The 14th IFToMM World Congress, Taipei, Taiwan, October 25-30, 2015
DOI Number: 10.6567/IFToMM.14TH.WC.PS20.014

Surface Reconstruction with Sparse Point Clouds of Velodyne® Sensor

Tianwei Zhang*
University of Tokyo
Tokyo, Japan

Yoshihiko Nakamura †
University of Tokyo
Tokyo, Japan

Abstract—*Fast surface reconstruction from dense point clouds data(PCD) is a popular topic in robotics. However, it is still challenging to make large-scale surface maps with sparse PCD. This paper proposes a method to build surface maps using the PCD generated by velodyne® HDL-32E laser scanner, which is wildly used in automatic driving vehicles and wild exploring robots. Supervoxel clustering method is employed in this paper to organise the spare point clouds, and scale variable triangular meshes are generated to represent the large-scale outdoor scenes. Experimental results show that the proposed method is efficient and robust even in uneven terrains contain complex obstacles.*

Keywords: surface maps, sparse point clouds, velodyne® HDl-32E

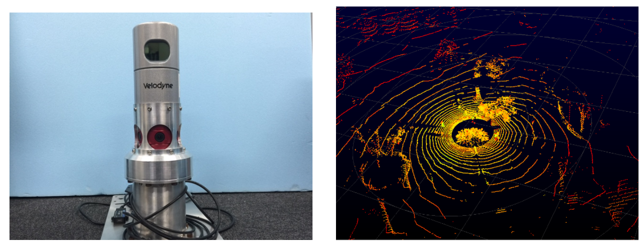
I. Introduction

3D point clouds processing methods have achieved significant process in many fields such as segmentation, frame-to-frame registration and SLAM. However, there are still changing problems in real-time implementation performances, which are important to robots and automatic driving vehicles. This paper proposes a fast surface reconstruction method using the sparse PCD generated by Velodyne® laser scanner. Velodyne® HDL-32E, which captures 70,000 points per second (10 frames with around 70,000 points per frame). Each frame provides $360°$ depth measurements horizontally, and $40°$ vertically with 32 laser beams. The pitch lasers range from $+10°$ to $-30°$. As the slope angles are fixed, each laser produces a ring of points on the ground, see Figure 1.

As discussed in [1], point clouds processing methods lacks speed due to the increasing of input data and lacks accuracy due to density decreasing. The real-time processing of velodyne® scanner's data has the two bottlenecks: On the one hand, to achieve enough space structure information from the unknown environment, Velodyne® HDL-32E laser scanner rotates 32 laser beams to extract $360°$ view. These discrete beams result quite sparse point clouds in far away distance. High sparsity of points may miss specific obstacles, hence restricts the application of greedy growing surface reconstruction.

In another respect, although the laser scanner has reduced

*zhang@ynl.t.u-tokyo.ac.jp
†nakamura@ynl.t.u-tokyo.ac.jp



(a) Velodyne® HDL-32E scanner (b) Point Clouds

Fig. 1. Velodyne® HDL-32E and its output point clouds

the PCD density a lot, the output PCD files are still very large. For instance, the Velodyne® scanner used in this paper outputs around 70,000 points in a nearly 2MB pcap format file per second. Dividing 3D space into small grids is an efficient and commonly used way to reduce PCD size in approximation methods. However, the sparse point clouds make these space adjacency based methods powerless, because there are not enough points to guarantee the structure information of each grid, or not enough neighbour grids to represent the local space information.

This paper proposed a fast surface reconstruction method to solve above problems. Aiming at real-time processing of sparse 2.5D point clouds, supervoxel segmentation method is employed to build larger grids with enough points data to represent the local space structure. Additionally, input size is reduced from dozens of thousand to several hundred by using the supervoxels as input. Triangular meshes are generated by connecting the supervoxels and their neighbours, then obstacles are estimated by a simple and fast sharp feature extraction method. Main contributions of this paper are as follows:

(i) Supervoxels are novelly utilized to segment the sparse PCD, different from the original supervoxel clustering method of [2], to cope with the density shifting in sparse PCD, a seeding method is proposed to generate the variable scale supervoxels and build the corresponding adjacency graph with discrete voxel maps.

