manifold. We introduce the method to get geodesics approximation results from the general Euclidean distance functions. The feature description process uses super voxels to interpret high level characteristics of each local region and to gain more useful features. Since the high level features are insensitivity to noise and uneven density, the segmentation algorithm could adapt to the different laser scanners which may produce different point clouds even with the same scene.

Third, once the graph models are available, an energy optimization work is conducted to cluster all super voxels. Since the final step is processed on the super voxels which drastically reduced the node quantity compared with the point-based graph, the approach maintains high computation efficiency. In sum, the workflow of the proposed approach is a full automation, and the final segmentation results are compact and each partition is adaptive to the local details with the appropriate scales.

https://doi.org/10.5194/isprs-annals-IV-3-135-2018 | © Authors 2018. CC BY 4.0 License.

3. PROPOSED METHOD

3.1 Octree-based Multi Scale Over Segmentation

Our aim is to develop a framework to segment the scene into sectional partitions with different sizes corresponding to the local structures. In general, piecewise planar well represent the urban environments with buildings and other man-made objects. Hence, we assume that a valid voxel from the over segmentation is nearly flat.

The first phase of the over segmentation process is to extract some coarse cluster centres using an adaptive octree algorithm. Given a point cloud $\boldsymbol{P}:\{p_1, p_2,..., p_n\}$, the octree algorithm aims to choose a set of leaf nodes $\boldsymbol{N}:\{n_1^{d1}, n_2^{d2},..., n_m^{dm}\}$ as initial centres. d_m denotes the octree depth where the node exhibits. At the beginning, we define a depth sequence $\{d_{\min}, d_{\min}+1,..., d_{\max}\}$. The minimal depth is mandatory to subdivide the space to a certain level as the initial partition. Setting the maximum depth is because that deeper decimations may lead to drowning small objects in noise.

In practice, the minimum cubic bounding box enclosing $\boldsymbol{P}$ is used as the initial root node. Then the root node is recursively subdivided into eight child cubes with the same size until reaching the termination criterion. We use three criteria to determine the termination of the subdivision. First, if the point number in the cube is less than the minimum number threshold, the cube is a bottom leaf. Secondly, if the minimum eigenvalue of the covariance C of all points in the cube is smaller than 0.5, it is marked as a bottom leaf. C is also known as the 3D structure tensor matrix and it is defined as shown in Eq.1, where *k* is the number of points inside the cube and $\bar{p}$ is the mean position value of these points.

$$C = \frac{1}{k}\sum_{i=1}^{k}(p_i - \bar{p})\cdot(p_i - \bar{p})^T \tag{1}$$

Since C is a symmetric positive definite matrix, the principal component analysis (PCA) allows to retrieve the three principal directions, and the eigenvalues provide their magnitude. For the last criterion, if the depth of the cube reaches the depth threshold $d_{\max}$, the looping is stopped. The coordinate and the depth of the bottom leaf centre are recorded in a data structure n_j^{dj}.

After the space decomposition, a merging work is conducted by incrementally grouping adjacent points that have similar characteristics to the nearest cluster centres, based on a distance matric combined with the Euclidean distance D_S and the normal vector D_N. The point feature computed from the point data involves the normal vector $[n_x, n_y, n_z]^T$ and the spatial location $[x, y, z]^T$. To combine the two features into a single measure, it is necessary to normalize the normal measure and the spatial measure. For a given point p_i and a candidate centre seed *c*, the distance measure $\boldsymbol{D}$ is defined as:

$$\boldsymbol{D} = (2\text{-}D_N)\cdot D_S \tag{2}$$

where

$$D_N = n_{x_i} n_{x_c} + n_{y_i} n_{y_c} + n_{z_i} n_{z_c}$$

$$D_S = \sqrt[2]{(x_i - x_c)^2 + (y_i - y_c)^2 + (z_i - z_c)^2}$$

Since the normal vectors are normalized, the product of two normal ranges from -1 to 1, and the value of the term $(2\text{-}D_N)$ is between 1 and 3. If two points have the same normal value, their distance $\boldsymbol{D}$ equals the Euclidean distance D_S. On the contrary, if two points have opposite normal, their distance will be 3 times D_S. In this way, the proposed distance $\boldsymbol{D}$ has the ability to adjust the Euclidean distance, basing on the normal differences between the points.

