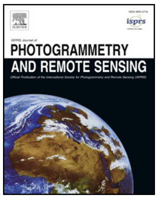
Contents lists available at ScienceDirect

ISPRS Journal of Photogrammetry and Remote Sensing

journal homepage: www.elsevier.com/locate/isprsjprs



Toward better boundary preserved supervoxel segmentation for 3D point clouds[☆]

Yangbin Lin[a], Cheng Wang[b,*], Dawei Zhai[b], Wei Li[b], Jonathan Li[c]

[a] *Computer Engineering College, Jimei University, Xiamen, China*
[b] *Fujian Key Laboratory of Sensing and Computing for Smart Cities, Department of Computer Science, Xiamen University, Xiamen FJ 361005, China*
[c] *Department of Geography & Environmental Management, University of Waterloo, Waterloo, ON N2L 3G1, Canada*

ARTICLE INFO

Keywords:
Supervoxel segmentation
Point clouds
Subset selection
Over-segmentation

ABSTRACT

Supervoxels provide a more natural and compact representation of three dimensional point clouds, and enable the operations to be performed on regions rather than on the scattered points. Many state-of-the-art supervoxel segmentation methods adopt fixed resolution for each supervoxel, and rely on the initialization of seed points. As a result, they may not preserve well the boundaries of the point cloud with a non-uniform density. In this paper, we present a simple but effective supervoxel segmentation method for point clouds, which formalizes supervoxel segmentation as a subset selection problem. We develop an heuristic algorithm that utilizes local information to efficiently solve the subset selection problem. The proposed method can produce supervoxels with adaptive resolutions, and dose not rely the selection of seed points. The method is fully tested on three publicly available point cloud segmentation benchmarks, which cover the major point cloud types. The experimental results show that compared with the state-of-the-art supervoxel segmentation methods, the supervoxels extracted using our method preserve the object boundaries and small structures more effectively, which is reflected in a higher boundary recall and lower under-segmentation error.

1. Introduction

As with superpixels in 2D image processing, the use of supervoxels greatly reduces the number of points. This is beneficial to applications that are time consuming with the original 3D points. Moreover, supervoxels provide a more natural and compact representation for 3D points, which enables operations (such as feature computing) to be performed on regions rather than on scattered points. For these reasons, supervoxels have become increasingly popular in many 3D remote sensing applications, such as object detection (Guan et al., 2016; Wang et al., 2015), semantic labeling (Luo et al., 2016), and saliency detection (Yun and Sim, 2016).

Here, we define the supervoxel as a compact point cluster, which is slightly different from the one in Papon et al. (2013). The general desirable properties of superpixel segmentation are also suitable for supervoxel segmentation. First, supervoxel segmentation should preserve object boundaries, and each supervoxel should overlap with only one object. Second, supervoxel segmentation must be efficient, and at least should not reduce the achievable performance of an application that is dependent on it. Third, each supervoxel should have a regular shape, which is convenient for subsequent applications.

Many state-of-the-art supervoxel segmentation methods adopt fixed resolution for each supervoxel, and rely on initialization of seed points. As a result, they may not preserve well the boundaries of the point cloud with a non-uniform density. In this paper, we formalize supervoxel segmentation as a subset selection problem, and present a simple but effective method to solve the problem. The major advantage of our method is that it adopts an adaptive resolution for each supervoxel and can preserve object boundaries more effectively than existing methods. An example is presented in Fig. 1, the supervoxels extracted by the proposed method better adhere to the ground-truth boundaries, even for road curbs with slight height differences (see the red[1] boxes in Fig. 1(c)).

The main contributions of this paper are as follows:

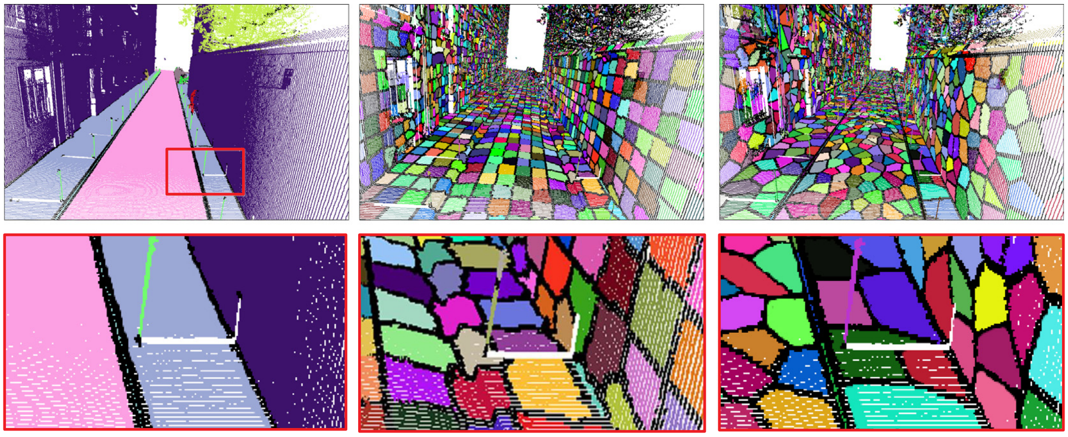
First, we formalize supervoxel segmentation as a subset selection problem. Our formalization involves an explicit energy function, which can be optimized directly. Second, in order to minimize the energy function for subset selection, we propose a simple but effective method that does not require seed points initialization and does not contain internal parameters. Finally, the proposed method significantly outperforms the state-of-the-art supervoxel methods with respect to

[☆] This work was supported by the National Science Foundation of China (Project No. 61701191, 41471379, and U1605254).

[*] Corresponding author.
E-mail address: cwang@xmu.edu.cn (C. Wang).

[1] For interpretation of color in 'Fig. 1', the reader is referred to the web version of this article.

https://doi.org/10.1016/j.isprsjprs.2018.05.004
Received 14 July 2017; Received in revised form 24 April 2018; Accepted 8 May 2018



(a) Point cloud with the groud-truth labels (b) VCCS (c) Proposed

Fig. 1. Comparison results for VCCS (Papon et al., 2013) and the proposed method. As emphasized in red boxes, the proposed method can preserve the boundaries more effectively.

boundary recall and under-segmentation error metrics on three publicly available point cloud segmentation benchmarks.

2. Related work

Unlike superpixel segmentation, which is already a well studied topic in image processing (Moore et al., 2008; Veksler et al., 2010; Liu et al., 2011; Bergh et al., 2013), supervoxel segmentation remains in the development stage.