(ii) The Delaunay triangles are novelly generated by connecting the centers of constructed voronoi polygons, which are generated by clustering the unit voxels with a defined distance. This triangulation is fast and nearly real-time.

(ii) We proposes an efficient sharp feature extraction method without using color information. This method can be used to identify sharp edges in dense point clouds, and estimate obstacles in sparse point clouds. It is flexible to generate big meshes in planner surfaces and small meshes in sharp edges and boundaries regions.

(iii) No priori information is used, so that the proposed method can be applied in both man-made streets and wild uneven terrain.

II. Related Works

A. Surface Reconstruction Methods

Fast Surface Reconstruction with depth sensors [3], [4] has become to be one of the most efficient methods to understand unknown environments. They are to create a respective and continuous triangular mesh surface using the given noisy point clouds data, in which the normals and coordinates of obstacles are also provided for collision detection. There are mainly two kinds of surface reconstruction methods: point interpolation based methods and points approximating methods. The former group try to interpolate points usually by projecting input point clouds to a space distribution such as Delaunay triangulation [5], which has beautiful triangles distribution. Famous methods like ball-pivoting method [6],[3] incrementally construct triangles with greedy growing and use local strategies to determine insertion of points. Obviously, interpolation methods require dense point clouds. Although we can scan repeatedly to get dense PCD, but it is hard to guarantee an acceptable density.

In another respect, approximation based methods are widely studied[2],[7]. These methods save a lot of time in points sampling by dividing the point clouds into small regular grids, and such grids, usually called voxels, are used to represent the real surface and extract the final triangle meshes [7]. As removing a lot of points, normals should be estimated before the division step. Widely used normals estimation method like [8], outputs normal clouds approximated by K-near neighbours, so that the normal clouds are smooth without sharp features. As sharp normals are important for recognizing boundary regions, [9] proposed a randomized hough transform method to determine the probability distribution of boundary normals, then make normal clouds sharper.

B. Supervoxels Methods

Recently, “Supervoxels” approaches are popular in PCD processing. The concept of supervoxels is an extension of “superpixel” segmentation [10], which is an over segmentation technique to cluster pixels of same feature into big “patches”. Such patches greatly reduce input complexity , and are usually used as input for afterward algorithms. [2] extends the 2D pixels to 3D voxels, and makes it is possible to process millions of input points in real time.

C. Other Works with Velodyne® Scanner

Velodyne® sensors are popular in the study of automatic driving vehicles. [12] compares several famous 3D LIDAR point clouds segmentation methods and proposes a probabilistic ground modeling method by accumulating multiple sensors. [14] designs a SLAM approach for velodyne® sensors and obtains precise trajectories. [1] presents a small planes finding algorithm for frame-to-frame registration and works well in velodyne® sparse datasets. Recent works like [15] provides a vehicle target tracking method for highway automatic driving application.

III. Supervoxels Segmentation

A. Space dividing

To reduce the size of input PCD, we divide the 3D space into unit voxels with a small resolution named voxel resolution. To indicate the local space structure, the normal of each voxel are computed with K-near neighbours of the center point by using the estimation method of [8]. Note that the other approximation method usually filter isolate points at this step, but we won’t do that here, since that in our sparse point clouds, a half points are “isolate”. What we need to do is to filter in “supervoxel” scale. Then, we divide the voxel center point clouds into bigger cube volumes with a larger seed resolution, and we remove the volumes which do not have enough unit voxels inside. Finally, the seeds which are used to start clustering in the coming phase are setted to the center coordinates of each volume. The normal of each seed is computed by the mean of normals inside the volume.

