

Fast Surface Reconstruction with Supervoxels and Sharp Features

Tianwei Zhang*
University of Tokyo
Tokyo, Japan

Yoshihiko Nakamura †
University of Tokyo
Tokyo, Japan

Abstract— *This paper proposes a fast surface reconstruction method by using supervoxels. The Point Cloud Data (PCD) acquired from depth sensors is divided into small volumes, and volumes containing sharp features are extracted after filtering isolated points. Then, the clustered supervoxels are used as input to generate triangle meshes of surface maps. No prior knowledge and color data are used in this method, so that the proposed method is suitable for both indoor and outdoor unknown environments. Experiments in large and noisy PCD show that the reconstruction method is reliable and fast.*

Keywords: surface reconstruction, supervoxel, sharp feature extraction

I. Introduction

People always expect that robots can totally take over special occasions working, such as outer space exploring, nuclear power station working and mine rescuing. To deal with such complex unknown environments, a robot needs efficient environment perception abilities. Therefore, full 3D environment perception or detection is a really meaningful topic in the field of robotics. With the development of depth sensors, such as Microsoft Kinect and laser scanners, environment informations containing both color and depth can be rapidly obtained from the output point clouds.

Depth sensors are important tools to percept workspace for robotics and industry applications. Recent years, researchers develop a lot different environment perception methods even works well in large and noisy point clouds [2]. The mostly way to understand unknown environment is to create a respective triangular mesh surface for representation of the given PCD. There are two big families of surface reconstruction methods: point interpolation methods and sampling approximating methods. For instance in [3], surface representation is obtained by realizing Delaunay triangulation through interpolation points into input point clouds. Variant methods like ball-pivoting method [4] and [1], in which incrementally growing triangles with local strategies to handle point inserting.

As interpolation methods use input points to form the final surface, the performance is highly sensitive to input noise, a smoothing filter is necessary. However, the incidental iterative computation results inefficient performance

in real time systems. Recent works like [5], which employs RANdom SAmple Consensus method to compute local low-degree surface's Delaunay triangulation meshes.

In other respect, approximation based methods are widely studied. they save a lot of time in points sampling by dividing the point clouds into small regular grids, and these grids, usually called voxels, are used to represent the real surface and extract the final triangle meshes [6]. The drawback of these methods is that, to indicate exact local information the knowledge of normal of each input point is necessary, therefore, method of normals estimation with their neighbourhoods were explored in [7]. As the normal clouds are approximated by K-nears neighbours, the result normal clouds is smooth without sharp features. To achieve sharp normals at the boundary regions, [8] proposed a randomized hough transform accumulator to decide the probability distribution of possible normals.

Recently, a group of approximation based methods expand the concept of superpixel to Supervoxels [9]. In 2D image processing, people use approximate sampling to decrease input pixel size, and follow this idea, supervoxel is a approximated sample for 3D point clouds data which can decrease the input PCD size from millions to hundreds. This concept is studied in application of 3D image segmentation [10] and 3D video segmentation [15].

In this paper, input point clouds are divided into small voxels with a setted resolution as container. The voxels containing isolated points will be removed as noise. And the sharp voxels are labelled by sharp feature extraction. Then, a larger resolution provides bigger seeding voxels for supervoxels clustering with a defined distance. After the adjacency searching, a Voronoi diagram is generated. Finally the Delaunay triangular meshes are generated by connecting the sites of voronoi diagram. The details will be introduced in following sections.

The main contributions of this paper are:

(i) A newly Delaunay triangular method is proposed. In our fast surface reconstruction method, triangular meshes are generated from constructed Voronoi polygons. The meshes generation is fast and well distributed.

(ii) We proposed an strategy to generate small sharp supervoxels and big platform supervoxels. We use sharp features to identify sharp voxels, this method is flexible to generate big meshes in planner surfaces and small meshes in edges and boundaries regions.