In video and 3D image segmentation, a supervoxel is usually defined as a stack of 2D image regions (Moore et al., 2008; Xu and Corso, 2012; Zhou et al., 2015). In this case, many superpixel segmentation methods can be directly extended to compute supervoxels. Moore et al. (2008) presented a graph-based superpixel segmentation method, where superpixels are iteratively partitioned from a 2D grid graph by horizontal and vertical cutting. The authors also extended this method to the computation of supervoxels on a 3D grid graph, which can be employed for video over-segmentation. Veksler et al. (2010) presented a framework for both superpixel and supervoxel segmentation. They formulated the superpixel or supervoxel segmentation problem as an energy minimization problem, and solved it using graph cut. Weikersdorfer et al. (2012) transferred superpixels to 3D space, by taking into account depth information for RGB-D image over-segmentation. This method has been further extended to RGB-D video over-segmentation. Zhou et al. (2015) proposed a multiscale superpixel and supervoxel algorithm using hierarchical edge weighted centroidal Voronoi tessellation. Here, superpixels or supervoxels in higher levels are clustered from superpixels or supervoxels in lower levels. In addition to video segmentation, Picciau et al. (2015) develop an adaptation of the SLIC superpixel algorithm (Achanta et al., 2012) for tetrahedral mesh over-segmentation.

However, the methods above are designed for regular data, where the primitives are uniformly distributed. More related to the proposed method is the VCCS algorithm developed in Papon et al. (2013). It first voxelizes the point cloud using octree, and then extracts the initial supervoxels by evenly partitioning the 3D space. These initial supervoxels are then grown using the local k-means clustering method (Achanta et al., 2012). VCCS is reported to be highly efficient, and achieves reasonably good results on RGB-D test data. However, the results of VCCS depends on the setting of voxel resolution. For the point cloud with non-uniform density (typically acquired by the current laser devices), more than one object can overlap with the same voxel. In this case, VCCS may not preserve well the object boundaries.

To make supervoxels conform better to object boundaries, Song et al. (2014) present a boundary-enhanced supervoxel segmentation (BESS) method. The method has two stages. In the first stage, it detects the boundary points by estimating the discontinuity of consecutive points along the scan line. In the second stage, it constructs a neighborhood graph that excludes the edges connected by boundary points, and then performs a clustering process on the graph to segment the point clouds into supervoxels. Although BESS can be used for outdoor scene data with the non-uniform density, it depends on the assumption that the points are sequentially ordered in one direction. This assumption greatly reduces the practicality of BESS method to general point cloud data.

3. Problem formulation

Given a point set $\mathscr{P} = \{p_1,\ldots,p_N\}$ with N points, the partitioning of $\mathscr{P}$ into K supervoxels $\mathscr{S} = \{S_1,\ldots,S_K\}$ can be regarded as a mapping from each point to a label of a supervoxel, i.e.,

$$s\colon \{p_1,\ldots,p_N\} \to \{1,\ldots,K\}, \tag{1}$$

where $s(p)$ represents the label of the supervoxel to which the point p belongs. In addition, the supervoxel S_k is defined as a set of points whose label is equal to k:

$$S_k = \{p|s(p) = k\}. \tag{2}$$

Note that any mapping in the form of Eq. (1) can result in a partition with no more than K supervoxels. Furthermore, the total number of different possible partitioning solutions is $\frac{K^N}{K!}$ (Bergh et al., 2013), which is an extremely large number. In order to reduce the solution space, we consider a representative point $r_i \in S$ for each supervoxel S. Assuming that we have already obtained K representative points $\{r_1,\ldots,r_K\}$, the mapping function s can easily be computed according to the following equation:

$$s(p) = \arg\min_i D(p,r_i). \tag{3}$$

where D is a distance metric to measure the dissimilarity between two points. Therefore, the problem of seeking a partitioning is transformed into the problem of selecting K representative points from N original points, which is known as the subset selection problem (Elhamifar et al., 2016; Tropp, 2008). Then, the solution space is reduced from $\frac{K^N}{K!}$ to $\binom{N}{K}$. Note that because $K \ll N$, $\binom{N}{K} \ll \frac{K^N}{K!}$.

The subset selection problem can be encoded as an optimization problem on unknown binary variables $z_{ij} \in \{0,1\}$. Here, $z_{ij} = 1$ has two meanings: first that p_i is a representative point, and second that p_j is a non-representative point that is represented by p_i. Therefore, the definition of a supervoxel in Eq. (2) can be rewritten as:

$$S_i = \{p_j | z_{ij} = 1\}. \tag{4}$$

To ensure that each point p_j is represented by exactly one supervoxel, we set $\sum_{i=1}^{N} z_{ij} = 1$. Our aim is to determine K representative points to minimize the sum of the dissimilarity distances between each point and its representative point, which can be formalized as follows:

$$\begin{aligned} &\min_{\{z_{ij}\}} \sum_{i=1}^{N}\sum_{j=1}^{N} z_{ij} D(p_i,p_j) \\ &\text{s. t.}\quad z_{ij} = \{0,1\}\ ,\forall\ i,j\ ;\quad \sum_{i=1}^{N} z_{ij} = 1\ ,\quad \forall\ j;\quad C(Z) = K \end{aligned} \tag{5}$$

Here, the function $C(\cdot)$ is used to count the number of representative points, and is defined as follows:

$$C(Z) = \sum_{i=1}^{N} I\left(\sum_{j=1}^{N} z_{ij}\right), \tag{6}$$

where $I(.)$ is the indicator function:

$$I(x) = \begin{cases} 0, & \text{if } x = 0; \\ 1, & \text{Otherwise.} \end{cases} \tag{7}$$

4. Optimization

Since subset selection problem is NP-hard, most methods rely on approximation strategies. For example, the state-of-the-art method, DS3 (Elhamifar et al., 2016), adopts a convex relaxation of the problem and achieves fairly good results in terms of solution quality and speed. Although DS3 has a time complexity of $O(N\log(N)K)$, it is still too slow for large-scale point cloud data with a large value of K. On the other hand, some faster approximation strategies, such as local K-means (Achanta et al., 2012; Papon et al., 2013), rely on the initialization of seed supervoxel points. In Fig. 2, we consider a point set located on a plane with a small protrusion. It can be seen that different seed point selections can result in different partitions. Furthermore, it is difficult to preserve the small structures unless they are covered by some seed points.

Here, we present an efficient energy descent method that takes advantage of local information, and does not require the initialization of seed points. First, we consider the following relaxation of Eq. (5), which takes the form of an energy function:

$$\begin{aligned} &\min\quad E(Z) = \sum_{i=1}^{N}\sum_{j=1}^{N} z_{ij} D(p_i,p_j) + \lambda|C(Z) - K| \\ &\text{s.t.}\quad z_{ij} = \{0,1\}\ ,\quad \forall\ i,j\ ;\quad \sum_{i=1}^{N} z_{ij} = 1\ ,\quad \forall\ j. \end{aligned} \tag{8}$$

Here, the first term ensures that the selected representative points can effectively approximate the collection of all points, and the second term constrains the number of representative points to be close to K. $\lambda > 0$ is a regularization parameter used to set the trade-off between two terms. A large value of λ results in a smaller deviation of the number of clusters from K, but may sacrifice the quality of these representative points.