B. Distance measure

To segment point clouds into groups having similar points, a defined distance measurement is important to separate points with different features. In [2], aiming at identifying obstacle boundaries, their supervoxels clustering distance measurement contains spatial distance, color distance and Fast Point Feature Histograms(FPFH) feature space distance. While in our case, laser scanner cannot provide color data and FPFH features are powerless in sparse point clouds, since normals are more robust and efficient, in this proposed method, distance measure is defined as:

$$D = \sqrt{\frac{\mu D_s^2}{3R_{seed}^2} + \theta D_n^2} \tag{1}$$

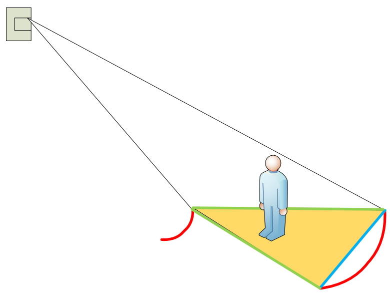
where the μ and θ are the influence factor of spatial distance and normal value. R_{seed} is the original seed resolution, which decides the scale of triangular meshes in the dense parts of PCD. Normal distance D_n is computed as

$$D_n = 1 - \vec{n_s} \cdot \vec{n_v} \tag{2}$$

where n_s indicates the normal of the center point of each cluster, and n_v is the normalized sum normals of the query voxel.

C. Clustering and segmentation

This paper uses K-mean for clustering, Firstly we computed the seeds of each volume, and add K nearest vox-



Reliable distance of supervoxel adjacency

Fig. 2. the red lines are Velodyne® point rings and the blue line is a valid edge. As the separated property of the sensor, obstacle maybe missed on the yellow mesh. To judge if these two green edges are reliable, we compare their length with their neighbour supervoxels' width multiple two

els into the cluster queue, compute distance between the query voxel to the seeds though Eq 1, add the query voxel into the nearest seed, and its neighbours (if exist) will be added to the searching queue of its owner seed. This iteration keeps on going until the twice failure expanding, i.e., no new voxel is added in. Note that K is a number decided by the ratio of seed resolution and voxel resolution. To deal with the sparse point clouds which don't have space continuous neighbourhood, in each iterator, a incremental K is used in K-nearest searching.

Each supervoxel expands at the same rate, and the center of supervoxel is updated by the mean of all the members after each expanding. Finally, the supervoxels grow to be voronoi polygons. If the center points are regarded as sites of voronoi map, they should be the nearest site for the member voxels belonging to that supervoxel. Then, as the dual graph of voroioi map, Delaunay triangular meshes can be obtained by connecting the center points of adjacency supervoxels. Note that, as mentioned above, the supvoxels may not have continuous adjacency. Meshes between discontinuous supervoxels may result missing of obstacles. To judge if the discontinuous neighbours are reliable to connect, we define twice supervoxel's average width as the reliable distance for triangulation. See Figure 2. Note that, the scale of supervoxels is changeable. At dense regions, the width of supervoxels will be around seed resolution, at sparse parts of the point clouds, the supervoxels may grow larger to cluster similar voxels. Therefore, the width of supervoxels are computed by average distance of new owned voxels, and updated after each expanding. Figure 3 indicates the clustering result.

IV. Sharp Feature Extraction

A sharp feature estimation method for large scale outdoor scenes is proposed in [11] called Difference of Normals (DoN) method. In DON, the normal clouds are computed with two different support radius, and the difference

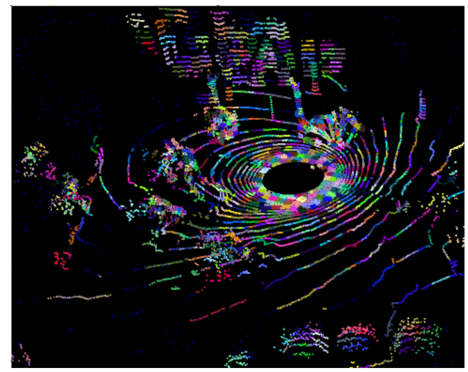
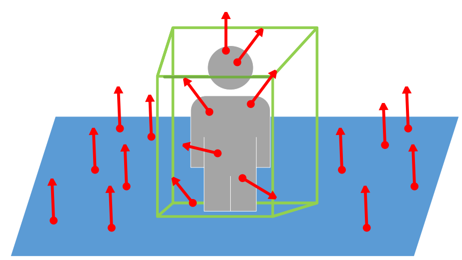


