Automatic Object Segmentation from Large Scale 3D Urban Point Clouds through Manifold Embedded Mode Seeking*

Zhiding Yu[1,2], Chunjing Xu[1], Jianzhuang Liu[1], Oscar C. Au[2] and Xiaoou Tang[1,3]
[1]Shenzhen Key Laboratory for Computer Vision and Pattern Recognition, Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences
[2]Department of Electronic and Computer Engineering, The Hong Kong University of Science and Technology
[3]Department of Information Engineering, The Chinese University of Hong Kong
{zd.yu, cj.xu, jz.liu}@siat.ac.cn, eeau@ust.hk, xtang@ie.cuhk.edu.hk

ABSTRACT

This paper presents a system that can automatically segment objects in large scale 3D point clouds obtained from urban ranging images. The system consists of three steps: The first one involves a ground detection process that can detect relatively complex terrain and separate it from other objects. The second step superpixelizes the remaining objects to speed up the segmentation process. In the final step, a manifold embedded mode seeking method is adopted to segment the point clouds. Even though the segmentation of urban objects is a challenging problem in terms of accuracy and problem scale, our system can efficiently generate very good segmentation results. The proposed manifold learning effectively improves the segmentation performance due to the fact that continuous artificial objects often have manifold-like structures.

Categories and Subject Descriptors

I.4.6 [**Image Processing and Computer Vision**]: Segmentation—*Pixel classification*

General Terms

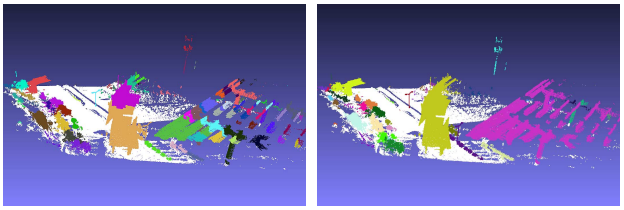
Algorithms, Experimentation

Keywords

3D point cloud, clustering, mode seeking, manifold

1. INTRODUCTION

3D models with geographical locations and semantical labels are the key for urban modeling that is widely used in a variety of applications, such as urban planning, simulation and visualization. Compared to modeling with traditional tedious and time-consuming CAD tools to create and visualize 3D digital urban models, many methods have been developed in recent years to obtain 3D information automatically. With these methods, it becomes possible that large scale 3D data can be obtained and processed.



(a) (b)

Figure 1: An example of urban object segmentation. (a) Mean shift segmentation. (b) Segmentation obtained by our system with manifold embedded mode seeking, which improves the result in (a) significantly.

As a common method to obtain 3D information, systems equipped with LIDAR (light detection and ranging) sensors are widely used. The obtained data, also called ranging images, can be represented by 3D point clouds. However, a cloud as a whole reveals only limited structure of the urban scene and is far from being an informative visualization. To further utilize it, a necessary step is to label and categorize the points in the cloud object by object.

Object segmentation plays a crucial role in the point cloud processing routine. By segmentation, the points composing an object is extracted from the scene for recognition and a semantical tag is then attached to it. Despite the abundant previous works, only a few are related to large scale urban scene object localization [5]. Most methods are designed for a much confined scenario, such as focusing on a specific class of objects like roads [6], vehicles [4, 10], trees [15, 14, 9], and buildings [7, 3, 11]. In these cases, either the scenario is relatively simple that contains only a few objects [8], or the input objects have been segmented from the scene. The theme of these papers is mainly related to recognition and reconstruction of a specific class of objects where object recognition (or classification) techniques play a central role.

The urban object recognition system proposed in [5] uses normalized cuts [13] and min cut [1] to localize and partition point cloud objects. Other graph partitioning methods such as graph cut [2] are also useful tools for partitioning a set of points into subgroups. Our data, however, consist of much more complicated urban scenes, which render these methods fail in our experiments. The major challenge is basically twofold: First, point clouds typically consist of complicated, densely aligned objects with large size variation. Second, the inherent computation consuming nature of segmentation further leads to difficulties in dealing with large scale

*Area chair: Pal Halvorsen

MM'11, November 28–December 1, 2011, Scottsdale, Arizona, USA.