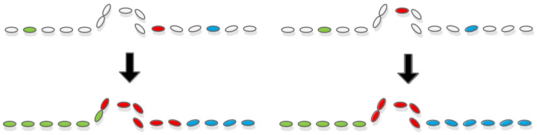


Fig. 2. For subset selection methods that rely on seed points, different seed point selections may result in different partitions (see the colored points in the upper part of the figure). Furthermore, the small structures are difficult to preserve completely unless they are covered by seed points. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

4.1. Fusion based minimization

Inspired by Nguyen and Brown (2015) and Xu et al. (2011), we adopt an adaptive strategy to automatically evaluate the value of λ. First, we set λ to a small value λ_0, and we solve Eq. (8) to obtain the initial representative points. Then, in each step we increase the value of λ and update the representative points by solving Eq. (8) with the new λ value.

Our insight is that at the beginning, the dissimilarity distance term is preferentially considered to ensure that the supervoxels do not cross the boundary. In this way, the boundaries between supervoxels will be the superset of ground-truth boundaries. With the increase of λ, the number of representative points will towards to K.

According to the adaptive strategy described above, we need to solve Eq. (8) several times with different values of λ. Thus, a fast optimizing algorithm is required. Considering that supervoxel is a local region, we can use local information to accelerate the optimization algorithm. Moreover, we expect to build a hierarchy, so that the boundaries obtained in the previous iteration can be preserved. Here, we present an energy minimization method based on a bottom-up fusion strategy. As illustrated in Fig. 3, suppose that there are two adjacent supervoxels, S_i and S_j, we denote by D'_{ji} the total dissimilarity distance from S_j to S_i, i.e.,

$$D'_{ji} = \sum_{p \in S_j} D(p,r_i), \tag{9}$$

where r_i is the representative point of S_i. Then, we consider merging the supervoxel S_j into the supervoxel S_i, the expected energy reduction can be expressed as:

$$\Delta = \lambda + (D'_{jj} - D'_{ji}). \tag{10}$$

The first term of Eq. (10) results from the fact that the number of representative points decreases by one after merging, and the second term represents the difference in the dissimilarity distance before and after the merging operation is performed. We only accept the merging operation when $\Delta > 0$.

To compute D' in Eq. (10), a time complexity of $O(|S_i| + |S_j|)$ is required, which is too high. Here, we assume that the dissimilarity D is a metric, and thus satisfies the triangle inequality. Therefore, we have that

$$D(p,r_i) \leqslant D(p,r_j) + D(r_j,r_i), \tag{11}$$

and can further deduce that

$$D'_{ji} = \sum_{p \in S_j} D(p,r_i) \leqslant \sum_{p \in S_j} (D(p,r_j) + D(r_j,r_i)) = D'_{jj} + c_j D(r_j,r_i), \tag{12}$$

where c_j is the number of points in S_j. With the simultaneous Eqs. (10) and (12), we have:

$$\Delta \geqslant \lambda - c_j D(r_j,r_i) \tag{13}$$

Now, we can determine whether a merging operation can reduce the energy function $E(Z)$ as follows:

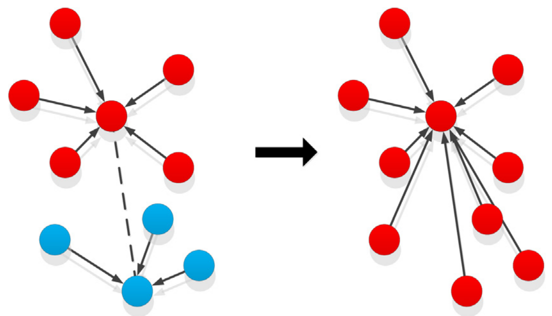


Fig. 3. An example of the merging operation for two adjacent representative points.

$$\Delta' = \lambda - c_j D(r_j, r_i) > 0 \tag{14}$$

which can be computed in $O(1)$ time.

In Eq. (8), we introduce an regularization parameter λ to set the trade-off between two terms. However, more parameters means more adjustments and less robustness to the method. Therefore, we try to eliminate this parameter. First, the initial value of λ can be set to the median of the lowest dissimilarity distances between each point and its neighboring points, i.e.,

$$\lambda_0 = \text{median}\{\min_p(D(p, \mathscr{N}_i)) | p \in \mathscr{P}\} \tag{15}$$

Thus, according to Eq. (14), the number of representative points will be reduced by half in the first iteration. Second, we do not need to set the maximal value of λ, but just terminate the iteration when the number of representative points decrease to K. Third, in each iteration we double the value of λ, so that the iteration will converge quickly.

Algorithm 1. Fusion based Minimization

```
Input: A point set, $\mathscr{P}$; the neighboring set, $\mathscr{N}$; the dissimilarity
  metric, $D$; and the expected number of supervoxels, $K$.
Output: $K$ supervoxels $\{S_1,...,S_K\}$ representing a partition of $\mathscr{P}$.
Initialize $\lambda_0$ as Eq. (15)
$\lambda \leftarrow \lambda_0; G \leftarrow \mathscr{N}; \mathscr{R} \leftarrow \mathscr{P}; c_i \leftarrow 1$
repeat
  for all $r_i \in \mathscr{R}$ do
    forall $r_j \in G_i \cap \mathscr{R}$do
      if $\lambda - c_j D(r_j, r_i) > 0$ then
        Merge $r_j$ into $r_i$
        $G_i \leftarrow G_i \cup G_j$
        $R \leftarrow R - \{r_j\}$
        $c_i \leftarrow c_i + 1$
      end if
    end for
    end for
    $\lambda \leftarrow 2\lambda$
until $|\mathscr{R}| = K$
return $\{S_1,...,S_k\}$ according to Eq. (4)
```

More details are described in Algorithm 1. In the input list, the neighboring set $\mathscr{N}_i$ is used to determine the adjacent points of $p_i \in \mathscr{P}$. In practice, $\mathscr{N}_i$ can be defined by k-Nearest-Neighbors of p_i. Furthermore, the dissimilarity metric D should satisfy the triangle inequality.

The intuition behind the optimization is that it preferentially aggregates the points located in the smooth areas. This encourages the supervoxels to avoid crossing the boundaries. Other heuristic operations (such as splitting one supervoxels into two supervoxels) may also be effective, but we consider only merging operation to maintain computational complexity.

4.2. Exchange based minimization

After K representative points have been determined by aggregation, we can continue to optimize the energy function $E(Z)$ by assigning each non-representative point to the representative point with the lowest dissimilarity distance from it. Specifically, given a pair of adjacent points p_i and p_j, if these satisfy $D(p_i, r_j) < D(p_i, r_i)$, where r_i and r_j are the representative points of p_i and p_j, respectively, then the energy function $E(Z)$ can be reduced by assigning p_i to r_j. We continue to reduce the energy function until no further improvement can be achieved.