(iii) No prior knowledge is used, the proposed method

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

†nakamura@ynl.t.u-tokyo.ac.jp

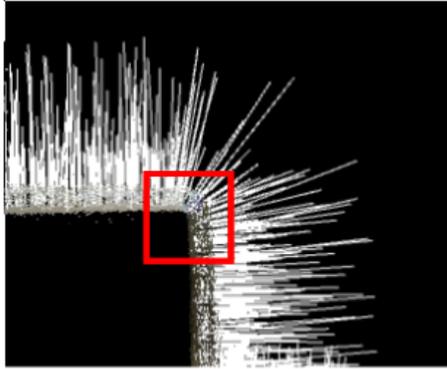


Fig. 1. Side view of the normal cloud near a corner voxel(red)

can be applied in unknown environments.

The following of this paper is organised as follows. Section II introduces the sharp feature extracting in voxel scale. Supervoxels clustering and triangulation processing are presented in section III. Section IV includes experiment and evolution. The last part is conclusion.

II. SHARP FEATURE

A sharp feature estimation method 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 between two results is used to indicate if the query point is sharp or not. However, as it is shown in Figure II, the normals of sharp edge points have almost Symmetrical neighbour normals so that the DoN is small on the corner but bigger on both sides of the corner. What's more, the normals of whole point clouds have been computed twice with the complex computing method introduced in [8], which make the normals cloud sharper but cost more computing time. To extract the sharp points efficiently, we consider about the length of the sum normals inside each unit voxel:

$$L_n = \left| \sum \vec{n}_i \right|, \vec{n}_i \in \mathcal{V} \quad (1)$$

Look at the simplest situation in figure II, there are two points in a voxel locating in corner or edge places. Obviously, the length of the sum of the normals (black arrow) should be shorter than two unit length. We set a threshold t , and if L_n is smaller than $t * N$ (N is the number of points inside the voxel), this voxel will be labelled as sharp voxel.

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 from [7], in which the normals are computed by analysis of the eigenvectors and eigenvalues of a covariance matrix created from the k -near neighbours of the query point. The neighbour size k determines the quality of the normal clouds. Theoretically, the bigger the k is, the bigger the t

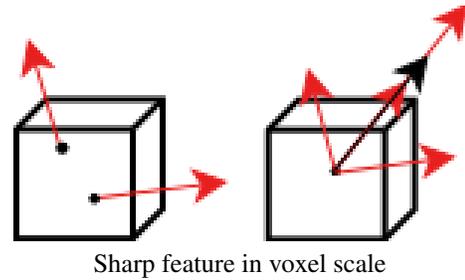


Fig. 2. Left: two black points are points inside this voxel, and red arrows are the unit normals of them. Right: the black arrow is the sum of normals inside of this voxel, the red arrows show the unit length of each normal.

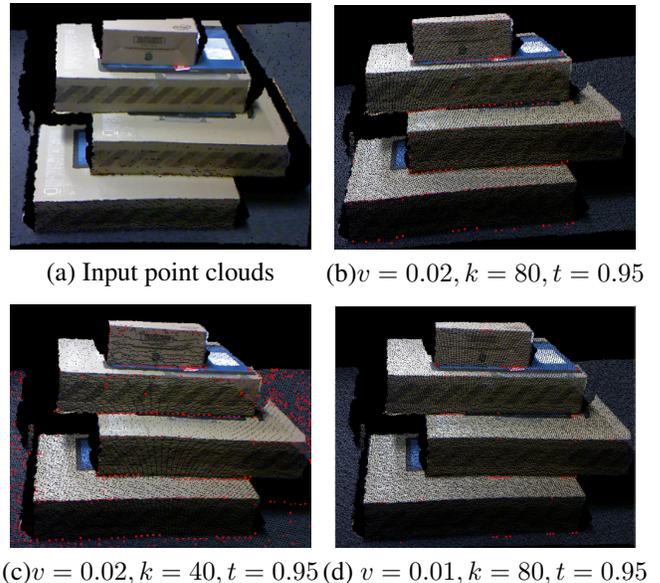


Fig. 3. Sharp feature extracting, (a) shows is input, (b), (c) and (d) show extracted sharp points in red under different parameters

is, because the edge points influence much on neighbour points and make their normals sloped. Obviously, k should be big enough to indicate local structure, but it results in extra computational costs. On the other hand, the resolution of unit voxel v also has a big influence on t . Bigger voxel contains more points and results in smaller t . Figure 3 shows the extracted sharp point clouds which are coloured in red with different values of t, k, v .

