

# CONTOUR EXTRACTION OF PLANAR ELEMENTS OF BUILDING FACADES FROM POINT CLOUDS USING GLOBAL GRAPH-BASED CLUSTERING

Y. Xu\*, U. Stilla

Photogrammetry and Remote Sensing, Technical University of Munich (TUM), Germany - (yusheng.xu, stilla)@tum.de

ICWG II/III: Pattern Analysis in Remote Sensing

**KEY WORDS:** Contour extraction, point clouds, planar building elements, global graph clustering

## ABSTRACT:

In this work, we present a surface-based method to extract the contours of planar building elements in the urban scene. A bottom-up segmentation method that utilizes global graph-based optimization and supervoxel structure is developed, enabling an automatic and unsupervised segmentation of point clouds. Then, a planarity-based extraction is conducted to segments, and only the planar segments, as well as their neighborhoods, are selected as candidates for the plane fitting. The points of the plane can be identified by the parametric model given by the planarity calculation. Afterward, the boundary points of the extracted plane are extracted by the alpha-shape. Optionally, line segments can be fitted and optimized by the energy minimization with the local graphical model. The experimental results using different datasets reveal that our proposed segmentation methods can be effective and comparable with other method, and the contours of planar building elements can be well extracted from the complex urban scene.

## 1. INTRODUCTION

Recently, light detection and ranging (LiDAR) technology has been widely used for acquiring geospatial information in urban scenarios (Pu et al., 2011, Rutzinger et al., 2011). Generally, unstructured 3D point clouds are used to represent the acquired geospatial information, which is usually of high density and large volume (Lin et al., 2017). However, using individual points to directly describe the 3D scene is not a practical solution since the topological information is missing, which cannot meet the demand of delineating the urban scene. Besides, the large-scale point cloud is a great challenge in data processing. Comparing with points, lines, edges, and contours can be regarded as better representations delineating the scene, especially in the urban area existing lots of regular shaped man-made structures (Lu et al., 2019).

Detecting line segments and contours from 2D images have been thoroughly studied and reported (Hong et al., 2015). However, to extract line segments and contours from 3D point clouds, there is still room for improvement. To extract these line segments or contours from unstructured point clouds, there are several different strategies, including two major ones: point-based strategy and surface-based strategy. To be specific, point-based strategies will directly detect those points belonging to the edges and boundaries, and then connect them to form the lines or contours. Classifying the whole point cloud in the feature space via designed features and classifier is one of the typical methods for extracting edge points (Lu et al., 2019). In the last decades, a wide variety of features has been developed for boundary/non-boundary point cloud classification. In (Maas, Vosselman, 1999, Hackel et al., 2016), eigenvalues and eigenvectors derived from the covariance matrix of points coordinates have been introduced to distinguish points of edges via a binary classifier. In (Ioannou et al., 2012), the difference of normal vectors calculated with multi-scale local neighborhoods is utilized to detect those points of edges. Similarly, in (Bazazian et al., 2015) the curvature or surface variation is also

used to detect edge points. Besides, robust statistics reflecting sharp features are also employed to detect edge points, such as GaussMap (Weber et al., 2010), moving least squares (Fleishman et al., 2005), and spline (Daniels Ii et al., 2008). Once the edge points are extracted, methods like Least Square Fitting (Liu et al., 2005) or cell decomposition (Kada, McKinley, 2009) can be applied to connect those points into complete contours or lines. The surface-based strategy is an alternative for extracting lines and contours. Since the intersection of two surfaces can construct lines and curves, if we can find out the intersecting surfaces, curves and lines can be found. The surfaces can be achieved by segmentation or over-segmentation like region growing (Rabbani et al., 2006) and model fitting (Nurunnabi et al., 2012). In (Lin et al., 2015, Lin et al., 2017), the intersections of planes (e.g., line-segment-half-planes) and facets (e.g., supervoxels) are used to extract lines. The advantage of these methods is that there is no need to fit the lines or curves as the intersection has embedded the topological information. Besides, the boundary of isolated surfaces is also considered as contours to be extracted. For the isolated surfaces, convex hull (Lin et al., 2017), rotate calipers (Toussaint, 1983), or alpha shape (Edelsbrunner et al., 1983) can be used to extract the contours. However, for surface-based methods, it is difficult to determine the terminals of intersection lines and hard to be applied to small surfaces (Lu et al., 2019).

To address existing problems, we present a surface-based method to extract the contours of planar building elements in the urban scene. A bottom-up segmentation method that utilizes global graph-based optimization and supervoxel structure is developed, enabling an automatic and unsupervised segmentation of point clouds. Then, a planarity-based extraction is conducted to segments, and only the planar segments, as well as their neighborhoods, are selected as candidates for the plane fitting. The points of the plane can be identified by the parametric model given by the planarity calculation. Afterward, the boundary points of the extracted plane are extracted by the alpha-shape. Optionally, line segments can be fitted and optimized by the energy minimization with the local graphical model. The inno-

\*Corresponding author

vative contributions that specific to our proposed approach are: (1) A bottom-up point cloud segmentation method that utilizes supervoxel structure and global graph-based optimization. By using a supervoxel structure instead of points to organize the entire point cloud, over-segmented supervoxels can identify the boundaries of 3D objects. The global graphical model is constructed based on the geometric features of supervoxels. The unsupervised clustering is conducted via the optimization of the global graph. (2) A planarity-based selection and model-fitting based refinement for the detection and extraction of planar surfaces is developed, which provide accurate boundaries for contour extraction using alpha-shape. Unlike traditional model fitting based planar extraction method, without iterative process, our plane extraction method is more efficient and adaptive to the real condition of urban scenes. The calculation of smoothness and planarity can provide the estimation of coefficients for the plane model.