Each point is associated with one nearest centre according to the distance $\boldsymbol{D}$, then the cluster centre is iteratively updated as the mean values $[x, y, z, n_x, n_y, n_z]^T$ of all the points belonging to its group. This procedure is a change of *k*-means algorithm. The spatial distance between the new cluster centre and the previous centre is the residual error that determines when to terminate the iteration. In most case, after three times of iteration, the cluster centres will be stable.

Since the octree nodes are selected by the covariance of the points in the cells, only few times iterations could converge to local compact voxels. An example of the octree-based multi scale over segmentation is shown in Fig. 2. Here, three typical regions are highlighted as the region of interest (ROIs) to emphasize the effects of the proposed approach. Seen in Fig.2 (b), the first ROI (1st row) belongs to a facade of a building with nearly uniform normal vectors; the second one (2nd row) is a vehicle whose normal vectors transform smoothly on its surface; the third ROI (3rd row) is one plant whose normal vectors have unregularly distribution. The balls in the second column depicted the neighbor sizes for calculating the normal vectors. All of the three ROIs have a similar point number (around 550 points) and a similar size. However, after the over segmentation, the first ROI is assigned into one voxel, the second one is split into 6 voxels and the last one is split into 11 voxels.

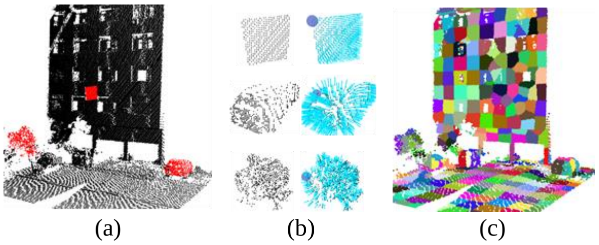


Figure 2. An example of the multi scale super voxels. (a) a point cloud with 3 ROIs, (b) the ROI points and their normal distributions, (c) the super voxel results.

3.2 Voxel-based Riemannian Graph

By now, the super voxels have been created with geometric and topological guarantees that adapt to the feature description need. In this section, we provide the theoretical basis for our graph embedding method. Our aim is to embed each super voxel as a graph node on a Riemannian manifold. We do this by viewing pairs of adjacent voxels as points connected by a geodesic on a manifold. The Riemannian graph connects points that are close together, while the distance and the edge weight are defined with a non-Euclidean distance function.

In practice, the construction of the Riemannian graph is done by adding edges between *k*-neighbouring vertices to the Euclidean Minimum Spanning Tree (EMST). The EMST favours points that are closest to each other in Euclidean space, but as explained in the literature (Hoppe et al., 1992), two parallel flattened shapes contain points who are close but they may have very different orientations. An example is shown in Fig.3, where P1 and P2 are two close points with obvious spatial difference. The Riemannian graph introduces the geodesic distance that is different with the Euclidean distance function, therefore we could precisely describe consistent features from local details with the geodesic distance. In Fig.3, the black curve connecting P1 and P2 is a geodesic distance.

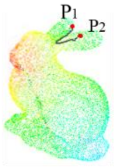


Figure 3. The geodesic distance between P1 and P2, who are close to each other but have obvious spatial difference.