The example shown in Fig. 4 demonstrates that exchange based minimization is useful for obtaining better supervoxel boundaries and more regular shapes. More details are described in Algorithm 2.

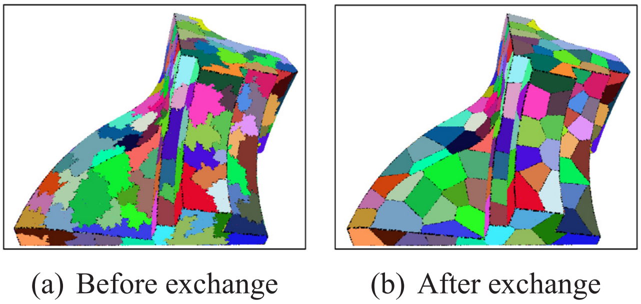

(a) Before exchange (b) After exchange

Fig. 4. Result of exchange based minimization.

Algorithm 2. Exchange based Minimization

```
Initialize a queue, $Q$, for points, i.e., $Q \leftarrow \mathscr{P}$
while $Q \neq \emptyset$ do
  Remove the front point $p_i$ from $Q$
  for all $p_j \in \mathscr{N}_i$ do
    if $D(p_i, r_j) < D(p_i, r_i)$ then
      Assign $p_i$ to $r_j$
      if $p_i \notin Q$ then
        Add $p_i$ to the back of $Q$.
      end if
    end if
  end for
end while
```

Note that, the exchange operation here is similar to the re-assignment step in k-means or k-medoids. And, the energy in Eq. (8) can be further reduced by applying k-medoids update step. That is, for each supervoxel, we swap its representative point and non-representative points to reduce the energy. Fig. 5 shows the improved results by performing full k-medoids-style iterations. We noticed that in the first few iterations, the energy is significantly reduced. And then, it tends to be flat. However, each of the k-medoids iterations requires $O(N^2/K)$ time complexity (N is the number of points and K is the number of supervoxels), which is too slow for large-scale point clouds. Therefore, we only consider the exchange of the supervoxel's boundary points without considering k-mediods method to change the representative points.

4.3. Time complexity

In the fusion phase, the number of representative points is decreased by almost half at each iteration. Thus, the number of iterations can be estimated as $\log(N/K)$. In each iteration, the number of merging operations is equal to the cardinality of G, which is no greater than M.

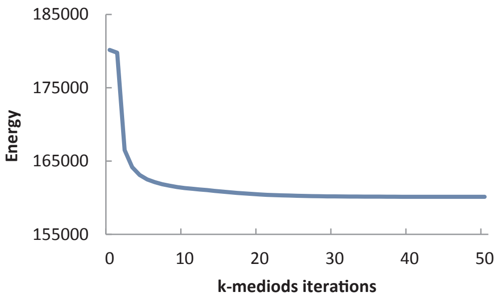


Fig. 5. This plot shows the energy in Eq. (8) for each extra k-medoids-style iteration. The input point cloud is shown in Fig. 4. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Here, M is the total number of neighbor points in $\mathcal{N}$. Thus, the merging phase can be performed in $O(M\log(N/K))$. Because the size of G is decreased after performing a merging operation, this time complexity is not a tight upper-bound.

In the exchange phase, for each point p_i there are at most N_i different adjacent representative points. It is only necessary to compare the dissimilarity distance $O(N_i)$ times. Therefore, the total number of comparisons is $O(M)$.

As a result, the proposed algorithm can be performed in a computational time of $O(M\log(N/K))$. If we adopt the k-nearest neighbors for each point, then M is equal to kN, and the time complexity can be rewritten as $O(kN\log(N/K))$. Usually, k is a small constant number, so this time complexity is acceptable in practice.

5. Experiments

The proposed method was coded in C++, and run on one core of an Intel Core i7-6500U 2.50 GHz CPU, with 8 GB memory, on a Linux Ubuntu 16.04 operating system.

For comparison, we implemented the VCCS method according to the public source code in PCL (Rusu and Cousins, 2011).[2] Because VCCS is based on voxels, for a more equitable comparison we modified VCCS so that it can be performed directly on points. We refer to the modified version of VCSS, in which voxel based neighborhoods are replaced by k-Nearest Neighbors, as VCCS_kNN.

5.1. Parameter settings

Only two user-specified parameters are needed for our method: the expected number of supervoxels, K, and the distance metric, D, used to evaluate the dissimilarity between two points.

In practice, the value of K can be evaluated by desired resolution of supervoxels, R. Because the resolution of supervoxels has a more definite geometric meaning than K. Fig. 6 gives an example of the supervoxel segmentation results for different R. As emphasized in black boxes, even if the resolution of structures is smaller than R, the proposed method can preserve the boundaries. In the later experiments, we will quantitatively analyze the proposed method for different values of R.

For distance metric, D, we adopt a measure similar to VCCS:

$$D(p,q) = 1-|n_p\cdot n_q| + 0.4\frac{\|p-q\|}{R}, \tag{16}$$

where n_p and n_q are the normal vectors of p and q, respectively. Because color information of LiDAR point cloud data is usually not available, we do not consider the color distance between points. Beside that, Eq. (16) is the same as the VCCS implementation in Rusu and Cousins (2011).

In addition, the voxel resolution of VCCS method was set to 0.1 m, and the number of nearest neighbors for VCCS_kNN and the proposed method was set to 20. These parameters setting for VCCS and VCCS_kNN have been fine tunned to obtain the best results.

5.2. Evaluation metrics

As previously mentioned, supervoxels should preserve, and not cross, object boundaries. To quantitatively evaluate these abilities of supervoxel segmentation methods, we adopt three standard metrics: boundary recall, under-segmentation error, and Martin error.

5.2.1. Boundary recall (BR)

Boundary recall measures the percentage of ground-truth boundaries that are covered by supervoxel boundaries. We adopt the same definition of BR as given in Liu et al. (2011):

$$BR_{\mathcal{G}}(S) = \frac{\sum_{p\in\delta\mathcal{G}} \mathbb{I}(\min_{q\in\delta\mathcal{S}}(\|p-q\| < \epsilon))}{|\delta\mathcal{G}|}, \tag{17}$$

where $\delta\mathcal{S}$ and $\delta\mathcal{G}$ denote supervoxel boundaries and ground-truth boundaries, respectively, and $\mathbb{I}$ is an indicator function to check whether a ground-truth boundary point is covered by supervoxel boundaries. In our implementation, we denote a point p as a boundary point if there exists a k-nearest neighbor point of p whose label is different from that of p. The value of ϵ is set to 0.01 m for indoor scene benchmarks, and 0.03 m for outdoor scene benchmarks.