## 2. METHODOLOGY

Conceptually, the implementation of our proposed plane reconstruction method consists of two major phases: detection and extraction of planar segments and geometric modeling of planar segments. To be specific, the first phase can be divided into the segmentation of the point cloud and the detection of planar surfaces. For the segmentation, we propose a bottom-up point cloud segmentation method that utilizes supervoxel structure and global graph-based optimization, enabling an automatic and unsupervised segmentation of point clouds. In the subsequent step, a planarity-based extraction is conducted to segments, and only the planar segments, as well as their neighborhoods, are selected as candidates for the plane fitting. The points of the plane can be identified by the parametric model given by the planarity calculation. Afterward, the boundary points of the extracted plane are extracted by the alpha-shape. Line segments are extracted and merged by the mean-shift clustering. For the geometric modeling of planes, a cell decomposition method is adopted to get the polygon representation of extracted planes. In Figure 1, the processing workflow is sketched, with the core steps of involved methods and sample results illustrated. The detailed explanation of each step will be introduced in the following sections.

### 2.1 Geometric feature of supervoxels

To organize the entire point cloud into a supervoxel structure, the space is firstly divided into a small 3D cubic grid by means of octree partitioning, which splits each node into eight equal child nodes, in order to generate the octree-based voxel structure. Then, the geometric feature of each supervoxel consists of three parts: spatial position, orientation, and local geometry, which are the unary features abstracted from the points within it. To obtain the attribute, we firstly adopt the assumption of implicit plane representation (Dutta et al., 2014) to represent the structural patch, defining an approximate plane via the normal vector  $\vec{N}_i$  and centroid  $\vec{X}_i$  of the point sets  $P_i$  within the patch  $V_i$ .

$$\langle \vec{N}_i, \vec{X}_i \rangle - d_i^c = 0 \quad (1)$$

where  $d_i^c$  stands for the distance from the origin to the approximate plane. The spatial position stands for spatial coordinates of the centroid  $\vec{X} = (x, y, z)$  for the points set  $P = \{p_1, p_2, \dots, p_n\}$  inside the patch. The orientation represents the normal vector  $\vec{N} = (n_x, n_y, n_z)$  of the approximated surface formed by  $P$ . While geometric features refer to four of the eigenvalue-based

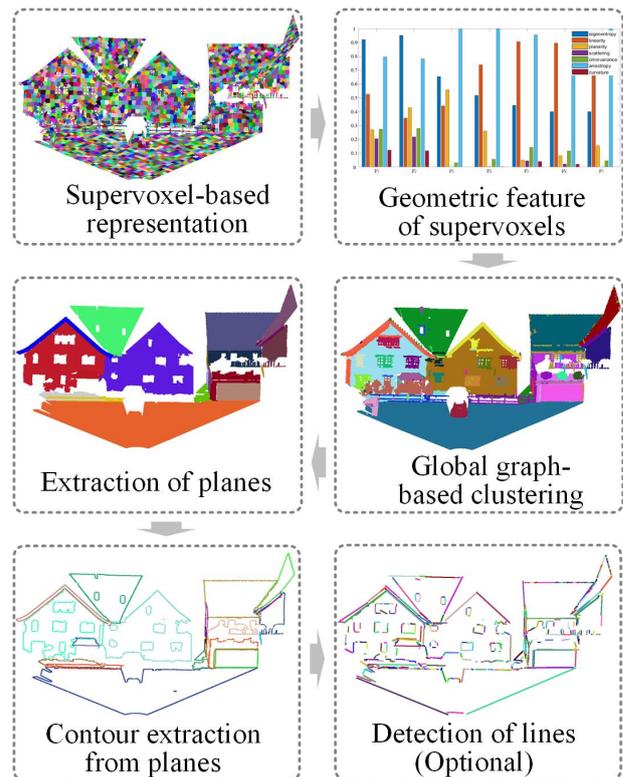


Figure 1. Workflow of the proposed method.

covariance features (Weinmann et al., 2015) encapsulating linearity  $L_e$ , planarity  $P_e$ , variation of curvature  $C_e$ , and sphericity  $S_e$ , which are calculated by the eigenvalue  $e_1 \geq e_2 \geq e_3 \geq 0$ , via the eigenvalue decomposition (EVD) of the 3D structure tensor, namely the covariance matrix  $\vec{M} \in R^{3 \times 3}$  of points coordinates of  $P$ .

### 2.2 Construction of global graphical model

For 3D point analysis, the structure of the global graph is to represent the similarity between nodes connected with edges. In this global graphical model, the node stands for the supervoxel generated from points, while the edge connecting nodes are assigned with the weight of affinity. The structure of the graph matters the representation of the topology of the 3D scene, to simplify the graph structure, we built the affinity graph based on the spatial connection between supervoxels, which is based on the KNN graph developed in (Funkhouser, Golovinsky, 2009). Here, the connection between supervoxels is identified by the check of sharing boundaries.

### 2.3 Global graph-based clustering

In the field of computer vision, the clustering of points is also formulated as graph construction and partitioning problems. The graphical model can explicitly represent points with a mathematical sound structure (Peng et al., 2013), utilizing context for deducing hidden information from given observations (Yao et al., 2010). Graph-based clustering aims to divide a dataset into disjoint subsets with members similar to each other from the affinity matrix. In (Xu et al., 2018b), the use of the local graph structure for the description of the 3D geometry with the supervoxel structure has been tested. The use of the local graph model can make the clustering process quite efficient and

available for parallel computing when combined with region-growing strategy. However, the local graph structure can merely encode the local geometry information, which can hardly represent the optimal in the global scale, so that over-segmentation frequently occurs when dealing with surfaces with irregular geometric shapes (e.g., points of vegetation). To tackle the drawbacks of local graph model, we developed the global graph-based clustering, which constructs a global graph model to describe the local characteristics of 3D scenes with different complexities, and details of objects are preserved among the clustered nodes. By clustering the nodes  $V$  into cliques  $C$ , the supervoxels clustered in the same cliques will be merged into a single segment  $S$  of points. In Figure 2, we illustrate this global graph-based clustering process.

