

Boundary-enhanced supervoxel segmentation for sparse outdoor LiDAR data

Soothan Song, Honggu Lee and Sungho Jo

Voxelisation methods are extensively employed for efficiently processing large point clouds. However, it is possible to lose geometric information and extract inaccurate features through these voxelisation methods. A novel, flexibly shaped 'supervoxel' algorithm, called boundary-enhanced supervoxel segmentation, for sparse and complex outdoor light detection and ranging (LiDAR) data is proposed. The algorithm consists of two key components: (i) detecting boundaries by analysing consecutive points and (ii) clustering the points by first excluding the boundary points. The generated supervoxels include spatial and geometric properties and maintain the shape of the object's boundary. The proposed algorithm is tested using sparse LiDAR data obtained from outdoor urban environments.

Introduction: Correctly perceiving a three-dimensional (3D) outdoor environment is still a challenging task for autonomous vehicles. For this task, voxelisation methods are generally employed for efficiently processing large amounts of data [1]. Voxelisation is a method that divides the 3D terrain into fixed-length cubes and extracts their features. When a point cloud is converted into voxels, it is possible to lose geometric information about an unstructured outdoor object. Furthermore, the density of Velodyne light detection and ranging (LiDAR) data which is popularly used for 3D outdoor environments is low. In particular, point clouds distant from the sensor are much sparser. In this sparse region, it is possible that more than one object can be overlapped by the same voxel, and this voxel could contain inappropriate information about the objects. To overcome this problem, we propose a novel, flexibly-shaped supervoxel for sparse outdoor Velodyne LiDAR data. So far, there are no appropriate supervoxel algorithms for the Velodyne data.

To achieve this goal, we have developed two key algorithms: (i) a supervoxel algorithm that maintains the shape of the object's boundary and includes spatial and geometrical properties: we call it boundary-enhanced supervoxel segmentation (BESS); (ii) a boundary detection method that works well in sparse point clouds. We then evaluate our algorithm against other algorithms using standard error metrics. Fig. 1c shows the result of our supervoxel algorithm for 3D point clouds in outdoor urban environments.

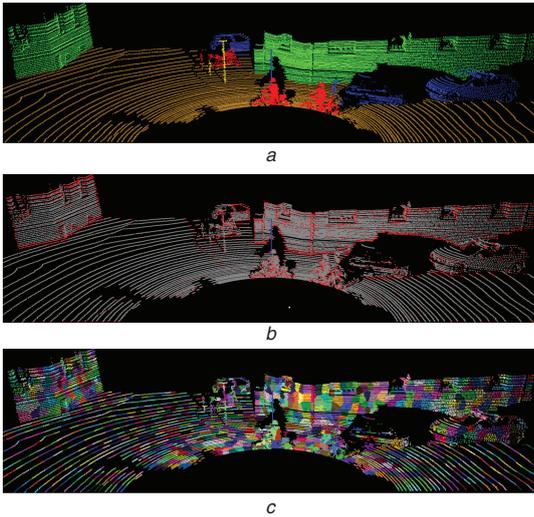


Fig. 1 Example of supervoxels produced by our algorithm

- a Ground truth point clouds
- b Detected edge points ($\hat{\theta}_i > 50^\circ$ and $\hat{\theta}_i < 20^\circ$)
- c BESS ($R = 0.6$ m and $w = 0.5$)

Boundary points detection: Our approach generates supervoxels by detecting boundary points. Point discontinuity is checked based on the angles formed by the lines connecting the point with its previous and next points. We consider vertical and horizontal consecutive points separately. Assuming that points are sequentially ordered in one direction, we define two vectors passing from a point $\mathbf{p}_i = [x_i, y_i, z_i]$ to its K (in our

setting $K = 3$) previous and next consecutive points as

$$\bar{\mathbf{v}}_i^+ = \frac{1}{K} \sum_{k=1}^K \left(\frac{\mathbf{p}_{i+k} - \mathbf{p}_i}{|\mathbf{p}_{i+k} - \mathbf{p}_i|} \right), \quad \bar{\mathbf{v}}_i^- = \frac{1}{K} \sum_{k=1}^K \left(\frac{\mathbf{p}_i - \mathbf{p}_{i-k}}{|\mathbf{p}_i - \mathbf{p}_{i-k}|} \right) \quad (1)$$