Fig. 3. Segmented Supervoxels



Sharp feature extraction

Fig. 4. Red points and arrows stand for voxels and their normals, the green cube is a clustered supervoxel within an obstacle. As the directions of obstacle normal vectors are not same, the length of sum normal is shorter than voxels number

between two results is used to indicate if the query point is sharp or not. And to cope with smooth normal clouds, the normals of whole point clouds have been computed twice with the complex computing method introduced in [9]. However, as it is shown in Figure 5 , the sharp normals estimation method of [9] can not work in sparse point clouds, so that the DON is not obvious. Therefore, we propose a method to extract sharp feature for obstacle detecting and tracking applications by considering about the length of the sum of normals inside each segmented supervoxel:

$$L_n = |\sum \vec{n_i}|, \vec{n_i} \in \mathcal{S} \tag{3}$$

Look at the situation in figure 4, there are several voxels (instead by red points) in a supervoxel (the green frame) occupied by an obstacle (the gray man). Obviously, the length of sum normal vector should be shorter than unit length multiple voxels number. We set a threshold t, and if L_n is smaller than $t * N$ (N is the number of points inside the voxel), this supervoxel will be labelled as sharp.

The magnitude of t depends on the computed normal clouds of the input PCD. There are many different normals estimation methods exist. We chose one of the simplest

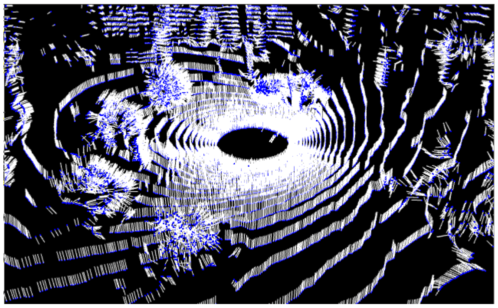


Fig. 5. normal clouds

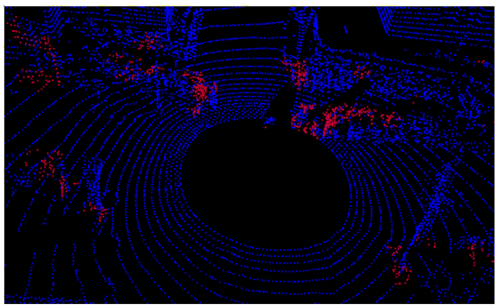
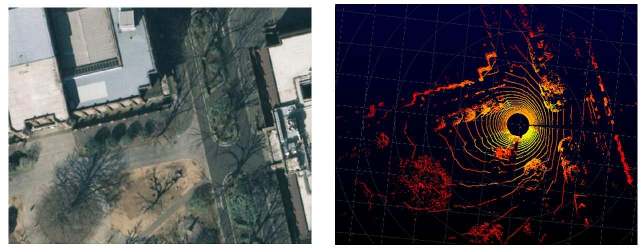


Fig. 6. Sharp supervoxels in red



(a) Satellite view from Google Map (b) Point clouds datasets

Fig. 7. Campus scene datasets captured by Velodyne® HDL-32E

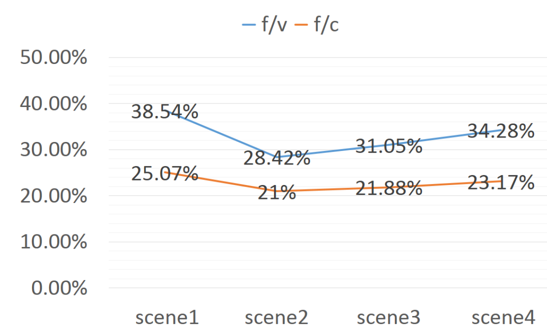


Fig. 8. face/vertex and face/edge

from [8], in which the normals are computed by analysis of the eigenvectors and eigenvalues of a covariance matrix created from their k-near neighbours. The neighbour size k determines the quality of the normal clouds. Theoretically, the bigger the k is, the bigger the t is, because the edge points influence much on neighbour points and make their normals sloped. Obviously, k should be big enough to indicate the local structure, but it results in too big supporting platforms in sparse regions. On the other hand, the number of voxels owned by supervoxels also has a big influence on t. Less voxels cannot provide reliable sharp feature, that's why we remove the supervoxels which don't have enough voxels. Figure 6 shows the extracted sharp point clouds coloured in red.