Once the global graph of all the supervoxels is constructed, we can optimize the connection of each supervoxel by clustering nodes of the constructed global graph. To this end, similarly to work in (Xu et al., 2018b), we resolve the graph clustering problem via the adaption of the efficient graph-based segmentation method proposed in (Felzenszwalb, Huttenlocher, 2004). After the connections of all the voxels are identified, the connected voxels are clustered into one segment. This clustering process is performed repeatedly by traversing all the voxels with a depth-first strategy. All the connected voxels are aggregated into one segment. Additionally, a cross-validation process is required to examine the correctness of connections. In detail, for adjacent  $V_i$  and  $V_j$ , after segmenting the graph of  $V_i$ , if  $V_i$  is identified as connected to  $V_j$ , then in the segmentation of graph of  $V_j$ ,  $V_j$  should be connected to  $V_i$  in turn. Otherwise, they are identified as disconnected ones.

#### 2.4 Extraction of planes

Once the segments are obtained, for each segment, the smoothness and the curvature of the surface will be calculated by the eigenvalue  $e_1 \geq e_2 \geq e_3 \geq 0$  from the EVD of the 3D structure tensor of points coordinates.

$$M_e = (e_1 - e_2)/e_1 \quad (2)$$

$$C_e = e_3/(e_1 + e_2 + e_3) \quad (3)$$

where  $M_e$  stands for the smoothness and  $C_e$  stands for the curvature. The segment with the smoothness and curvature following given thresholds are extracted as the planar segments. Here, the planarity is not used, because what we want to extracted is not only evenly isotropic planes, but also those anisotropic ones.

The supervoxels of the planar segment will be considered as planar supervoxels, and points within these supervoxels are regarded as candidate points of the extracted plane. By the EVD calculation, the centroid and the normal vector of the segment are achieved as well, which will be used as the coefficients of the plane model. Using these coefficients as initial values, all the candidate points are examined by the RANSAC process (Schnabel et al., 2007), for estimating the optimized plane model of the planar segment. Since the initial values are approximately fitted to the plane models, the RANSAC process can find the inliers efficiently. It is noted that, for the planar supervoxels of one planar segment, the points of their neighboring supervoxels located at the outer boundary of the segment are included as the candidate points for the refinement of the extracted plane. This is designed for overcoming the “zig-zag” edges caused by the voxel-based segmentation methods (Sun et

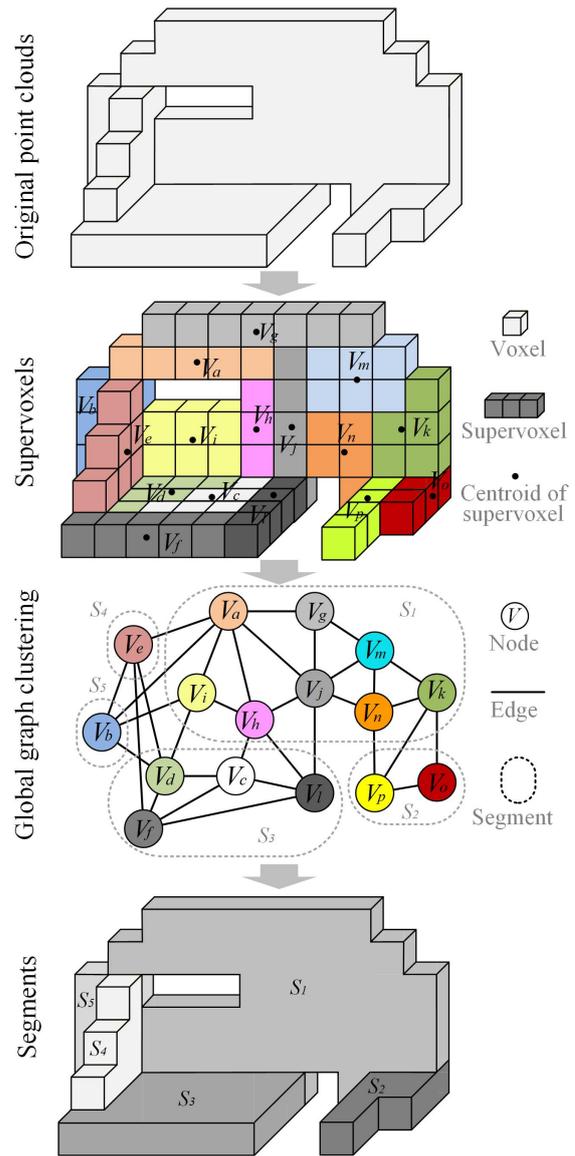


Figure 2. global graph-based clustering.

al., 2018). The coefficients of the refined plane model will be calculated by the least square algorithm using the inliers of the RANSAC process. At last, the method of plane grouping (Xu et al., 2018a) is applied to merge these neighboring planes having co-planarity.

#### 2.5 Contour extraction from planar segments

For extracted planar segments, 3D points set  $P_3$  will be firstly projected to the 2D plane of this segment with the transformation matrix  $\vec{T}$ . Here, the transformation is defined by the bounding box of segments and the vertical direction. With the alpha-shape algorithm, these projected 2D points set  $P_2$  will provide the 2D contour  $B_2$  of the segments. Then, the points of 3D contour  $B_3$  of the segment can be achieved by  $\vec{T}^{-1}B_2$ . Here, the alpha shape algorithm (Edelsbrunner et al., 1983) has been used in determining the boundaries from points of a 2D segment especially the boundaries of convex objects. In our case, the alpha shape algorithm can reduce the redundancy of the initial linear structure which can benefit the subsequent linear extraction and refinement. For the alpha shape algorithm, an alpha value ( $0 < \alpha < \infty$ ) is a parameter imposing the precision of the final

boundary. A large value ( $\alpha \rightarrow \infty$ ) results in the alpha boundary of a convex hull while a small value ( $\alpha \rightarrow 0$ ) means that every point can be the boundary points. In Figure 3, we illustrate the boundary points detected by the alpha shape algorithm with different alpha values.