Since the distance matrices in Euclidean space lack the ability to describe local traits of the scene, we design a new distance matric d_{Proj} that utilizes the voxel curvatures to explore the spatial proximity of adjacent voxels. In our graph model, each voxel is regarded as a compact surface patch, i.e. a node. As shown in Fig. 4, from voxel centre v_i to its neighbour v_j the distance is calculated by the projected point v'_j of v_j on the tangent plane over point v_i, and ρ here is the radius of the curvature. If the tangent plane fits well all points in the voxel, $\rho \to \infty$ and v'_j is close to v_j, so d_{Proj} will be close to the Euclidean distance; on the other hand, if the value of the curvature is big, d_{Proj} will be a larger value than the Euclidean distance to better distinguish between v_i and v_j.

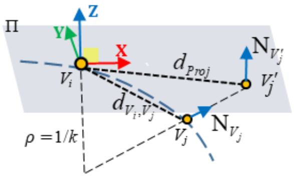


Figure 4. Distance metric considering local curvature

When building the MST, we use Kruskal coordinate (Robles-Kellya and Hancock, 2007) to realize the topological indexing. Given a voxel node v_i, we sort all of its neighbours as $\{v_{neighbors}\}$ depending the distance d_{Proj} from small to large. Then, loop from the edge with the minimum distance to the next edge until all edges are connected to a complete component. Next, we extract the Riemannian graph $G = (V, E)$ by adding edges between k-neighbouring nodes based on the geodesic from MST. The number k is set to 8 which is a suitable trade-off between tuning accuracy and computing time. An illustration of the MST result and the Riemannian graph representation of the scene are shown in Fig. 5.

Our method has a similar mechanism with the work in the literature of Yang and Dong (2013), i.e., firstly use over-segment process to split the points into many small patches, then merge these patches to get meaningful segments. The difference is that our method uses super voxels as the formation of the over-segmentation that adaptively fit local details without the need of geometric features of points. Besides, compared with the SVMs based over-segmentation, the generation of super voxel is more efficient. Taking advantage of the voxel-based graph, the subsequent processing can be applied in tractable times.

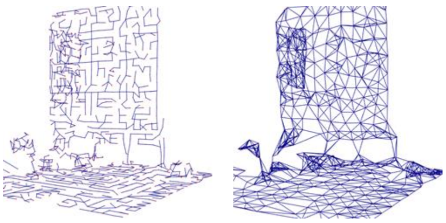

(a) MST (b) Riemannian graph

Figure 5. The construction of the voxel-based graph.

3.3 Graph Optimization for Voxel Clustering

The voxel-based graph representation consists of rich edge features encoding the contextual relationship between object parts in the point clouds. For each voxel, we compute a set of geometric features to characterize its shape. First, the normal direction values (n_x^v, n_y^v, n_z^v) are the most important feature elements. Besides, the planarity and scattering values are extracted by analysing the eigenvalues of the covariance C of all points in the voxel. Given the eigenvalues as $\lambda_1 > \lambda_2 > \lambda_3$, we adopt the normalized values defined in (Demantk et al., 2011),

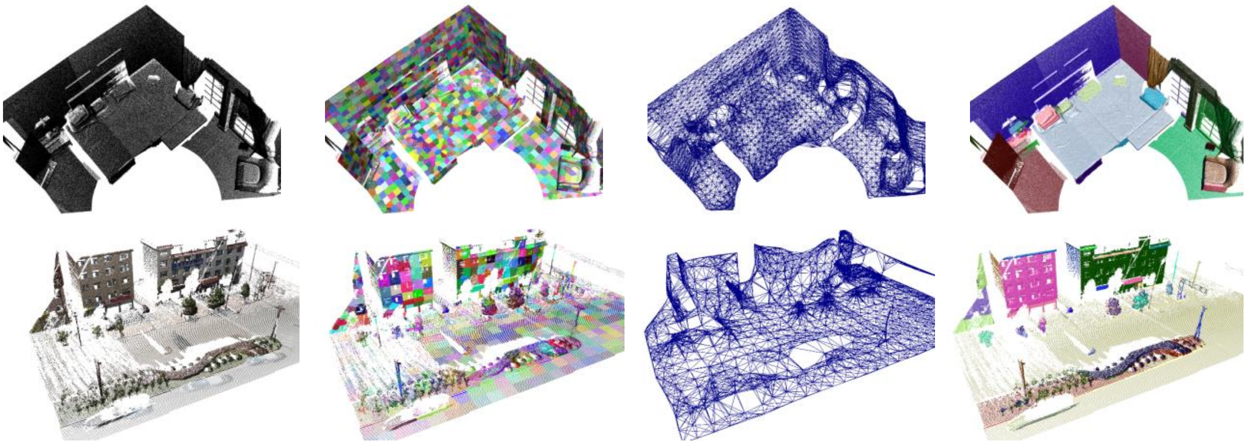
$$a_p = \frac{\lambda_2 - \lambda_3}{\lambda_1},\ a_s = \frac{\lambda_3}{\lambda_1} \tag{3}$$