5.2.2. Under-segmentation error (UE)

Under-segmentation error is another important metric for measuring the amount of leakage of supervoxels across the ground-truth boundaries (Papon et al., 2013). It is defined as:

$$UE_{\mathcal{G}}(S) = \frac{1}{N}\left(\sum_{i=1}^{M}\left(\sum_{S_j|S_j\cap\mathcal{G}_i\neq\emptyset}|S_j|\right)-N\right), \tag{18}$$

where $\mathcal{G}_1,\ldots,\mathcal{G}_M$ are the regions of ground-truth segmentation, and N is the total number of labeled points in $\mathcal{G}$. For each segmented region $\mathcal{G}_i$, we determine the overlapping supervoxel set $\{S_j, S_j\cap\mathcal{G}_i\neq\emptyset\}$ that covers $\mathcal{G}_i$. Then, we count the number of points that leak out the object boundaries, and normalize this by N. A low UE value indicates that the supervoxels do not tend to cross the boundaries.

5.2.3. Global consistency error (GCE)

We also adopt an object-level metric, Global Consistency Error (GCE) (Martin et al., 2001), to simultaneously evaluate both over-segmentation and under-segmentation errors. Martin et al. (2001) define the error between ground truth annotation $\mathcal{G}_i$ and supervoxel S_j as

$$P_{ij} = \frac{|\mathcal{G}_i\backslash S_j|}{|\mathcal{G}_i|}\times|\mathcal{G}_i\cap S_j| = \left(1-\frac{|\mathcal{G}_i\cap S_j|}{|\mathcal{G}_i|}\right)\times|\mathcal{G}_i\cap S_j|. \tag{19}$$

Similarity, the error between supervoxel S_i and ground truth annotation $\mathcal{G}_j$ is defined as

$$Q_{ij} = \frac{|S_i\backslash\mathcal{G}_j|}{|S_i|}\times|S_i\cap\mathcal{G}_j| = \left(1-\frac{|S_i\cap\mathcal{G}_j|}{|S_i|}\right)\times|S_i\cap\mathcal{G}_j|. \tag{20}$$

And the total intersection between ground truth and supervoxels is computed by

$$n = \sum_{i}^{M}\sum_{j}^{N}|\mathcal{G}_i\cap S_j| \tag{21}$$

Then, GCE is defined as

$$GCE(S) = \frac{1}{n}\min\left\{\sum_{i}^{M}\sum_{j}^{N}P_{ij},\sum_{i}^{N}\sum_{j}^{M}Q_{ij}\right\} \tag{22}$$

The range of GCE is [0,1], where 0 indicates no error and 1 worst segmentation. In most cases, the resolution of supervoxel is smaller than the resolution of the object. Therefore, GCE mainly penalizes over-segmentation here.

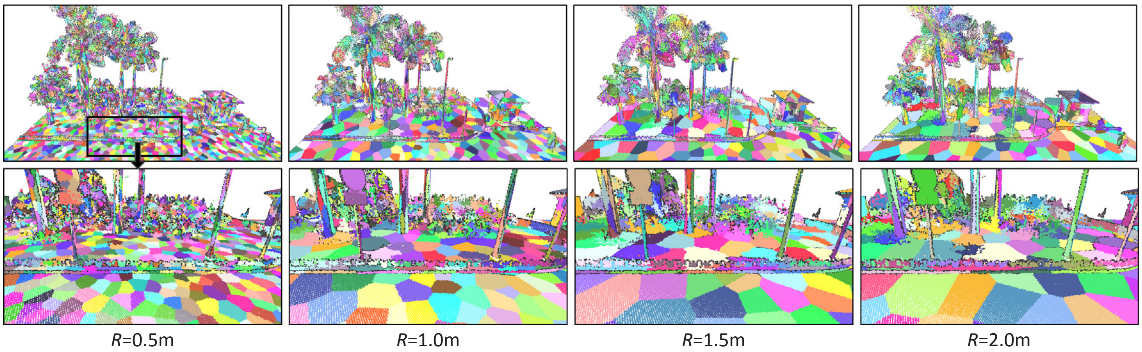
5.3. Indoor scene performance

For indoor scenes, we adopted NYU Depth Dataset V2 (Silberman et al., 2012).[3] which contains 1449 labeled RGBD images. Here, we converted the RGBD images into 3D point cloud data. Each RGBD image of NYU data has a resolution of 640 × 480, thus the number of 3D points for each data is nearly 307,200.

Indoor point clouds usually have lower accuracy which results in inaccurate normal vectors. Therefore, the BR and UE values of three

[2] http://www.pointclouds.org/.

[3] http://www.cs.nyu.edu/silberman/datasets/nyu_depth_v2.html.



R=0.5m *R*=1.0m *R*=1.5m *R*=2.0m

Fig. 6. Supervoxel segmentation results for different resolution of supervoxels.

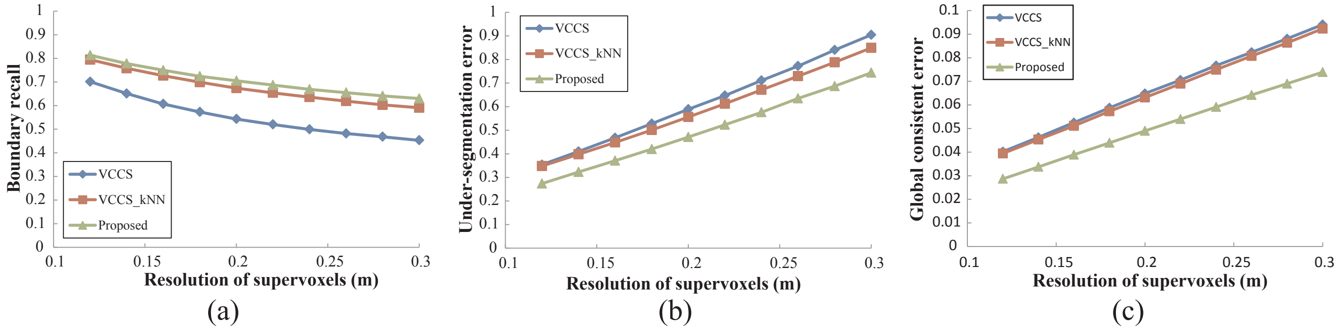
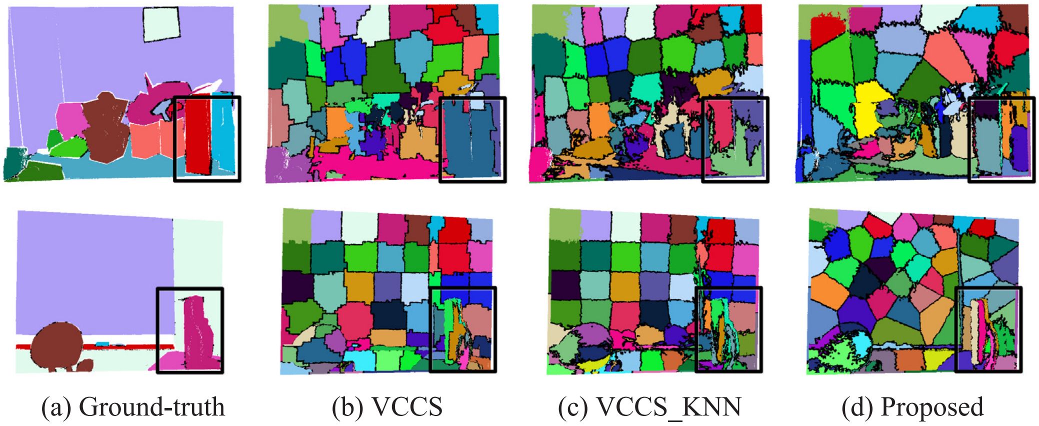