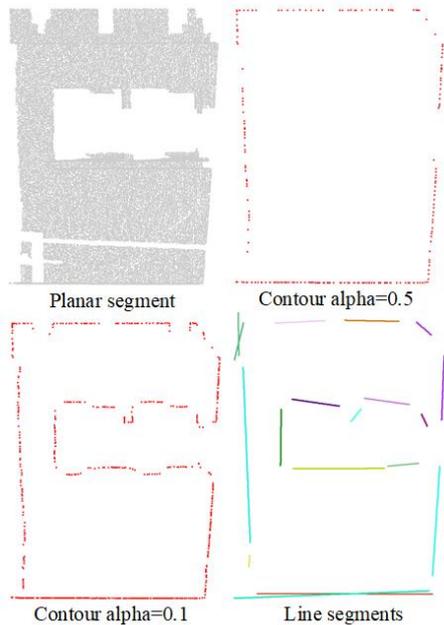


Figure 3. (a) Points of a planar segment. (b) and (c) Extracted contours with different alpha values.(d) Fitted line segments.

## 2.6 Detection of line segments (Optional)

Given the points of segments contours from the previous steps, we perform the RANSAC algorithm (Fischler, Bolles, 1981) to fit the potential line segment candidates. The reason of choosing RANSAC is due to that it can generate more complete line segments and more robust to noise and outliers than other methods like point-to-point vectors. In order to reduce the effects of outliers and fine structures such as irregular bumps and craters in the 2D floor plan, we discard the lines whose supporting points are less than a threshold of  $n_r$ . We define  $L = \{l_k | k \in 1, \dots, m\}$  as the detected line segment candidates from the contour points sets  $B_3$ . For each line segment  $l$ , all the neighboring line segments having similar orientation angles in a give neighbor region will be regarded as the neighboring set  $N(l)$ . In Figure 4, we illustration of the selection of neighboring set in the neighborhood.

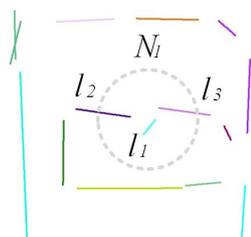


Figure 4. Neighboring set  $N_1 = \{l_1, l_2, l_3\}$  of a line segment  $l_1$ .

To eliminate the redundant line segment candidates and obtain the real and concise line segment representation, we further re-

fine the orientations of fitted line segments, so that the refined line segments can be merged into complete lines with smooth connections. Similar to the approaches (Poullis, 2013), we refine the detected line segment using the regularization of orientations. To regularize orientation of line segments. We can convert the problem of determining line orientations to a classification (i.e., a labeling task) problem of assigning lines with predefined orientations, which can be formulated as an optimization of labels and then solved by Graph Cuts algorithm (Kolmogorov, Zabih, 2004). To be specific, for each line segment, the orientation  $\theta_p$  is directly achieved by direction parameters of its line model. Here, we make an assumption that line segments constructing the same polygon may only have a limited number of orientation angles (i.e., labels in the energy function). In other words, edges are encouraged to be parallel or perpendicular with longer ones (Xie et al., 2017). Moreover, we also assume that the refined angles should not have a large deviation from its initial angles (Xie et al., 2017).

Based on these two assumptions, we first define a set of orientation angles  $\Phi$  for all the line segments. Then, for each line segment  $l$ , a candidate line segments set  $L$  is generated sharing the same center of line but having different orientation angles in  $\Phi$ . Afterward, the cost function encoding both the smoothness between neighbors and the degrees of line orientations is constructed, which is similar to the work of (Xie et al., 2017). By solving this cost function, we can determine regularized orientation angles of line segments. Here, the cost function is give as follows:

$$E = \sum_{p \in P, \theta \in \Phi} \lambda \cdot D(p, \theta) + \sum_{(p,q) \in N} S(\theta_p, \theta_q) \quad (4)$$

In this cost function,  $\theta$  stands for the initial orientation angle of a given line segment. The data term is defined by line fitting residuals.  $D(p, \theta)$  denotes the residual when comparing the orientation of the line segment to the regularized orientation. Here, this residual is measured by calculating the distance between the points of the line segment to the regularized line, which is illustrated as follows (Xie et al., 2017):

$$D(p, \theta) = \sum_{o_i} l_{\perp}(o_i, \theta) \quad (5)$$

where,  $o_i$  is the  $i$ th point in line segment  $p$ , while  $l_{\perp}(o_i, \theta)$  denotes the perpendicular distance from point  $o_i$  to line segment  $p$  having orientation angle of  $\theta$ . In the smooth term, which penalize adjacent line segments having large differences between initial orientation angles,  $N$  is the neighboring set of a line segment in Fig. 4. Here,  $\lambda$  balances the weights of the data term, which is a scale factor. Here,  $\delta$  is the normalization angle value calculated by  $\theta$  of all line segments in  $N$ . For considering the right angle connection between lines, we also make augmentation of initial orientation angles in both perpendicular and diagonal directions.

$$S(\theta_p, \theta_q) = e^{-\frac{(\theta_p - \theta_q)^2}{\delta^2}} \quad (6)$$

After the minimization of this cost function with Graph-Cut, the labeling result are translated to the corresponding orientation angles. Those line segments with the same orientation angles are merged to form a new one and corners are the intersection of two unparallelled or perpendicular line segments.