where a_p and a_s respectively represent the planarity and the scattering probabilities of the voxels. Although Demantk et al. (2011) analysed the linearity value of the point as well, our research objects have converted to super voxels so the linearity value is abandoned. Finally, these values consist the feature vector $f_v : (n_x^v, n_y^v, n_z^v, a_p, a_s) \in \mathbb{R}^5$ for the voxel shape, and a graph based optimization method is designed to get the geometrically homogeneous partition. Our objective function is given as below:

$$g^* = \underset{g_i \in \mathbb{R}^5}{\arg\min} \sum_{i \in V} \| g_i - f_i \| + \mu \sum_{(i,j) \in E} \delta\left(g_i - g_j \neq 0\right) \tag{4}$$

where $g^* \in \mathbb{R}^{5 \times V}$ denotes the optimized vectors of the voxel features, g_i specifies the i th optimized feature vector, and $\delta(\cdot \neq 0)$ is the function of $\mathbb{R}^5 \mapsto \{0,1\}$ equals to 0 in 0 and 1 everywhere else. The energy function can be seen as a continuous-space version of the Potts energy model. The first part of this energy is the fidelity function, and the second part is the regularization term adding a penalty for each edge linking two components with different values. The factor μ is the regularization strength, determining the trade-off between fidelity and simplicity.

The minimization problem is non-convex, so that the problem cannot realistically be solved exactly for large dataset. However, the $\ell_0 - cut$ pursuit algorithm (Landrieu and Obozinski, 2017) is able to quickly find an approximate solution with a few graph-cut iterations. Compared with to other graph optimization methods such as $\alpha -$ expansion (Ural and Shan, 2012), the $\ell_0 - cut$ pursuit algorithm does not require selecting the number



(a) Point cloud (b) Super voxels (c) Riemannian graph (d) Segmentation result

Figure 6. Example visualizations on both datasets. The colours in (b) and (d) are chosen randomly for each partition.

of the partition in advance. The results of this optimization could be automatically adaptive to the underling scenes without the predefine characteristics for some potential objects.

4. EXPERIMENTAL RESULTS

We test our method with two dataset. The first one is a scanning of a bedroom and the second one is a large outdoor scene. The point cloud of the bedroom is obtained by a Leica C10 laser scanner from the range around 3*m*. The input point cloud is consisted of mass points with a density about 9000 points/m^2. The second terrestrial dataset is provided by LIESMARS Lab at Wuhan University, named as WHU data. In our experiment, the WHU data is collected from a variety of urban scenes which consist of many kinds of different street objects, and have a varying point density ranging from 40 points/m^2 to 200 points/m^2. Even though they are quite different in nature, our graph model could automatically adapt for both. Experiment demonstrated the advantages of the proposed method based on the voxel-based graph.