Fig. 7. Quantitative evaluation of three methods on NYU RGBD benchmark.



(a) Ground-truth (b) VCCS (c) VCCS_KNN (d) Proposed

Fig. 8. Visual comparison of supervoxel segmentation results on NYU RGBD Dataset.

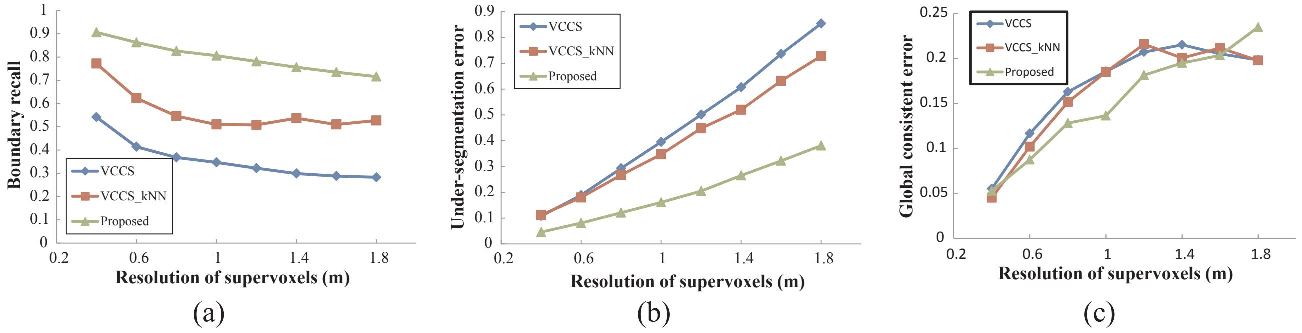


Fig. 9. Quantitative evaluation of three methods on IQTM benchmark.

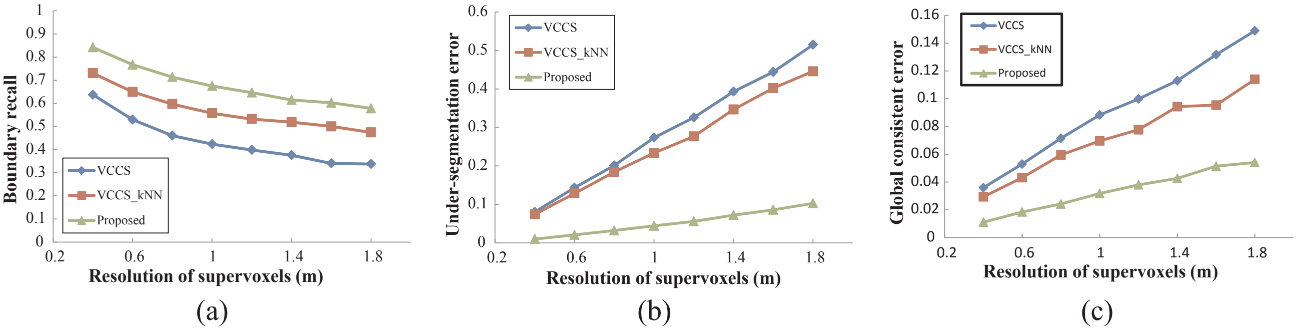


Fig. 10. Quantitative evaluation of three methods on Semantic3D benchmark.