III. SUPERVOXEL CLUSTERING

A. Distance Measure

In [10], to identify obstacle boundaries, their supervoxels clustering distance measurement contains spatial distance D_s , color distance D_c and Fast Point Feature Histograms(FPFH) feature space distance. In case of surface construction, color information is not necessary, while the normals are important to indicate local surface information.

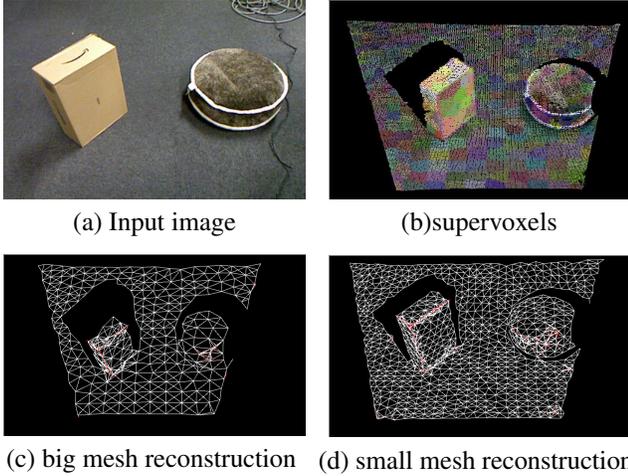


Fig. 4. Supervoxels clustering, (a) original scene, (b) supervoxels in different colors (c) surface reconstruction with $R_{seed} = 0.1$ and (d) surface reconstruction with $R_{seed} = 0.05$

Thus, in this proposed method, distance measure is defined as:

$$D = \sqrt{\frac{\mu D_s^2}{3R_{seed}^2} + \theta D_n^2} \quad (2)$$

where the μ and θ are the influence factor of spatial distance and normal value. R_{seed} is the seed resolution, limits the searching depth of each cluster. Normal distance D_n is computed as

$$D_n = 1 - \vec{n}_s \cdot \vec{n}_v \quad (3)$$

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.

B. Clustering

The clustering is a k-means similarity clustering, where the space is divided into big search volume with resolution R_{seed} , and the center points of each seed volume are initial seeds. Note that the sharp point clouds generated from last step are divided with a smaller resolution R_{seed}^i , and their center points are treated as seeds, too. Start from the voxel nearest to the seeds of each volume, and compute the distance between the query voxel to the seeds though equation (2), add this voxel into its nearest supervoxel, and its neighbours will be added to the searching queue of the owner supervoxel. This iteration keeps on going until the boundary of that volume or there is no neighbour left.

Each supervoxel expands at the same rate, and the center of supervoxels are updated as the mean of all the members after each each expansion. 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 belong to that supervoxel. Then, as the dual graph of Voronoi map, Delaunay triangular meshes can be achieved by connecting the center points of adjacency supervoxels. Note that the supervoxels of sharp point clouds will be smaller than the others, resulting in small triangular meshes in sharp places. Figure 4 indicates the clustering result and reconstruction surfaces with two different seed resolution.

IV. EXPERIMENTS AND EVALUATION

A. Dataset

To evaluate the quality of our method, we compare with the state of art method [1] which is also for large noisy point clouds. In experiments, we use the recently created RGB-D Scenes Dataset v2 made by [12]. This dataset consists of 14 scenes containing furniture (chair, table, sofa) and some objects (bowls, caps, cereal boxes, coffee mugs, and soda cans). Each scene is a point cloud created by aligning a set of video frames. The machine used for our experiments is an eight-core Intel(R) Xeon(R) @ 2.66 GHz with 16 GB system memory.