4.1 Evaluation of the Proposed Method

The proposed method was implemented on a computer with 8 GB RAM and an Intel (R) Core (TM) i7-6500U @ 2.50 GHz CPU. In Fig. 6, we provide qualitative results of our framework. Our framework improves significantly on the state of the art of segmentation for these datasets. The *k*-means method combining with an adaptive octree algorithm produces reliable super voxels as shown in Fig. 6(b) which significantly reduces the workload of the subsequent operations. During the segmentation phase, Riemannian graph connects all super voxels in a unicom component based on the approximations of the geodesic distance matrices, as shown in Fig. 6(c). Based on the features extracted from super voxels, the final graph optimization algorithm makes similar voxels merging into the semantics awareness segment results. Execution times are presented in Tab. 1.

Processing timing	Super voxels	Riemannian graph	Graph optimization
Bedroom data (451K points)	4431	1345	4766
WHU data (123K points)	5515	1235	5205

Table 1. Execution time of test (in milliseconds).

To illustrate the effect of the minimal octree depth employed in section 3.1, we compare the resulting voxel numbers and segment numbers in Tab. 2. The gain of the super voxel number depends on the minimal depth of the octree and the scene size. The increased voxels improve the performance especially for smaller shapes, as their relative size decreases and a graph optimization strategy requires far more candidates. With the number of super voxels increases, the segmentation results tend to be stable. In general the gain of the super voxel evaluation should increase with the size of the scene.

Min. depth	Bedroom data		WHU data	
	Voxels	Segments	Voxels	Segments
4	440	44	651	55
5	1861	65	2336	95
6	6833	67	8024	102
7	26947	70	23877	104

Table 2. The effect of the minimal octree depth threshold on the super voxel number and the segmentation number.

4.2 Comparison

Because there are hardly any publicly available implementations, the comparison is difficult. We implemented the octree-based region growing method presented in (Vo et al., 2015) based on their literature. The comparison is studied between our method and the octree-based region growing method, which is a region-growing based algorithm that used an octree-based voxelized representation of the input point cloud to extract major segments followed by a refined process.

As shown in Fig. 7, Octree-based region growing method could extract many major planes that reflect the distribution of the whole scene, but the many detail information will not be detected since these small shapes are often merged with the roads or facades when performing spatial regularization. On the contrary, our super voxels fit the local geometries well, even when the shapes are very small. Hence, the final segments consist of many super voxels could achieve a better segmentation performance. Even though the point cloud captured with ground lasers have different point densities depending on the distance to the sensor, our approach has the ability to distinguish different objects, such as vehicles, vegetation, street lights and billboards, who have

https://doi.org/10.5194/isprs-annals-IV-3-135-2018

singular shapes. The two algorithms have a comparable accuracy, while the proposed algorithm wins in the segment numbers benefited from the super voxels which preserve the shape characteristics of the scene.

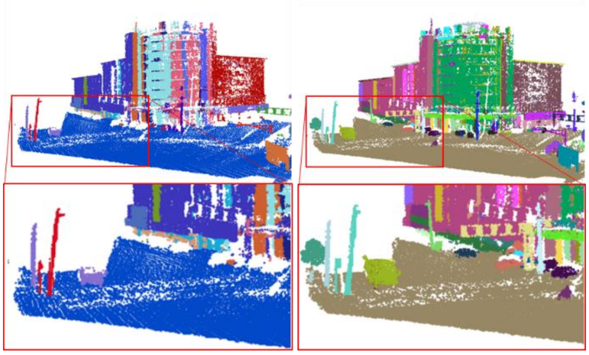


Figure 7. An experiment on vehicle borne LiDAR data. (a) The result of octree-based region growing method (Vo et al., 2015); (b) the result of the proposed method.

CONCLUSIONS

In this paper, we have shown how to embed super voxel-based nodes on a Riemannian graph. The analysis hinges on the properties of super voxels, and we show how to compute an edge-weight matrix in which the elements reflect the sectional curvatures associated with the geodesic paths. The proposed graph optimization approach employs an adaptive octree and super voxel nodes to achieve both better convergence and a more convincing point segmentation. The proposed segmentation algorithm is a novel tool to extract partition relationship in a mass point cloud. Experiments demonstrate that the method can be successfully applied to the artificial point cloud in challenging situations. In the future work, we will extend the segmentation task with machine learning methods to resolve the semantic interpretation problems.