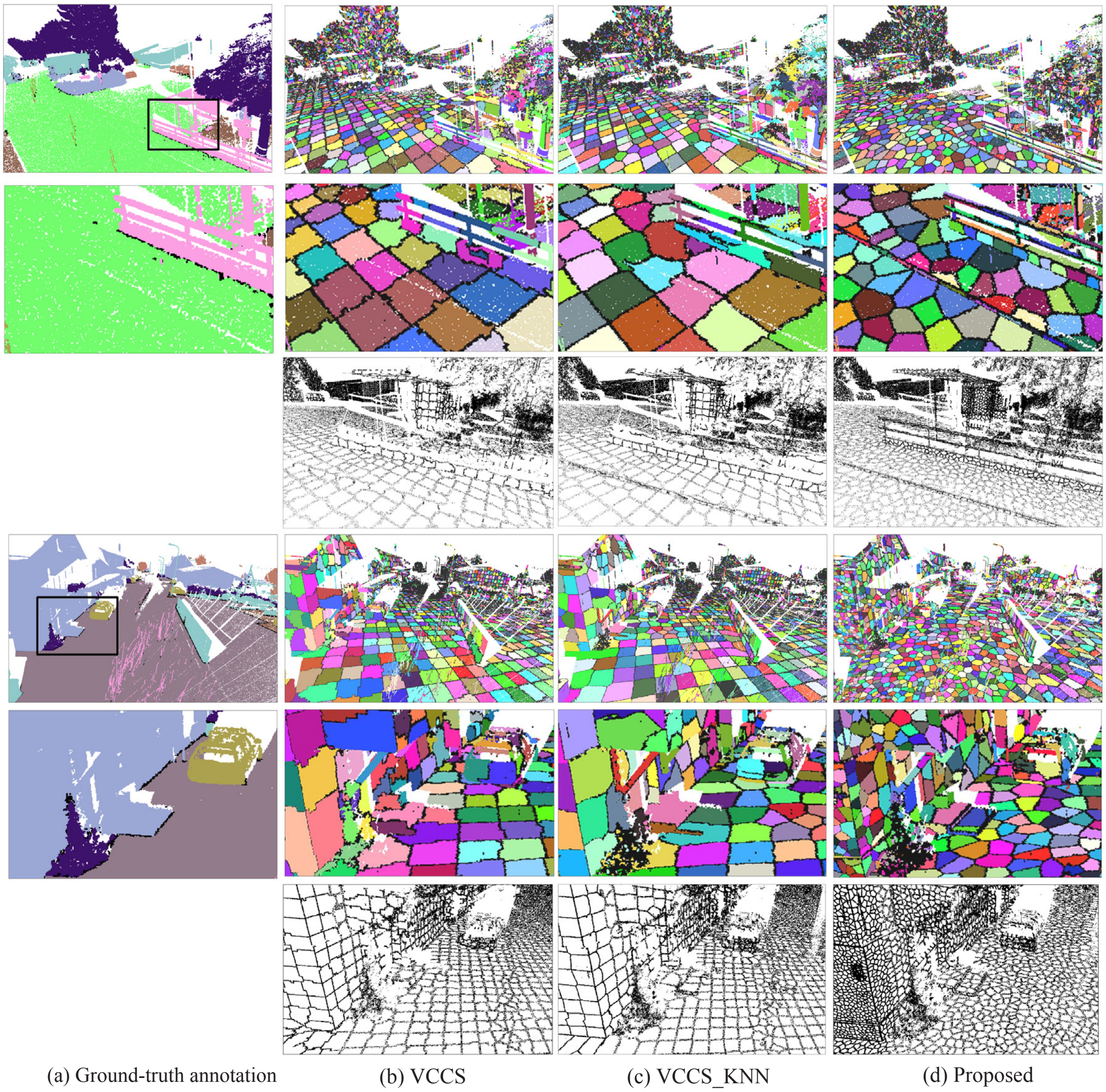


Fig. 11. Visual comparison of supervoxel segmentation results on Semantic3D benchmark.

methods are close (see Fig. 7(a and b)). Despite that, the proposed method achieves a best BR and UE values for all supervoxel resolutions. The comparison results of GCE (Fig. 7(c)) also show that our method outperforms VCCS and VCCS_kNN.

The typical results of three methods are shown in Fig. 8. We can observe that only the proposed method can separate the objects emphasized in the black boxes.

5.4. Outdoor scene performance

For outdoor scenes, we selected two benchmarks: IQmulus & TerraMobilita (IQTM) (Vallet et al., 2015)[4] and Semantic3D.[5]

IQTM benchmark is a mobile laser scanning (MLS) point cloud data

[4] http://data.ign.fr/benchmarks/UrbanAnalysis/index.html.

[5] http://www.semantic3d.net/.

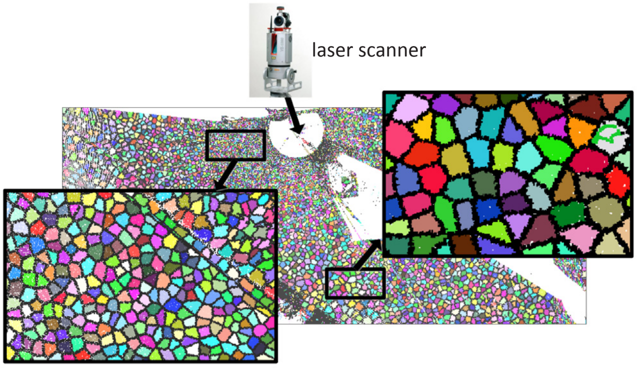


Fig. 12. For the point cloud with non-uniform density, supervoxels obtained by the proposed method have adaptive resolution.

for 3D city analysis. It involves 12 million manually labeled points, which cover a 200-m street in Paris (France). A typical scene of the IQTM benchmark is presented in Fig. 1. Compared with the indoor point clouds, MLS data usually has a larger volume, and highly non-uniform density. As shown in Fig. 9, the proposed method has a significant advantages for these point clouds with non-uniform density, which results in higher boundary recall and lower under-segmentation error. For GCE metric, the proposed method obtains the best results when the resolution of supervoxels smaller than 1.8 m (Fig. 7(c)).

Semantic3D is a large-scale point cloud classification benchmark. It contains 15 manually labeled point cloud data, which consists of more than 1 billion points. In addition, the Semantic3D benchmark is acquired by a static laser scanner, and therefore has a higher point density than IQTM benchmark. Due to insufficient memory, we down-sampled Semantic3D data, so that each point cloud data contains only 10 million points.

As shown in Fig. 10, similar to the results on IQTM benchmark, the proposed method achieves the best BR, UE, and GCE values at all resolution setting. Some typical results on Semantic3D benchmark are shown in Fig. 11. We also extract the boundaries of supervoxels to facilitate visual inspection. Intuitively, the proposed method achieves the best perceptually satisfactory segmentation results.

It should be emphasized that in all experiments, the three methods obtained the same number of supervoxels. The reason why the number of supervoxels in Fig. 10(d) looks more is that the supervoxels obtained by the proposed method have an adaptive resolution. In contrast, the size of supervoxels obtained by VCCS is fixed. An example is shown in Fig. 12, due to the distance from the laser scanner, the point cloud has a non-uniform density. Usually, the supervoxels obtained by the proposed method have smaller resolution in regions with higher point cloud densities, and larger resolution at sparse area.

Compared to fixed resolution, adaptive resolution has obvious advantage. First, in dense area, point clouds have a higher accuracy (because the distance to laser scanner is smaller), and therefore have more complete details. Supervoxels with smaller resolution better preserve these details. Second, not many supervoxels are needed in sparse area. A larger resolution of supervoxels is more reasonable. The proposed method can automatically obtain supervoxels with adaptive resolution, and thus obtains the better results.

5.5. *Time performance*

Time performance of three methods is shown in Figs. 13. The proposed method is slower than VCCS and VCSS_kNN. Since VCCS is performed on voxels, it greatly reduces the size of problem. Compared with VCCS_kNN, which directly performed on original points, the proposed method requires at most twice the amount of time. Taking into account the benefits of sacrificing speed, we believe that this efficiency is reasonable. Moreover, the proposed method has the advantage when multiple resolutions are considered. In this case, the proposed method can be run only once to build a multi-resolution hierarchy. In contrast, VCCS need to run several times under the different resolution.

5.6. *Discussion*

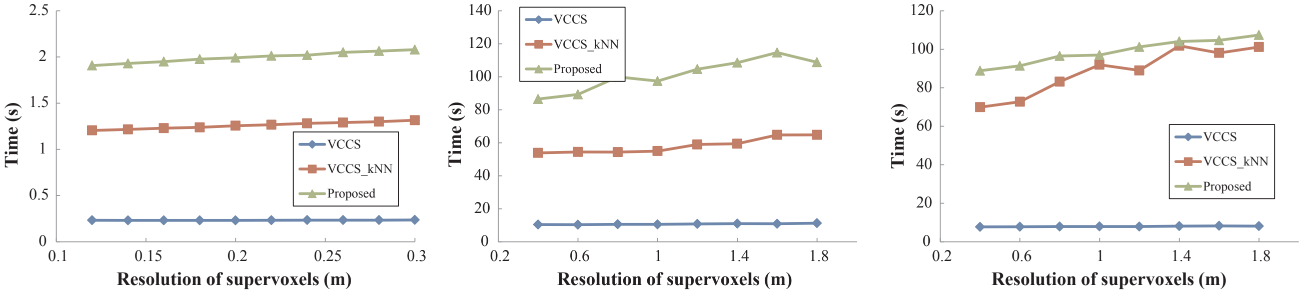
Because the three methods adopt the same dissimilarity metric in the experiments, the results are mainly determined by the optimization methods. VCCS obtains the initial seed points by uniformly partitioning the 3D space, and adopts a local K-means method to extract the supervoxels on the basis of initial seed points. This has the advantage that each supervoxel has a similar resolution, but has the disadvantage of ignoring structures that are smaller than the supervoxel resolution. For the outdoor point clouds, which typically have non-uniform density, this disadvantage is more pronounced (see Figs. 9 and 10).