B. Results

A set of screen shots are given in Figure 5, the first column are the input point clouds, and the parameters are given in Table I, in which, FSR is the method of [1]. All the inputs point clouds are large noisy clouds with around one million points. SVSR is short for SuperVoxels Surface Reconstruction (the proposed method). FSR is short for Fast Surface Reconstruction from [1]. Seed resolution which decide the triangle's side length is 0.1 m in SVSR. Voxel resolutions are shown in the table. At the same resolution, FSR generated a large number of vertices, so a bigger resolution, 0.04 m, is also shown in the table. Results data are averaged over 100 experiments.

C. Evaluation

To evaluating the resulting triangular surfaces, we compare the number of triangles and vertices. As the point clouds are 2.5D data (captured from only one camera view, so that no full 3D modules can be made), assert that the 2.5D surfaces are projected to a 2D plane. From Euler Characteristic, in a continuous Delaunay triangulation, F (facets) is smaller than $2V - 2 - B$, in which V is vertices and B is the number of convex hull vertices. Obviously, the more the F approaching to $2 \times V$, the smaller B is, which means less holes and disconnections in the final surfaces. Figure 7 shows that the proposed method has higher F/V which indicates better surface quality. This result can be seen in table scene in Figure 6.

For the time efficiency, see Figure 8, FSR method is faster with a higher resolution. However, when the resolution is lower, the time cost increases greatly. SVSR shows a steady time cost around $7\mu s$ each point with large size PCD,

TABLE I. Experiments Data, SVSR is the proposed method

Scene	Date Size	SVSR (resolution 0.02 m)			FSR (resolution 0.04 m)			FSR (resolution 0.02 m)		
		Vertices	Triangles	Time	Vertices	Triangles	Time	Vertices	Triangles	Time(s)
1	1,249,801	5139	11690	8.39	33055	44270	5.63	130857	168203	23.47
2	907,839	3917	8535	6.65	27823	31572	4.74	112805	149439	19.91
3	1,053,410	5254	11957	8.29	33617	44165	5.69	133996	178539	23.98
4	737,587	1315	2590	2.56	8490	11046	1.46	33821	48311	5.87