ACKNOWLEDGEMENTS

This work was supported in part by the Natural Science Foundation of Jiangsu Province, China (Grants: BK20170781). The author would like to appreciate LIESMARS Lab at Wuhan University for making their datasets online available to us.

REFERENCES

Alismail, H., Baker, L.D., Browning, B., 2014. Continuous trajectory estimation for 3D SLAM from actuated LiDAR. In: *IEEE International Conference on Robotics and Automation (ICRA), 2014*, Hong Kong, China, 1296, pp. 6096-6101.

Boulch, A., Le Saux, B., Audebert, N., 2017. Unstructured point cloud semantic labeling using deep segmentation networks. In: *Eurographics Proceedings, Eurographics Workshop on 3D Object Retrieval*, pp.17-24.

Boyko, A., Funkhouser, T., 2011. Extracting roads from dense point clouds in large scale urban environment. *ISPRS Journal of Photogrammetry and Remote Sensing*, 66 (6), S02–S12.

Boykov, Y.Y., Jolly, M.P., 2001. Interactive graph cuts for optimal boundary & region segmentation of objects in n-d images. In: *IEEE International Conference on Computer Vision*, 2001, Vancouver, B.C., Canada, pp. 105-112.

Demantk, J., Mallet, C., David, N., Vallet, B., 2011. Dimensionality based scale selection in 3d lidar point clouds. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XXXVIII-5/W12, pp.97-102.

Golovinskiy, A., Funkhouser, T., 2009. Min-cut based segmentation of point clouds. *IEEE Workshop on Search in 3D and Video at ICCV*, Oct. 2009, Kyoto, Japan.

Guinard, S., Landrieu, L., 2017. Weakly supervised segmentation-aided classification of urban scenes from 3d LiDAR point clouds. In: *ISPRS The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XLII-1/W1, pp. 151-157.

Gorte, B., 2002. Segmentation of TIN-structured surface models. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences.* 34 (4), pp. 465-469.

Guo, Y., Bennamoun, M., Sohel, F., Wan, J., 2014. 3D Object Recognition in Cluttered Scenes with Local Surface Features: A Survey. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 36(11), pp. 2270-2287.

Laga, H., Nakajima, M., 2007. A Boosting approach to content-based 3D model retrieval. In: *International Conference on Computer Graphics & Interactive Techniques in Australia & Southeast Asia*, pp. 227-234

Landrieu, L., Obozinski, G., 2017. Cut pursuit: fast algorithms to learn piecewise constant functions on general weighted graphs. *SIAM Journal on Imaging Sciences*, 10(4), pp. 1724-1766.

Li, M., Nan, L., Liu, S., 2016. Fitting boxes to Manhattan scenes using linear integer programming. *International Journal of Digital Earth*, 9(8), pp. 806-817.

Musialski, P., Wonka, P., Aliaga, D., Wimmer, M., Van Gool, L., Purgathofer. W., 2013. A survey of urban reconstruction. *Computer Graphics Forum*, 32(6), pp. 146-177.

Nurunnabi, A., Belton, D., West, G., 2012. Robust segmentation in laser scanning 3D point cloud data. In: *IEEE International Conference on Digital Image Computing Techniques and Applications (DICTA)*, pp. 1-8.

Overby, J., Bodum, L., Kjems, E., Iisoe, P., 2004. Automatic 3D building reconstruction from airborne laser scanning and cadastral data using Hough transform. *International Archives of Photogrammetry and Remote Sensing*, 35 (B3), pp. 296-301.

Pang, G., Neumann, U., 2013. Training-based object recognition in cluttered 3D point clouds. In: *International Conference on 3D Vision*, Seattle, WA, USA, 8768(2), pp. 87-94.