V. Implementation and results

The proposed method is applied on a novel complex dataset containing three short and one long loop pcag files captured in campus scenes.There are totally 2480 frames with around 65000 to 70000 points extracted on each frame. This dataset contains both man made roads and uneven terrains. Obstacles include trees, people, vehicles, buildings and streetlights, as shown in Figure 7, and the corresponding satellite view on the left. The machine used for our experiments is an eight-core Intel(R) Xeon(R) @ 2.66 GHz with 16 GB system memory.

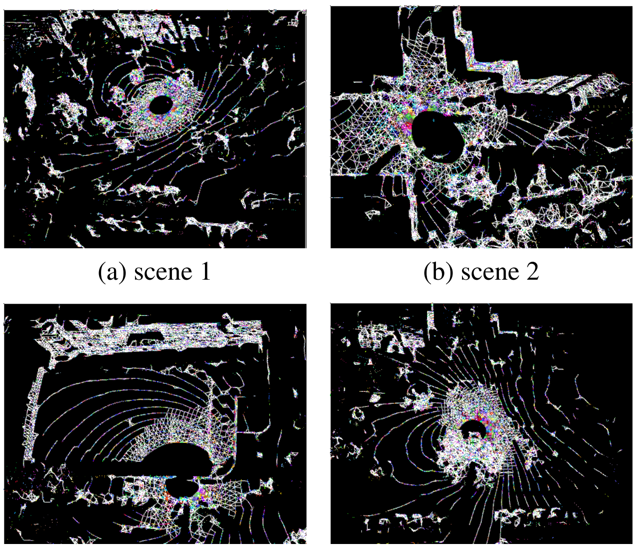
See Figure 7, the dataset is captured in large campus scenes, where the scale is about $100m \times 100m$(estimate from google map). Using Delaunay triangular strategy, the generated triangles are well distributed at dense point region near to the senser. As mentioned above, we limit the triangles' edge length in $1 \sim 2$ m, to ensure that there is no meshes at unreliable far away regions. Using scale changeable supervoxels and extracted sharp features, even remote obstacles in sparse point clouds can be recognised and reconstructed, such as the far away buildings, cars and people in (a), (c) and (d) in Figure 9. While the drawback is that this method generates disordered surfaces on irregular normal regions, for instance, the trees of (b). Without priority, the proposed method works well in complex terrain like construction sites in (c) and the ramp in (d).

In term of the time efficiency, Table I shows the time cost of each step. All results are the average value of 10 successive frames in each scene, and 50 times running on each frame. Frame streams are chosen randomly. And the experiment parameters voxel resolution and seed resolution are respectively 0.2 m and 1.0 m. In preparation stage, the mainly cost is normal estimation. The expanding phase costs a lot, and there is some room to improve. Specifically, expanding of seeds compute all the k nearest neighbours, in which the distance between inner voxels and centers are computed for many times. As the discreteness of sparse point clouds, we cannot obtain outer layer neighbour voxels directly by adjacency graph. Finally, the sharp feature extraction is fast. From these results we can find that the proposed method is robust to cope with different and difficult terrains with a nearly real-time computation speed.

Assert that 2.5D surfaces can be projected to 2D plane. Base on the Euler Characteristic, for a continuous Delau-

TABLE I. Experiment results: input size, segmented supervoxels, output sharp supervoxels and time costs

Scene	Points	voxles	supervoxels	sharp SV	time(s)			
					prepare	expand	sharp extract	total
1	64480	22876	6235	1276	0.2762	0.0404	0.6526	0.9692
2	66752	23692	7930	3302	0.2748	0.0530	0.7275	1.0554
3	67008	23337	7513	1502	0.2710	0.0494	0.7323	1.0528
4	65600	25013	7955	1723	0.2928	0.0522	0.7557	1.1101



(a) scene 1 (b) scene 2 (c) scene 3 (d) scene 4

Fig. 9. Campus datasets captured by Velodyne® HDL-32E

nay triangulation, the ratio of faces and vertexes should be near to 2. Obviously, the generated surfaces in Figure 7 contain holes and disconnections due to the high sparsity of the dataset, but the ratio values are still meaningful to evaluate the reconstructed surfaces. See Figure 8, the blue line shows that this value is stable in different complex scenes, which indicates the approach works well even with different amount of complex obstacles around. While it is possible to improve this ratio and generate lager surface meshes in each frame by setting a larger resolution, but as mentioned in section 3, unreliable connections may miss small obstacles on the ground. The orange line for face/edge shows the useful connections. As the special rotation property, Velodyne® datasets generate many big ring shape point clouds, which results in many unnecessary connections. Take measures to improve this ratio can improve algorithm efficiency.