### 3. EXPERIMENTAL RESULTS

#### 3.1 Testing datasets

To test our proposed method, we use two different datasets from both mobile laser scanning (MLS) and terrestrial laser scanning (TLS). The MLS dataset is measured at the Arcisstrasse along the main entrance of Technical University of Munich (TUM) city campus, which covers about an area of around  $29000\text{ m}^2$  and has been already displayed in Figure 5a. Fraunhofer Institute of Optronics originally acquires this dataset, System Technologies and Image Exploitation (IOSB) (Gehring et al., 2017). Two Velodyne HDL-64E acquires the used point clouds mounted at an angle of  $35^\circ$  on the front roof of the vehicle.

The TLS point clouds are from the large-scale point cloud classification benchmark dataset published on [www.semantic3d.net](http://www.semantic3d.net) by ETH Zurich (Hackel et al., 2017), which covers a wide variety of diverse building scenes like churches, streets, squares, villages, and castles. Specifically, two point clouds of different scenes are tested (see Figure 5b and Figure 5c): one is scanned in the area of the cathedral of St. Gallen, whereas the other one is measured in the area of a town square. A clipping process is also conducted to remove the irrelevant and distant parts in the point cloud of the scene. For the original point cloud shown in Figs. 5b and c, the color represents the intensity of laser reflections, with brighter color showing stronger intensities. In our current work, the intensity of the point is not involved. Noise and outliers are kept in the datasets.

The MLS dataset is also used for a brief evaluation of segmentation performance. Manual segmentation from (Xu et al., 2018c) is used as references (see Figs. 5d- 5g) in a way similar to the work in (Vo et al., 2015). Namely, each reference dataset are segmented independently by persons who are familiar with point cloud segmentation work. Then, automatic segmentation results will be compared against two reference datasets of the same scene individually. For the reference datasets 1, there are in total 101 and 66 segments obtained for the scenes of St. Gallen cathedral and Townsquare, respectively. While for the reference datasets 2, there are 100 and 84 segments obtained for the scenes of St. Gallen cathedral and Townsquare, respectively.

In the experiments, all the algorithms are implemented via C++ with the help of PCL 1.8.0. The Graph Cuts algorithm for optional line refinement is achieved via the GCO-v3.0 library (Kolmogorov, Zabih, 2004, Boykov, Kolmogorov, 2004). All experiments run on an Intel i7-6700 CPU @ 3.4GHz and with 32.0 GB RAM. The parameters we used are set empirically.

#### 3.2 Point cloud segmentation

We first test our proposed global graph-based clustering method on the reference datasets, which has been mentioned and used in (Xu et al., 2018c). The voxel sizes used in our is set to 0.1m, and seed resolutions of supervoxels in our proposed method, Locally Convex Connected Patches (LCCP), and Supervoxel and graph-based segmentation (SVGS) are both set to 0.25 m. The size of the neighborhood for aggregating the supervoxels is set to 0.5m. The threshold for graph segmentation is set to 0.85. In Table 1, we provide the comparison of results using the local graph structure (i.e., unsupervised hierarchical clustering) and the global graph structure (i.e., global graph-based clustering). As shown in the table, it is clear that the

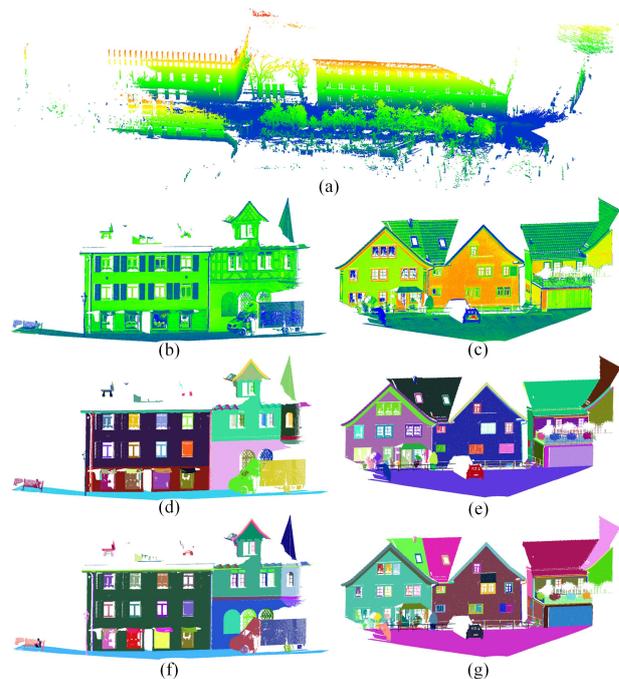


Figure 5. Testing datasets. Original point clouds of (a) Arcisstrasse, (b) St. Gallen, and (c) Town square. Reference 1 of (d) St. Gallen and (e) Town Square. Reference 2 of (f) St. Gallen and (g) Town Square.

new global graph-based clustering method has comparable performance as the unsupervised hierarchical clustering method, but it seems that the result of global graph-based clustering can provide better recall values, especially for the tests of using the scene Town square. This reveals that the global graph-based clustering method prefers to generate under-segmented segments, but the accuracy of found boundaries of segments still needs to be improved. When clustering the global graphical model, the inner difference of a clique is less essential compared to the difference between cliques of different objects, resulting in a better ability of clustering points from the same object but having irregular shapes (e.g., bushes, flowers, and window frames). Benefiting from this advantage, global graph-based clustering is quite suitable for applying on datasets with a scene of mixture objects, namely urban scenes containing buildings, vegetation, and vehicles. In Figure 6a, we illustrate the result of segmenting the MLS dataset of the area near Arcisstrasse of TUM city campus using global graph-based clustering. This dataset is contaminated with outliers and points of moving vehicles and pedestrians. As seen from the figure, major structures like facades, ground surface, vehicles, tree stems, and walls are well separated. Considering the semantic labels obtained in the semantic labeling step, we can easily extract individual building structural components from the entire scanned point cloud.