Pu, S., Rutzinger, M., Vosselman, G., Oude Elberink, S., 2011. Recognizing basic structures from mobile laser scanning data for road inventory studies. *ISPRS Journal of Photogrammetry and Remote Sensing*, 66(6), pp. S28-S39.

Pu, S., Vosselman, G., 2006. Automatic extraction of building features from terrestrial laser scanning. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 36, pp. 33-39.

Robles-Kellya, A., Hancock, E., 2007. A Riemannian approach to graph embedding. *Pattern Recognition*, 2007, 40, pp. 1042-1056.

Rottensteiner, F., Sohn, G., Gerke, M., Wegner, J., Breitkopf, U., Jung. J., 2014. Results of the ISPRS benchmark on urban object detection and 3D building reconstruction. *ISPRS Journal of Photogrammetry and Remote Sensing*, 93(7), pp. 256-271.

Rusu, R., Holzbach, A., Blodow, A., Beetz, M., 2009. Fast Geometric point labeling using conditional random fields. *In Proceedings of the 22nd IEEE/RSJ International Conference on Intelligent Robots and Systems*, St. Louis, MO, USA, 2009.

Serna, A., Marcotegui, B., 2014. Detection, segmentation and classification of 3d urban objects using mathematical morphology and supervised learning. *ISPRS Journal of Photogrammetry and Remote Sensing*, 93, pp. 243-255.

Schnabel, R., Wahl, R., Klein, R., 2007. Efficient RANSAC for point cloud shape detection. *Computer Graphics Forum*, 26(2), pp. 214-226.

Tarsha-Kurdi, F., Landes, T., Grussenmeyer, P., 2007. Hough transform and extended RANSAC algorithms for automatic detection of 3D building roof planes from LiDAR data. *International Archives of Photogrammetry, Remote Sens.* pp. 407-412.

Ural, S., Shan, J., 2012. Min-cut based segmentation of airborne LiDAR point clouds. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, Melbourne, Australia, pp. 167-172.

Vieira, M., Shimada, K., 2005. Surface mesh segmentation and smooth surface extraction through region growing. *Computer Aided Geometric Design*, 22(8), pp. 771-792.

Vo, A. V., Truong-Hong, L., Laefer, D. F., Bertolotto, M., 2015. Octree-based region growing for point cloud segmentation. *ISPRS Journal of Photogrammetry and Remote Sensing*, 104, pp. 88-100.

Vosselman, G., Gorte, B., Sithole, G., Rabbani, T., 2004. Recognising structure in laser scanner point clouds. *International Archives of Photogrammetry Remote Sensing & Spatial Information Sciences*, 46(8), pp. 33-38.

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

Wu, Z., Song, S., Khosla, A., Yu, F., Zhang, L., Tang, X., Xiao, J., 2015. 3D shapenets: A deep representation for volumetric shapes. In: *IEEE Conference on Computer Vision and Pattern Recognition*, Boston, USA, 2015, pp. 1912-1920.

Yang, B., Dong, Z., A shape-based segmentation method for mobile laser scanning point clouds. ISPRS Journal of Photogrammetry & Remote Sensing, 2013, 81(7): 19-30.

Yang, B., Dong, Z., Liu, Y., Liang, F., Wang, Y., 2017. Computing multiple aggregation levels and contextual features for road facilities recognition using mobile laser scanning data. *ISPRS Journal of Photogrammetry and Remote Sensing*, 126, pp.180-149.

Yu, Y., Li, J., Guan, H., Jia, F., Wang, C., 2015. Learning hierarchical features for automated extraction of road markings from 3-D mobile LiDAR point clouds. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 8(2), pp. 709-726.

Zhou, L., Vosselman, G., 2012. Mapping curbstones in airborne and mobile laser scanning data. *International Journal of Applied Earth Observations & Geoinformation*, 18 (1), pp. 293-304.