Then, the angle $\hat{\theta}_i$ representing the discontinuity of \mathbf{p}_i is defined as

$$\hat{\theta}_i = \cos^{-1} \left(\frac{\bar{\mathbf{v}}_i^+ \cdot \bar{\mathbf{v}}_i^-}{|\bar{\mathbf{v}}_i^+| \times |\bar{\mathbf{v}}_i^-|} \right) \quad (2)$$

Owing to point noise, analysing the consecutive points in a relatively very dense and sparse point clouds region could extract an incorrect discontinuous point. Furthermore, since points in unstructured objects such as foliage and grass are generally discontinuous, these points should be excluded from the boundary point set. Therefore, we should measure the variation among its K consecutive points and regard high variation points as noise points or part of an unstructured object. The variation angle $\hat{\theta}_i$ of a point \mathbf{p}_i is defined as

$$\hat{\theta}_i = \frac{1}{K} \sum_{k=1}^K \cos^{-1} \left(\frac{\bar{\mathbf{v}}_i^+ \cdot (\mathbf{p}_{i+k} - \mathbf{p}_i)}{|\bar{\mathbf{v}}_i^+| \times |\mathbf{p}_{i+k} - \mathbf{p}_i|} \right) \quad (3)$$

The points of high $\hat{\theta}_i$ and low $\hat{\theta}_i$ are determined according to a boundary point acting as a threshold, and they are computed in both vertical and horizontal directions. In this Letter, points with $\hat{\theta}_i > 50^\circ$ and $\hat{\theta}_i < 20^\circ$ are regarded as boundary points. We denote the discontinuity and variation angle in the vertical direction as $\hat{\theta}_i^V$ and $\hat{\theta}_i^H$, and in the horizontal direction as $\hat{\theta}_i^V$ and $\hat{\theta}_i^H$. The extracted boundary points are shown in Fig. 1b.

Point feature and distance measure: The geometric features of a point are extracted by analysing the local distribution of consecutive points [2]. For point features, we consider the angle $\bar{\theta}_i$ between the vector $\bar{\mathbf{v}}_i^+$ and the x - y -plane, defined as

$$\bar{\theta}_i = \sin^{-1} \left(\frac{\bar{\mathbf{v}}_i^+ \cdot \mathbf{z}}{|\bar{\mathbf{v}}_i^+| \times |\mathbf{z}|} \right) \quad (4)$$

where \mathbf{z} is a z -axis unit vector.

We denote the angle of the vertical direction as $\bar{\theta}_i^V$ and the horizontal direction as $\bar{\theta}_i^H$. In this Letter, the angles $\bar{\theta}_i^V$ and $\bar{\theta}_i^H$ and the variation angles $\hat{\theta}_i^V$ and $\hat{\theta}_i^H$ are used as geometrical point features. For normalisation, the angle features are divided by 90° . The point feature \mathbf{f}_i used for the supervoxel over-segmentation task is defined as

$$\mathbf{f}_i = \{x_i, y_i, z_i, \bar{\theta}_i^V, \bar{\theta}_i^H, \hat{\theta}_i^V, \hat{\theta}_i^H\} \quad (5)$$

By applying a weight w for point distances, the distance d of two point features is computed as follows:

$$d(\mathbf{f}_i, \mathbf{f}_j) = \sqrt{w \left(\frac{|\mathbf{f}_i^s - \mathbf{f}_j^s|}{\sqrt{3}R} \right)^2 + (1-w)(|\mathbf{f}_i^g - \mathbf{f}_j^g|)^2} \quad (6)$$

where $\mathbf{f}_i^s = \{x_i, y_i, z_i\}$ and $\mathbf{f}_i^g = \{\bar{\theta}_i^V, \bar{\theta}_i^H, \hat{\theta}_i^V, \hat{\theta}_i^H\}$. R is the resolution of an initial rectangular grid.