In addition, the framework of global graph-based clustering can be elegantly applied to any kind of elements from different data structures. For example, we can directly regard each voxel as a node in the graphical model without conducting the supervoxelization process. The weighted edges between nodes are determined by the affinity of geometric features of connecting nodes. In Figure 6, we display an example of using voxels as basic elements under the global graph-based clustering framework. Compared with our former voxel-based method, the optimization at the global scale can provide a good ability of clus-

Table 1. Comparison of segmentation results using global graph-based clustering (global graphical model) and the unsupervised hierarchical clustering (local graphical model).

Test scenes	global graph-based clustering			Unsupervised hierarchical clustering		
	$P_r$	$R_e$	$F_1$	$P_r$	$R_e$	$F_1$
St. Gallen (Ref1)	0.7004	0.7563	0.7273	0.8769	0.6352	0.7367
St. Gallen (Ref2)	0.7134	0.6571	0.6841	0.7381	0.5988	0.6612
Town square (Ref1)	0.5911	0.8369	0.6928	0.6840	0.6847	0.6843
Town square (Ref2)	0.5841	0.8126	0.6796	0.6904	0.6510	0.6701

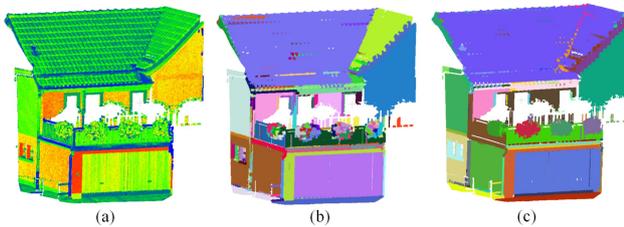


Figure 6. Comparison of segmentation results using. (a) Original point clouds, (b) segmentation results using method from (Xu et al., 2018b), and (c) segmentation results using voxel-based global graph-based clustering.

tering points of objects with irregular shapes, and at meanwhile, preserve the boundaries of objects well. This phenomenon can be clearly observed from the comparison of segmented flowers and fences on the balcony shown in Figs. 6b and 6c. For our former voxel-based method, the flowers can only be clustered as a set of small cliques of points without completeness. In contrast, when utilizing the voxel-based global graph-based clustering method, all the flower clusters can be separated neatly. The point densities are also critical for the segmentation. However, by using voxel structure to organize the point cloud, the geometry of points within the voxels can be represented by the approximated plane (Xu et al., 2017), which can help us to resist the unevenly distributed points.

### 3.3 Plane extraction

Based on the segmentation result, we conduct the extraction of planes from these segments. In this step, the smoothness and the curvature thresholds are both set to 0.01. Whereas the threshold of RANSAC fitting is set to 0.1m. In Table 2, we give the numbers of extracted planar segments for the aforementioned three testing datasets, reporting that finally there are 10, 27, and 17 planes extracted from the above-mentioned testing scenes, respectively.

Table 2. Number of extracted planes.

Scenes	TUM	St. Gallen	Town square
Original points	1886937	3336659	2359831
Supervoxels	17559	14537	21016
Segments	1057	371	308
Initial planes	10	48	20
Merged planes	10	27	17

In Figure 7, the illustration of extracted planes is given. As seen from the table and the figure, we can find that the major planar structures of the scene are extracted, and small planar segments have been merged into a large complete surface. However, the merging step is a double-blade sword, which can optimize the completeness of the extracted plane and at the meanwhile, the merging is counterproductive to the separation of co-planar objects, for example, the facades of two houses in Figure 7c are

merged as one large facade. In addition, when using RANSAC to refine the planar segment, the fitting threshold is crucial to the number of extracted planes. In some cases (e.g., the left facade in Figure 7b), multiple overlapped planes may be generated from points of the same planar object.

### 3.4 Contour extraction

Once the planes of the major structures are extracted, the contour hulls are extracted by the use of the alpha-shape algorithm. Here, the alpha value is set to 0.1m for the datasets of scenes Town Square and St. Gallen, while for the dataset of the scene Arcisstrasse, the alpha value is set to 0.2 m. In Figure 7, we illustrate the extracted contour hulls of planes in these three scenes. As shown in the figure, the extracted contour hulls have covered salient boundaries of the planar object. However, for the area with unevenly distributed densities of points, especially in the scene Arcisstrasse, the data of which is measured by MLS, the extracted hulls follow the patterns formed by the scanning lines instead of the borders of the object. In Table 3, we give the numbers of extracted contours and contour points.

Table 3. Number of contour points.

Scenes	TUM	St. Gallen	Town square
Original points	1886937	3336659	2359831
Extracted contours	10	27	17
Contour points	26253	26814	18078

## 4. CONCLUSION

In this work, we present a surface-based method to extract the contours of planar building elements in the urban scene. A bottom-up segmentation method that utilizes global graph-based optimization and supervoxel structure is developed, enabling an automatic and unsupervised segmentation of point clouds. Then, a planarity-based extraction is conducted to segments, and only the planar segments, as well as their neighborhoods, are selected as candidates for the plane fitting. The points of the plane can be identified by the parametric model given by the planarity calculation. Afterward, the boundary points of the extracted plane are extracted by the alpha-shape. Optionally, line segments can be fitted and optimized by the energy minimization with the local graphical model. The experimental results using different datasets reveal that our proposed segmentation methods can be effective and comparable with other method, and the contours of planar building elements can be well extracted from the complex urban scene. In the future, we will try to improve the method, so that not only the contours of planes can be extracted, but also those of irregular surfaces. Moreover, the feature representation (e.g. for lines or curves) removing redundant information with dimension reduction (Hong et al., 2018) or projecting to low dimensional space (Huang et al., 2019), or transferring to other domain (Tong et al., 2015), will also be considered in our further studies, in order to extract those essential and representative structural features.