problems, reducing the practicality of a method. In this paper, we develop a system that can automatically segment objects in a 3D cloud of a large scale urban scene that contains billions of points. An example of the segmentation using our system is illustrated in Fig. 1(b). It is worth mentioning that Golovinskiy et al. [5] and Lim et al. [8] also developed a system for similar purpose. The differences between our work and theirs include that 1) our system can extract complicated terrain maps, while [5] and [8] assume relatively simple, flat terrains. Without accurate terrain extraction, segmentation of some objects is easy to fail, 2) buildings that later can be used for further mesh reconstruction are included and segmented, 3) our data are highly noisy and heterogeneous due to the data acquisition and preprocessing method[1], and 4) the objects in our data are highly occluded since the data are generated by going through the streets once.

2. THE PROPOSED METHOD

Fig. 2 shows how our method works. The three main steps, ground (terrain) detection, superpixelization, and object segmentation, are described as follows.

2.1 Terrain Detection

The first step of our system is to detect terrain from a point cloud. The terrain of a scene is related to the resolution we consider:

DEFINITION 2.1. *Let $C = \{p_i = (x_i, y_i, z_i) | i = 1...N\}$ be a perfectly captured scene without occlusion and noise where N is the number of points in C, and Δ be the resolution with which each pixel on a terrain map represents a $\Delta \times \Delta$ square in the real scene. The terrain map T^Δ with resolution Δ is defined as:*

$$T^\Delta(k,l) = \min_{\{p \in C | (x_\Delta(p), y_\Delta(p)) = (k,l)\}} z(p), \quad (1)$$

where $z(p)$ is the function to have the z-coordinate, along with $x_\Delta(p)$ and $y_\Delta(p)$ being the functions that retrieve the grid indices along the x-axis and the y-axis respectively when the xy-plane is masked with a $\Delta \times \Delta$ grid.

It is not difficult to develop a simple algorithm to generate the terrain map based on the definition if a perfectly captured scene is given. In real cases, however, occlusion and noise are ubiquitous. For example, parts of the ground hidden under cars cannot be detected. We use the following techniques to overcome this problem: 1) a maximum threshold is set to detect an abrupt change of the local minimum of heights for points, 2) paved road detection, and 3) expansion from paved road to obtain a better terrain.

The ground extraction algorithm is a two-pass process consisting of rough ground extraction and further refinement. In the first step, we divide the xy-plane into regular grids of size $\Delta \times \Delta$. For each grid, we eliminate points higher than the lowest point with a height difference larger than $Thre_1$. Mathematically, we define the roughly extracted ground as a set of extracted points satisfying the following condition:

$$\begin{aligned} T_1^\Delta(k,l) = \{z(q) | q \in C, (x_\Delta(q), y_\Delta(q)) = (k,l), \\ z(q) - \min_{\{p \in C | (x_\Delta(p), y_\Delta(p)) = (k,l)\}} z(p) \leq Thre_1\}, \end{aligned} \quad (2)$$

[1]The ranging images are obtained by driving a vehicle loaded with LIDAR censors on the street. Due to the local traffic, the objects in the scene, such as vehicles and working people, are not stationary, causing some objects distorted. Another reason for low quality of the data is that our scene covers a much larger urban area, with the spanning about 10 kilometers. So the density of the data at many locations is low where it is hard to define objects.

The first step can effectively remove the majority of the objects while preserving uneven portions of the ground. Since we set Δ to 1.5 and $Thre_1$ to 1.7 in this paper, our system can tolerate inclination up to approximately 40 degrees. The method can also eliminate vertical structures with abrupt rise. Most of these structures correspond to building facades, walls, overpass pillars and trunks and should be considered as objects.

The second step aims at eliminating undesired object residues. Since the detected point clouds show an elongated shape along the road, we perform principal component analysis (PCA) with the cloud points on the xy-plane and select the eigenvector corresponding to the largest eigenvalue to indicate the road direction. We then divide the points into stripes of width Δ_s along the road direction, using multiple hyperplanes perpendicular to the road. Within each stripe, the points are further divided into grids of size $\Delta_g \times \Delta_s$.