Supervoxel over-segmentation algorithm: The supervoxel over-segmentation algorithm is summarised in Algorithm 1. Our approach clusters the points by expanding a cluster region following the neighbourhood graph. The neighbourhood graph is constructed by connecting consecutive vertical and horizontal points. We then remove the edges of boundary points extracted from $\hat{\theta}_i$ and $\hat{\theta}_i$ so that objects cannot be connected to each other. Therefore, an expanding cluster is unable to grow past the boundary point and into the region of another object. Each cluster centre C_k is initialised to be the points of lowest discontinuity from the rectangular grid with step R in a 3D point cloud. Each cluster region changes by comparing the similarity and proximity of points in range $2R$ with each cluster centre. At the end of this process, the boundary points are assigned to the closest connected cluster.

Algorithm 1: Boundary-enhanced supervoxel segmentation

- Construct a neighbourhood graph G .
- Remove the edges of boundary points in G .
- Place cluster centres C_k on a rectangular grid with step R .
- Move each C_k to the lowest point of discontinuity in the range $R/2$.
- Set minimum distance $D_i^{\min} = \infty$ for each point i .

for each cluster C_k do

Extract the neighbouring points P^* in a $2R$ region around C_k .

Calculate distances D^* between P^* and C_k .

Remove points whose distance D^* is higher than D^{\min} from P^* .

Find the connected points \hat{P}^* of P^* from C_k using *breadth-first search* of G .

Assign each point of \hat{P}^* to cluster C_k and update D^{\min} to its distance D^* .

end for

Assign each boundary point to the closest connected cluster centre C_k .

Experimental results: To generate supervoxels, we use publicly available Velodyne datasets [3], which were recorded from an HDL-64E LiDAR sensor. To evaluate the performance of our algorithm, we compared our BESS algorithm against the voxel cloud connectivity segmentation (VCCS) algorithm [4]. It generates supervoxels by clustering voxel-clouds extracted from indoor RGB-D images. Since the VCCS generates supervoxels only in dense indoor RGB-D images, we modified it to be usable in Velodyne point clouds; it is denoted as point cloud connectivity segmentation (PCCS). This algorithm clustered point clouds instead of voxel-clouds using a neighbourhood graph constructed in the same way as ours. The distance in PCCS is defined as

$$d_{\text{PCCS}}(f_i, f_j) = \sqrt{w \left(\frac{|f_i^s - f_j^s|}{\sqrt{3R}} \right)^2 + (1-w) \left(|f_i^{\text{FPFH}}, f_j^{\text{FPFH}}|_{\text{HiK}} \right)^2} \quad (7)$$

where f^{FPFH} are the fast point feature histogram (FPFH) [5] features and their distance is calculated using the histogram intersection kernel. Then, the rest of the clustering algorithm is the same as [4].

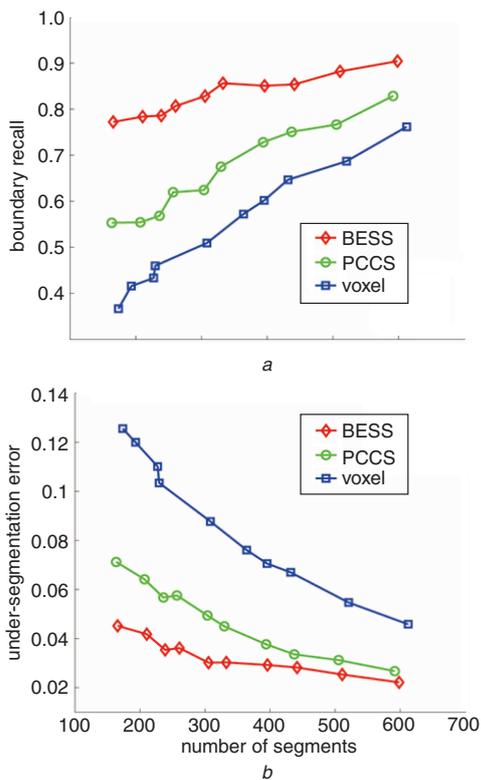


Fig. 2 Boundary recall and under-segmentation error of each algorithm

a Boundary recall

b Under-segmentation error

The boundary recall and under-segmentation error of the algorithms with respect to different grid steps R are shown in Figs. 2a and b, respectively. In these Figures, the number of segments is inversely proportional to the grid step R . That is, when R is high, the number of segments is low. Each result is the average of processing 10 frames. For supervoxel algorithms, we set the supervoxel parameter w of BESS and VCCS to 0.5. The results show that the BESS always shows better performance than the other algorithms. In particular, the boundary recall of BESS does not fall below 0.77. This indicates that our algorithm clustered point clouds by maintaining the object boundary even with a high grid step R .