In contrast, the proposed method does not need to initialize seed points. It extracts the supervoxels by directly minimizing the energy function, without the need to limit the resolution of the supervoxels. Thus, the supervoxels extracted by the proposed method have variable size, which can better preserve small structures. As a result, we obtain a better boundary recall value and under-segmentation error than VCCS and VCCS_kNN, especially for point clouds with non-uniform density.

6. Conclusion

In this paper, we have formalized the supervoxel segmentation problem as a subset selection problem. By utilizing local information for each point, we have presented a heuristic method for efficiently optimizing this problem. Our method does not require the initialization of seed points, and has a theoretical time complexity of $O(M\log(N/K))$. Moreover, our method only relies on user-defined parameters, namely the dissimilarity metric D and the expected number of supervoxels K. It does not contain any internal parameters.

Our method was tested on three publicly available point cloud segmentation benchmarks, and a quantitative analysis was provided. Compared with the state-of-the-art supervoxel segmentation methods



(a) NYU (b) IQTM (c) Semantic3D

Fig. 13. Running time of three methods on NYU, IQTM, and Semantic3D benchmarks, respectively.

PCLV, VCCS, and its modified version VCCS_kNN, our method can preserve object boundaries and small structures more effectively, which was reflected in higher boundary recall values and lower under-segmentation error.

Acknowledgments

This work was supported by National Natural Science Foundation of China under project 61701191, 41471379, and U1605254). The authors would like to thank the anonymous reviewers for their valuable comments.

References

Achanta, R., Shaji, A., Smith, K., Lucchi, A., Fua, P., SuSstrunk, S., 2012. Slic superpixels compared to state-of-the-art superpixel methods. IEEE Trans. Pattern Anal. Mach. Intell. 34 (11), 2274–2282.

Bergh, M.V.D., Boix, X., Roig, G., Capitani, B.D., Gool, L.V., 2013. Seeds: superpixels extracted via energy-driven sampling. Int. J. Comput. Vis. 111 (3), 298–314.

Elhamifar, E., Sapiro, G., Sastry, S.S., 2016. Dissimilarity-based sparse subset selection. IEEE Trans. Pattern Anal. Mach. Intell. 38 (11), 2182–2197.

Guan, H., Yu, Y., Li, J., Liu, P., 2016. Pole-like road object detection in mobile lidar data via supervoxel and bag-of-contextual-visual-words representation. IEEE Geosci. Rem. Sens. Lett. 13 (4), 520–524.

Liu, M.Y., Tuzel, O., Ramalingam, S., Chellappa, R., 2011. Entropy rate superpixel segmentation. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 2097–2104.

Luo, H., Wang, C., Wen, C., Cai, Z., Chen, Z., Wang, H., Yu, Y., Li, J., 2016. Patch-based semantic labeling of road scene using colorized mobile lidar point clouds. IEEE Trans. Intell. Transport. Syst. 17 (5), 1286–1297.

Martin, D., Fowlkes, C., Tal, D., Malik, J., 2001. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In: Proc. of ICCV, vol. 2, pp. 416–423.

Moore, A.P., Prince, S.J.D., Warrell, J., Mohammed, U., Jones, G., 2008. Superpixel lattices. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–8.

Nguyen, R.M.H., Brown, M.S., Dec. 2015. Fast and effective L0 gradient minimization by region fusion. In: Proc. of ICCV, pp. 208–216.

Papon, J., Abramov, A., Schoeler, M., Worgotter, F., 2013. Voxel cloud connectivity segmentation – supervoxels for point clouds. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 2027–2034.

Picciau, G., Simari, P., Iuricich, F., Floriani, L.D., 2015. Supertetras: a superpixel analog for tetrahedral mesh oversegmentation. In: International Conference on Image Analysis and Processing, pp. 375–386.

Rusu, R.B., Cousins, S., 2011. 3D is here: Point cloud library (PCL). In: IEEE International Conference on Robotics and Automation, pp. 1–4.

Silberman, N., Hoiem, D., Kohli, P., Fergus, R., 2012. Indoor segmentation and support inference from rgbd images. In: European Conference on Computer Vision, pp. 746–760.

Song, S., Lee, H., Jo, S., 2014. Boundary-enhanced supervoxel segmentation for sparse outdoor lidar data. Electron. Lett. 50 (25), 1917–1919.

Tropp, J.A., 2008. Column subset selection, matrix factorization, and eigenvalue optimization. In: ACM-SIAM Symposium on Discrete Algorithms (SODA), pp. 978–986.

Vallet, B., Brdif, M., Serna, A., Marcotegui, B., Paparoditis, N., 2015. TerraMobilita/IQmulus urban point cloud analysis benchmark. Comput. Graph. 49, 126–133.

Veksler, O., Boykov, Y., Mehrani, P., 2010. Superpixels and supervoxels in an energy optimization framework. In: European Conference on Computer Vision, pp. 211–224.

Wang, H., Wang, C., Luo, H., Li, P., Chen, Y., Li, J., 2015. 3-D point cloud object detection based on supervoxel neighborhood with hough forest framework. IEEE J. Sel. Top. Appl. Earth Obs. Rem. Sens. 8 (4), 1570–1581.

Weikersdorfer, D., Gossow, D., Beetz, M., 2012. Depth-adaptive superpixels. In: International Conference on Pattern Recognition, pp. 2087–2090.

Xu, C., Corso, J.J., 2012. Evaluation of super-voxel methods for early video processing, vol. 157(10), pp. 1202–1209.

Xu, L., Lu, C., Xu, Y., Jia, J., 2011. Image smoothing via L0 gradient minimization. ACM Trans. Graph. 30 (6), 61–64.

Yun, J.S., Sim, J.Y., 2016. Supervoxel-based saliency detection for large-scale colored 3D point clouds. In: IEEE International Conference on Image Processing, pp. 4062–4066.

Zhou, Y., Ju, L., Wang, S., 2015. Multiscale superpixels and supervoxels based on hierarchical edge-weighted centroidal voronoi tessellation. IEEE Trans. Image Process. 24 (11), 3834–3845.