Suppose we use l_k to indicate the l_kth grid in the k stripe, T_1^k to indicate the point set in the kth stripe, and T_1^{k,l_k} to indicate the point set corresponding to the l_kth grid. To locate the expansion starting point, the grid containing the paved road in the kth stripe is detected as:

$$l_{Road} = \arg\min_{l_k} \frac{1}{|T_1^{k,l_k}|} \sum_{p \in T_1^{k,l_k}} (z(p) - z_{Road})^2, \quad (3)$$

where the road height z_{Road} is defined as:

$$z_{Road} = \text{mean}(\{z(q) | q \in T_1^k, z(q) \in Bin_{max}(q)\}) \quad (4)$$

and the maximum bin Bin_{max} is defined as the largest bin of the histogram of point heights ranging from $\min_{p \in T_1^k}(z(p))$ to $\min_{p \in T_1^k}(z(p)) + Thre_2$. Experiments show that most road points are well detected, with similar heights densely concentrated in the height histogram to form the largest bin. Thus it is reasonable to estimate the road height by calculating the mean of the largest bin.

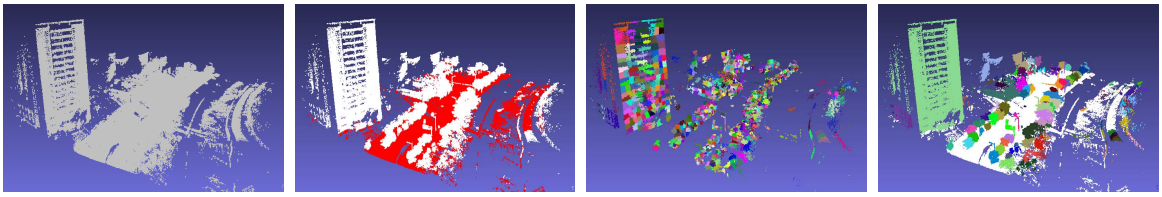
The detected grid is taken as the starting point. We eliminate points within this grid that are higher than the lowest point with height difference more than $Thre_3$ and mark this grid as "refined". We then propagate the refinement from the starting grid in a spatially continuous manner along the stripe. For each unrefined grid, we refer to the highest point that is identified as the ground in the previously refined grid and denote it as the reference point. Each unidentified point in the unrefined grid is compared with the reference point. Those higher with height differences larger than $Thre_3$ are eliminated and the rest are identified as ground. The highest among these points is selected as the new reference point. If there are no newly identified ground points, the reference point is not changed. The above process is repeated until all grids are refined in the map. We select Δ_s, Δ_g, $Thre_2$ and $Thre_3$ respectively as 10, 1, 10 and 1 and set the bin size as 1.

2.2 Point Cloud Superpixelization

The residual $C - T$ by removing the detected terrain T from a given cloud C is to be segmented. In order to solve large scale segmentation problem, a necessary step prior to the segmentation of objects is superpixelization. We first use mean shift oversegmentation with a bandwidth 5. The obtained clusters are further refined by recursively performing cluster split using 2-cluster k-means clustering, until each cluster contains less than 300 points.

2.3 Segmentation of Objects

Since the number of objects in a given urban scene is unknown, a clustering algorithm that needs this number known is not suitable for such a task. Mode seeking methods such as mean shift can serve as an appropriate tool. Besides the ability to automatically



(a) (b) (c) (d)

Figure 2: An example illustrating the segmentation process. From left to right are figures of the original point cloud, ground (in red) detection, superpixelization and object segmentation.

determine the cluster number and the potential for parallelization, mode seeking can accurately detect many objects (such as trees) even though they are densely aligned. We observe that these objects tend to have high point density at their centers while showing low density at their boundaries, which particularly favors such method.

Our method is closely related to mean shift but has many additional features that prove to be essential for obtaining good segmentations: 1) a directional biased kernel bandwidth with z axis suppression, 2) minimum panning tree (MST) embedded mode seeking [16] that can detect compact structures and favor point connectivity with manifold-like point clouds, and 3) a large size prior for buildings with adaptive kernel size.

The observation for introducing the first feature is that points tend to have larger correlation along the vertical direction than horizontal directions. It inspires us to increase the kernel bandwidth along the vertical direction to strengthen point connectivity. This is equivalent to suppressing the z-axis coordinates for all the points and performing subsequent operations including MST construction and mode seeking in the z axis suppressed coordinate space. In this paper, we suppress the z coordinates with a ratio of 0.5.