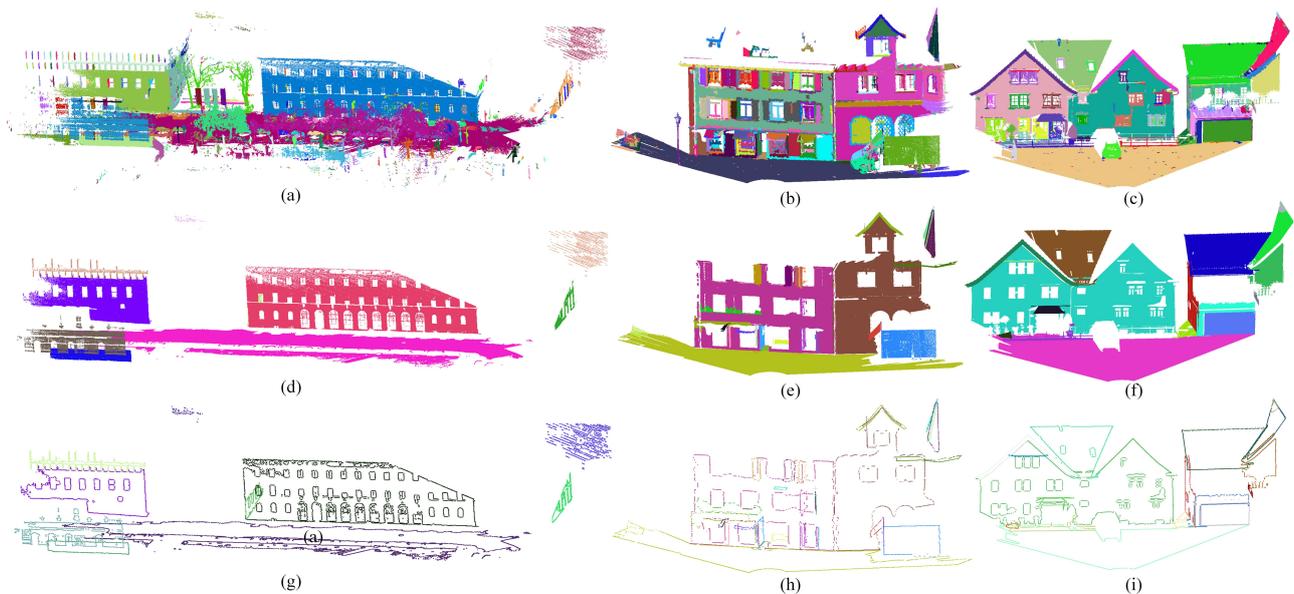


Figure 7. Illustration of segmentation results using global graph-based clustering. (a) Segmentation result of the TUM scene (Arcisstrasse), (b) segmentation results of the Town Square scene, and (c) segmentation results of the St. Gallen scene. Illustration of extracted planes. (d) Extracted planes of the TUM scene (Arcisstrasse), (e) extracted planes of the Town Square scene, and (f) extracted planes of the St. Gallen scene. Illustration of extracted hulls. (g) Extracted hulls of the TUM scene (Arcisstrasse), (h) extracted hulls of the Town Square scene, and (i) extracted hulls of the St. Gallen scene.

## REFERENCES

Bazazian, D., Casas, J. R., Ruiz-Hidalgo, J., 2015. Fast and robust edge extraction in unorganized point clouds. *2015 International Conference on Digital Image Computing: Techniques and Applications (DICTA)*, IEEE, 1–8.

Boykov, Y., Kolmogorov, V., 2004. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(9), 1124–1137.

Daniels Ii, J., Ochotta, T., Ha, L. K., Silva, C. T., 2008. Spline-based feature curves from point-sampled geometry. *The Visual Computer*, 24(6), 449–462.

Dutta, A., Engels, J., Hahn, M., 2014. A distance-weighted Graph-Cut method for the segmentation of laser point clouds. *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XL-3(3), 81–88.

Edelsbrunner, H., Kirkpatrick, D., Seidel, R., 1983. On the shape of a set of points in the plane. *IEEE Transactions on Information Theory*, 29(4), 551–559.

Felzenszwalb, P. F., Huttenlocher, D. P., 2004. Efficient graph-based image segmentation. *International Journal of Computer Vision*, 59(2), 167–181.

Fischler, M. A., Bolles, R. C., 1981. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6), 381–395.

Fleishman, S., Cohen-Or, D., Silva, C. T., 2005. Robust moving least-squares fitting with sharp features. *ACM*

*transactions on graphics (TOG)*, 24number 3, ACM, 544–552.

Funkhouser, A., Golovinsky, A., 2009. Min-cut based segmentation of point clouds. *Proc. Int. Conf. Workshop Comput. Vision*, 39–46.

Gehring, 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 Photogrammetry, Remote Sensing and Spatial Information Sciences*, IV-1/W1, 107–114.

Hackel, T., Savinov, N., Ladicky, L., Wegner, J. D., Schindler, K., Pollefeys, M., 2017. Semantic3D. net: A new Large-scale Point Cloud Classification Benchmark. *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*, IV-1/W1, 91–98.

Hackel, T., Wegner, J. D., Schindler, K., 2016. Contour detection in unstructured 3d point clouds. *IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, 1610–1618.

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., Xu, J., Zhu, X., 2018. Joint & progressive learning from high-dimensional data for multi-label classification. *Proceedings of the European Conference on Computer Vision (ECCV)*, 469–484.