VI. Conclusions

In this paper, an efficient surface reconstruction method for Velodyne® scanner is introduced. Supervoxels are used to make voronoi polygons and generate Delaunay triangular meshes. Local planes are estimated by comparing vertexes' normals, so that the proposed method can be used in both man made highway and wild uneven terrain. The sharp features are extracted in segmented clusters and used to detect obstacles without any prior knowledge. Experimental results show that the final surface generated from sharp feature is reliable and nearly real-time efficient. Future work is to deal with local meshes overlapping and explore suitable frame-to-frame registration method for real-time wild exploring robots and automatic driving vehicles applications.

References

[1] Grant W S, Voorhies R C, Itti L. Finding planes in LiDAR point clouds for real-time registration[C]//Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on. IEEE, 2013: 4347-4354.

[2] Papon J, Abramov A, Schoeler M, et al. "Voxel cloud connectivity segmentation-supervoxels for point clouds," Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on. IEEE, 2013: 2027-2034.

[3] Marton Z C, Rusu R B, Beetz M. "On fast surface reconstruction methods for large and noisy point clouds," Robotics and Automation, 2009. ICRA'09. IEEE International Conference on. IEEE, 2009: 3218-3223.

[4] Steinbrucker F, Kerl C, Cremers D. "Large-Scale Multi-Resolution Surface Reconstruction from RGB-D Sequences," Computer Vision (ICCV), 2013 IEEE International Conference on. IEEE, 2013: 3264-3271.

[5] Gopi M, Krishnan S, Silva C T. "Surface reconstruction based on lower dimensional localized Delaunay triangulation," Computer Graphics Forum. Blackwell Publishers Ltd, 2000, 19(3): 467-478.

[6] Bernardini F, Mittleman J, Rushmeier H, et al. "The ball-pivoting algorithm for surface reconstruction." Visualization and Computer Graphics, IEEE Transactions on, 1999, 5(4): 349-359.

[7] Hoppe H, DeRose T, Duchamp T, et al. "Surface reconstruction from unorganized points,". ACM, 1992.

[8] Rusu R B. "Semantic 3D object maps for everyday manipulation in human living environments," KI-Knstliche Intelligenz, 2010, 24(4): 345-348.

[9] Boulch A, Marlet R. "Fast and robust normal estimation for point clouds with sharp features," Computer Graphics Forum. Blackwell Publishing Ltd, 2012, 31(5): 1765-1774.

[10] A. Moore, S. Prince, J. Warrell, U. Mohammed, and G. Jones. "Superpixel lattices," Computer Vision and Pattern Recognition, 2008 (CVPR). IEEE Conference on, pages 18, june 2008. 2

[11] Ioannou Y, Taati B, Harrap R, et al. "Difference of normals as a multi-scale operator in unorganized point clouds,"3D Imaging, Modeling, Processing, Visualization and Transmission (3DIMPVT), 2012 Second International Conference on. IEEE, 2012: 501-508.

[12] Douillard B, Underwood J, Kuntz N, et al. On the segmentation of 3D LIDAR point clouds[C]//Robotics and Automation (ICRA), 2011 IEEE International Conference on. IEEE, 2011: 2798-2805.

[13] Pfaff P, Triebel R, Burgard W. An efficient extension to elevation maps for outdoor terrain mapping and loop closing[J]. The International Journal of Robotics Research, 2007, 26(2): 217-230.

[14] Moosmann F, Stiller C. Velodyne slam[C]//Intelligent Vehicles Symposium (IV), 2011 IEEE. IEEE, 2011: 393-398.

[15] Zhang L, Li Q, Li M, et al. Multiple Vehicle-like Target Tracking Based on the Velodyne LiDAR[J]. Proc. of IFAC Intelligent Autonomous Vehicles, 2013: 126-131.