Suppose the superpixels of a point cloud are represented by a set $\mathbf{V} = \{\mathbf{v}_i | i = 1, \ldots, N, \mathbf{v}_i \in \mathbf{R}^d\}$, where N is the total number of superpixels. The dimensionality d equals 3 in our case and $\mathbf{v}_i$ is the mean z-suppressed space coordinate of the points belonging to the ith superpixel. For a given set of superpixels, we construct its full connection graph $\mathbf{G} = (\mathbf{V}, \mathbf{E})$ where $\mathbf{E} = \{e_{i,j} | i, j = 1, ..., N, i \neq j\}$ and $|e_{i,j}| = \|\mathbf{v}_i - \mathbf{v}_j\|$. Using Kruskal algorithm we are able to extract the minimum spanning tree (MST) $\mathbf{S} = (\mathbf{V}, \mathbf{E_S})$, which is a connected graph of $\mathbf{G}$ with $\mathbf{E_S} \subseteq \mathbf{E}, |\mathbf{E_S}| = N - 1$. For any node pair (i, j), $i \neq j$, there exists a unique path $\mathbf{E}_{ij}$ such that $\mathbf{E}_{ij} \subseteq \mathbf{E_S}$, i and j are connected sequentially by elements of $\mathbf{E}_{ij}$ and deleting any one of the elements results in the disconnection of i and j. In addition, we define $\mathbf{E}_{ij}$ to be $\emptyset$, if $i = j$.

We propose to use a joint representation of the MST distance space ("MST space" for short) and the feature space (the coordinate space) to define the density estimator. Consider the simplest case where the MST space kernel center is located exactly at a tree node $\mathbf{v}_j$. Then the density estimator can be written as follows:

$$f(\mathbf{v}) = c_0 \sum_i k\left(\frac{d(\mathbf{v}_j, \mathbf{v}_i)^2}{h_1^2}\right) k\left(\left\|\frac{\mathbf{v} - \mathbf{v}_i}{h_2}\right\|^2\right), \qquad (5)$$

where $d(\mathbf{v}_j, \mathbf{v}_i)$ is the cumulative length of the path connecting node $\mathbf{v}_i$ and $\mathbf{v}_j$, defined as:

$$d(\mathbf{v}_j, \mathbf{v}_i) = \sum_{e_{m,n} \in \mathbf{E_{ij}}} \min_{p_k \in \mathbf{V}_m, p_l \in \mathbf{V}_n} ||p_k - p_l||. \qquad (6)$$

In (6), p_k is the single point coordinate of the kth point and $\mathbf{V}_m$ represents the set of points belonging to the mth superpixel. We use the above form instead of $d(\mathbf{v}_j, \mathbf{v}_i) = \sum_{e_{m,n} \in \mathbf{E_{ij}}} ||v_m - v_n||$ defined in [16] to further reward attraction along manifolds. In (5), $\mathbf{v}$ is the feature space kernel center, h_1 and h_2 are the bandwidth parameters controlling the window size, c_0 is a constant determined by the sample size and bandwidth, and $k(x) = \exp(-\frac{1}{2}x)$ is the profile of a normal kernel.

The inference of mode seeking for the tree-embedded density estimator is well described in [16]. We employ the fast approximation of the mode seeking process by iteratively shifting the MST space kernel and the feature space kernel, which is described in [16]. We also employ an adaptive bandwidth, utilizing a large size prior for buildings. We re-cluster superpixels from clusters containing at least one superpixel higher than the road height for 7.5 in the z axis suppressed space, with the same superpixel set and MST structure as those in the previous step, but the bandwidths for re-clustered superpixels are enlarged to 7.2 and 12, which are six times the original bandwidths. Combining the cluster label for low altitude superpixels together with the re-clustered result, we can obtain the final cluster label for each point in the point clouds.