To verify the performance of the supervoxel algorithms under different point cloud densities, relatively dense and sparse point clouds representing one object are extracted from two different frames. Since points detected on objects closer the sensor are denser than those detected farther away, we selected two frames that contained an object of considerable depth.

Fig. 3 shows examples of supervoxels in relatively dense (Fig. 3a) and sparse (Fig. 3b) point clouds. We set the supervoxel parameters w and R to 0.5 and 0.6 m, respectively. In Fig. 3, the performance of PCCS suffers in the case of sparse clouds when compared with dense clouds. BESS, on the other hand, performed better with sparse clouds. BESS can detect the correct boundary points even in sparse point clouds since the change between consecutive points in a sparse cloud is more obvious.

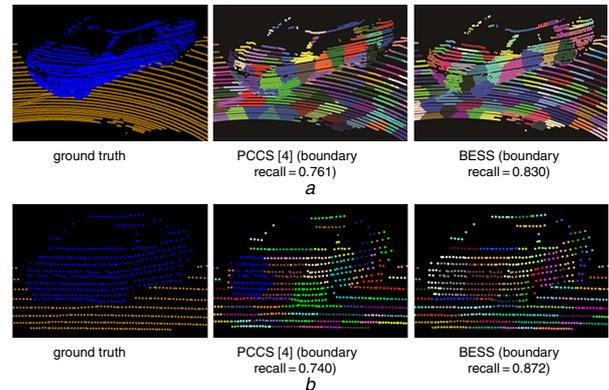


Fig. 3 Supervoxels of relatively dense and sparse point clouds

a Dense point clouds (distance of object from LiDAR centre is 5 m)

b Sparse point clouds (distance of object from LiDAR centre is 20 m)

Conclusion: We have presented a novel supervoxel algorithm for sparse outdoor LiDAR data. The algorithm detects boundary points and cluster points by excluding these boundary points. Therefore, the generated supervoxels maintain the shape of the object's boundary. Experimental results show that our algorithm performs better than other algorithms and the boundary detection is successful even for sparse point clouds.

Acknowledgments: This work was supported by the Technology Innovation Program, 10045252, funded by the Ministry of Trade, Industry and Energy (MOTIE, Korea).

© The Institution of Engineering and Technology 2014

6 September 2014

doi: 10.1049/el.2014.3249

One or more of the Figures in this Letter are available in colour online.

Soohwan Song, Honggu Lee and Sungho Jo (*Department of Computer Science, KAIST, 291 Daehakro, Yuseong-gu, Daejeon, Republic of Korea*)

E-mail: shjo@kaist.ac.kr

References

- Stoyanov, T., Magnusson, M., Andreasson, H., and Lilienthal, A.: 'Path planning in 3D environments using the normal distributions transform'. IEEE Conf. on Intelligent Robots and Systems, Taipei, Taiwan, October 2010, pp. 3263–3268, doi: 10.1109/IROS.2010.5650789
- Hadjiliadis, O., and Ioannis, S.: 'Sequential classification in point clouds of urban scenes'. Int. Symp. on 3D Data Processing, Visualization and Transmission, Paris, France, May 2010
- Geiger, A., Lenz, P., Stiller, C., and Urtasun, R.: 'Vision meets robotics: the KITTI dataset'. *Int. J. Robot. Res.*, 2013, doi: 10.1177/0278364913491297
- Papon, J., Abramov, A., Schoeler, M., and Worgotter, F.: 'Voxel cloud connectivity segmentation-supervoxels for point clouds'. IEEE Conf. on Computer Vision and Pattern Recognition, Portland, USA, June 2013, pp. 2027–2034, doi: 10.1109/CVPR.2013.264
- Rusu, R., Nico, B., and Michael, B.: 'Fast point feature histograms (FPFH) for 3D registration'. IEEE Conf. on Robotics and Automation, Kobe, Japan, May 2009, pp. 3212–3217, doi: 10.1109/ROBOT.2009.5152473