Huang, R., Ye, Z., Boerner, R., Yao, W., Xu, Y., Stilla, U., 2019. FAST PAIRWISE COARSE REGISTRATION BETWEEN POINT CLOUDS OF CONSTRUCTION SITES USING 2D PROJECTION BASED PHASE

- CORRELATION. *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XLII-2/W13, 1015–1020. <https://www.int-arch-photogramm-remote-sens-spatial-inf-sci.net/XLII-2-W13/1015/2019/>.
- Ioannou, Y., Taati, B., Harrap, R., Greenspan, M., 2012. Difference of normals as a multi-scale operator in unorganized point clouds. *International Conference on 3D Imaging, Modeling, Processing, Visualization & Transmission, 3DIMPVT '12*, IEEE Computer Society, Washington, DC, USA, 501–508.
- Kada, M., McKinley, L., 2009. 3D building reconstruction from LiDAR based on a cell decomposition approach. *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 38(Part 3), W4.
- Kolmogorov, V., Zabih, R., 2004. What energy functions can be minimized via graph cuts? *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(2), 147–159.
- Lin, Y., Wang, C., Chen, B., Zai, D., Li, J., 2017. Facet Segmentation-Based Line Segment Extraction for Large-Scale Point Clouds. *IEEE Transactions on Geoscience and Remote Sensing*, 55(9), 4839–4854.
- Lin, Y., Wang, C., Cheng, J., Chen, B., Jia, F., Chen, Z., Li, J., 2015. Line segment extraction for large scale unorganized point clouds. *ISPRS Journal of Photogrammetry and Remote Sensing*, 102, 172–183.
- Liu, Y., Yang, H., Wang, W., 2005. Reconstructing b-spline curves from point clouds—a tangential flow approach using least squares minimization. *International Conference on Shape Modeling and Applications 2005 (SMI'05)*, IEEE, 4–12.
- Lu, X., Liu, Y., Li, K., 2019. Fast 3D Line Segment Detection From Unorganized Point Cloud. *arXiv preprint arXiv:1901.02532*.
- Maas, H.-G., Vosselman, G., 1999. Two algorithms for extracting building models from raw laser altimetry data. *ISPRS Journal of photogrammetry and remote sensing*, 54(2-3), 153–163.
- Nurunnabi, A., Belton, D., West, G., 2012. Robust segmentation for multiple planar surface extraction in laser scanning 3d point cloud data. *Pattern Recognition (ICPR), 2012 21st International Conference on*, IEEE, 1367–1370.
- Peng, B., Zhang, L., Zhang, D., 2013. A survey of graph theoretical approaches to image segmentation. *Pattern Recognition*, 46(3), 1020–1038.
- Poullis, C., 2013. A framework for automatic modeling from pointcloud data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1.
- Pu, S., Rutzinger, M., Vosselman, G., Elberink, S. O., 2011. Recognizing basic structures from mobile laser scanning data for road inventory studies. *ISPRS Journal of Photogrammetry and Remote Sensing*, 66(6), S28–S39.
- Rabbani, T., Van Den Heuvel, F., Vosselman, G., 2006. Segmentation of point clouds using smoothness constraint. *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 36(5), 248–253.
- Rutzinger, M., Höfle, B., Oude Elberink, S., Vosselman, G., 2011. Feasibility of facade footprint extraction from mobile laser scanning data. *Photogrammetrie-Fernerkundung-Geoinformation*, 2011(3), 97–107.
- Schnabel, R., Wahl, R., Klein, R., 2007. Efficient RANSAC for point-cloud shape detection. *Computer Graphics Forum*, 26(2), 214–226.
- Sun, Z., Xu, Y., Hoegner, L., Stilla, U., 2018. Classification of MLS point clouds in urban scenes using detrended geometric features from supervoxel-based local context. *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*, IV-2, 271–278.
- Tong, X., Ye, Z., Xu, Y., Liu, S., Li, L., Xie, H., Li, T., 2015. A novel subpixel phase correlation method using singular value decomposition and unified random sample consensus. *IEEE Transactions on Geoscience and Remote Sensing*, 53(8), 4143–4156.
- Toussaint, G. T., 1983. Solving geometric problems with the rotating calipers. *Proc. IEEE Melecon*, 83, A10.
- 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, 88–100.
- Weber, C., Hahmann, S., Hagen, H., 2010. Sharp feature detection in point clouds. *2010 Shape Modeling International Conference*, IEEE, 175–186.
- 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, 286–304.
- Xie, L., Hu, H., Zhu, Q., Wu, B., Zhang, Y., 2017. Hierarchical Regularization of Polygons for Photogrammetric Point Clouds of Oblique Images. *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 42.
- Xu, Y., Tuttas, S., Hoegner, L., Stilla, U., 2018a. Reconstruction of scaffolds from a photogrammetric point cloud of construction sites using a novel 3D local feature descriptor. *Automation in Construction*, 85.
- Xu, Y., Tuttas, S., Hoegner, L., Stilla, U., 2018b. Voxel-based segmentation of 3D point clouds from construction sites using a probabilistic connectivity model. *Pattern Recognition Letters*, 102, 67–74.
- Xu, Y., Yao, W., Hoegner, L., Stilla, U., 2017. Segmentation of building roofs from airborne LiDAR point clouds using robust voxel-based region growing. *Remote Sensing Letters*, 8(11), 1062–1071.

Xu, Y., Yao, W., Tuttas, S., Hoegner, L., Stilla, U., 2018c. Unsupervised segmentation of point clouds from buildings using hierarchical clustering based on gestalt principles. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 1–17.

Yao, W., Hinz, S., Stilla, U., 2010. Automatic vehicle extraction from airborne LiDAR data of urban areas aided by geodesic morphology. *Pattern Recognition Letters*, 31(10), 1100–1108.