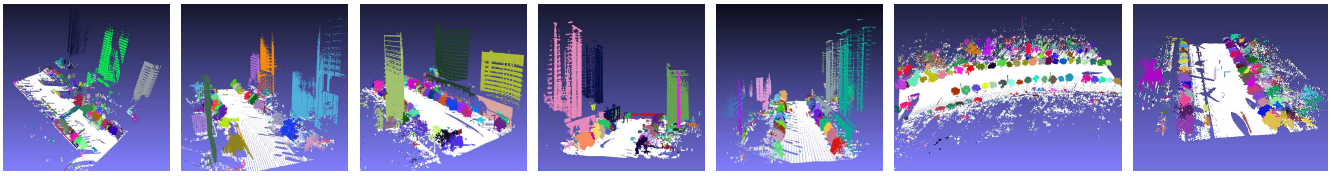
Finally, we emphasize again that our method can automatically label different objects without object recognition after the terrain detection. One example is given in Fig. 2(d), and more can be seen in the next session.

3. EXPERIMENTS

We conduct comprehensive experiments to test the proposed system on a data set containing 176 images obtained by scanning some streets of our city. To better illustrate its performance, the results are compared with results obtained by mean shift, a standard segmentation method widely adopted in the literature for point cloud segmentation [5, 12]. The mean shift algorithm is operated on the same object superpixels obtained through the first and second steps of our method. It also uses adaptive bandwidth in the our experiments, with the same parameter adaptivity settings to our method.

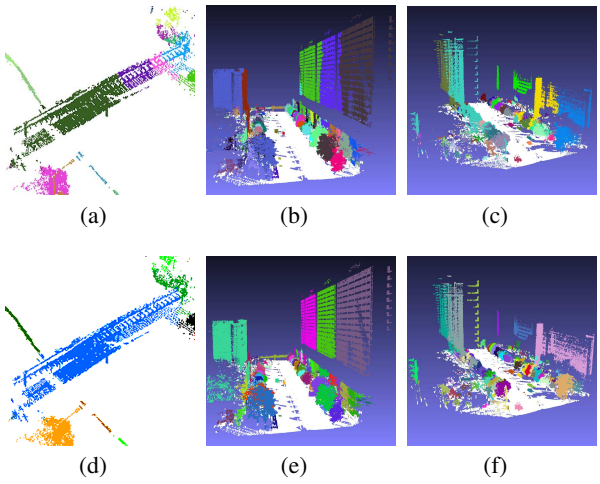
3.1 Qualitative Evaluation

We randomly illustrate some ranging images with representative urban scenes and perform ground detection and segmentation. The results indicate our system can generate very good segmentation under a variety of complicated scenes. The scenes illustrated in Figs. 1 and Fig. 2 contain complex terrain which is hard to extract. Our system can accurately extract the majority of the ground. It is also worth mentioning that the embedding manifold structure helps to improve segmentation on spanning objects, particularly overpasses and buildings, as illustrated in Fig. 1, Fig. 4(a) and Fig. 4(d). Notice that for the data illustrated in Fig. 1, there is no way for mean shift to segment the whole bridge without erroneously including nearby objects (the pylon on the right). Fig. 4(e) illustrates our segmentation result on a challenging task where large buildings are densely aligned while the scanned points are sparse on these buildings. The buildings are difficult to separate but our system can generate very accurate segmentation. Fig. 3(f) and Fig. 3(g) contains densely aligned trees difficult to separate. Our method works



(a) (b) (c) (d) (e) (f) (g)

Figure 3: A portion of the segmentation results obtained by our system. The test data contains typical urban scenes.



(a) (b) (c)

(d) (e) (f)

Figure 4: A comparison of segmentations obtained by our system and mean shift. The second row contains better results obtained by our system, while results obtained by mean shift contain both serious oversegmentation and oversmoothing.

well on such kind of data. Tested on the whole data set containing hundreds of scenes similar to the above illustrated ones, the visual examination suggests that our system has similar performance to the illustrated ones. In all our experiments, the bandwidth for mean shift is set to 4 and the two bandwidths h_1 and h_2 for our method are respectively set to 1.2 and 2. These parameters are empirically selected to optimize the performance of the two methods.

3.2 Quantitative Evaluation

We compare each automatic segmentation against the ground truth [2] segmentation by finding (a) how much of the automatic segmentation contains the whole object (precision), and (b) how much of the object is not oversmoothed with other objects (recall). The result is illustrated in Fig. 5. Results show that our method significantly outperforms mean shift.

4. DISCUSSION AND CONCLUSION

In this paper, we have developed a system that can automatically segment objects in complicated urban scenes. The system can separate single, relatively small objects while preserving the connectivity of large, spanning ones. It provides good initial interpretations of the urban scenes, based on which 3D object reconstruction, visualization and recognition can be carried out.

5. ACKNOWLEDGEMENTS

This work was supported by grants from Introduced Innovative R&D Team of Guangdong Province (Robot and Intelligent Information Technology), Natural Science Foundation of China (61005011, 60975029, 61070148), and Science, Industry, Trade and Information Technology Commission of Shenzhen Municipality, China (JC201005270358A, JC201005270350A, JC200903180635A, JC201005270378A, ZYC201006130313A).

[2]Since the work load is huge, we only label a portion of the data set and select objects that are easy to label.

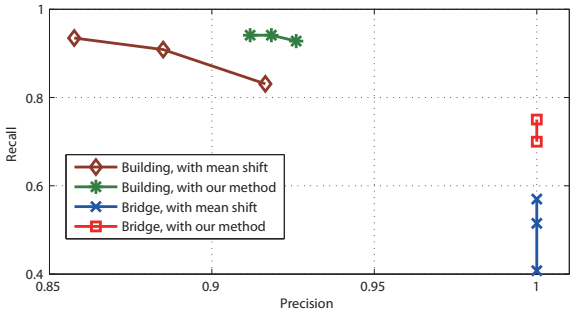


Figure 5: A quantitative comparison of segmentations obtained by our system and mean shift.

6. REFERENCES

[1] Y. Boykov and G. Funka-Lea. Graph cuts and efficient n-d image segmentation. *IJCV*, 70(2):109–131, 2006.
[2] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *IEEE T-PAMI*, 23(11):1222–1239, 2002.
[3] J. Chen and B. Chen. Architectural modeling from sparsely scanned range data. *IJCV*, 78(2):223–236, 2008.
[4] A. Frome, D. Huber, R. Kolluri, T. Bulow, and J. Malik. Recognizing objects in range data using regional point descriptors. *ECCV*, 2004.
[5] A. Golovinskiy, V. Kim, and T. Funkhouser. Shape-based recognition of 3d point clouds in urban environments. *ICCV*, 2009.
[6] A. Jaakkola, J. Hyyppa, H. Hyyppa, and A. Kukko. Retrieval algorithms for road surface modelling using laser-based mobile mapping. *Sensors*, 8(9):5238–5249, 2008.
[7] F. Lafarge, X. Descombes, J. Zerubia, and M. Pierrot-Deseilligny. Building reconstruction from a single DEM. *CVPR*, 2008.
[8] E. Lim and D. Suter. Conditional random field for 3D point clouds with adaptive data reduction. *Int'l. Conf. on Cyberworlds*, 2007.
[9] Y. Livny, F. Yan, M. Olson, B. Chen, H. Zhang, and J. El-Sana. Automatic reconstruction of tree skeletal structures from point clouds. *ACM Transactions on Graphics*, 29(5), 2010.
[10] E. Meier and F. Ade. Object detection and tracking in range image sequences by separation of image features. *IEEE International Conference on Intelligent Vehicles*, 1998.
[11] L. Nan, A. Sharf, H. Zhang, D. Cohen-Or, and B. Chen. SmartBoxes for interactive urban reconstruction. *ACM SIGGRAPH*, 2010.
[12] X. Shao, K. Katabira, R. Shibasaki, and Z. H.J. Multiple people extraction using 3D range sensor. *SMC*, 2010.
[13] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE T-PAMI*, 22(8):888–905, 2002.
[14] Y. Wang, H. Weinacker, and B. Koch. A lidar point cloud based procedure for vertical canopy structure analysis and 3D single tree modelling in forest. *Sensors*, 8:3938–3951, 2008.
[15] H. Xu, N. Gossett, and B. Chen. Knowledge and heuristic-based modeling of laser-scanned trees. *ACM Transactions on Graphics*, 26(4):19, 2007.
[16] Z. Yu, O. Au, K. Tang, and C. Xu. Nonparametric Density Estimation on A Graph: Learning Framework, Fast Approximation and Application in Image Segmentation. *CVPR*, 2011.