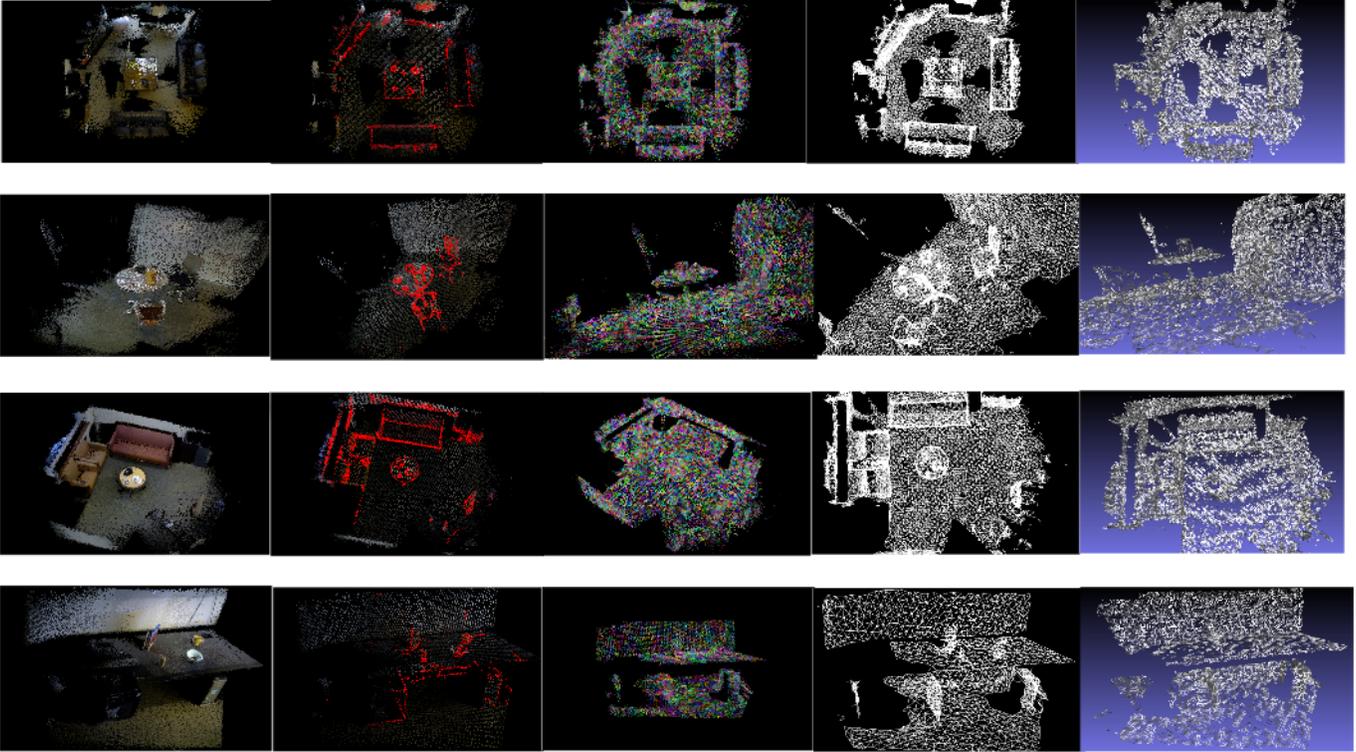


Fig. 5. First column is the input PCD files; Second column shows the extracted sharp voxels which color red; Third column is the generated supervoxels, in which each color instead for one supervoxel; Forth column is the reconstructed surfaces; Last column is the resulted surfaces from FSR method

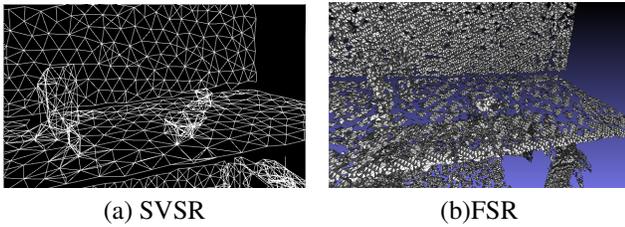


Fig. 6. Zooming in of table scene, (a) shows the final surfaces of the proposed SVSR method, obviously, meshes are well distributed and no holes on the table platforms, while FSR's result (b) show that there are many holes in the table plane even if the mesh size is small enough

and around 2 frames per second with kinect PCD inputs, which is efficient enough for real time robot motion planning. The time cost of ranges because the different structure of the four experiment scenes. SVSR can save time cost on platform regions by using big supervoxels, and construct

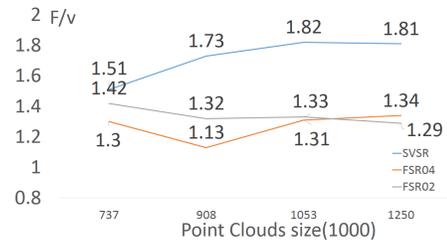


Fig. 7. Proportion of Facets and Vertices

small triangles at sharp edges to indicate boundary features for robots.

V. CONCLUSION

We proposed a fast surface reconstruction method for large noisy point clouds data in this paper. Input point

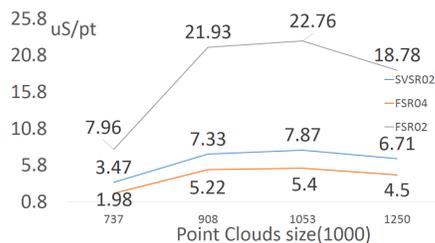


Fig. 8. Computation time for per point

clouds are presented by Supervoxels which have big volumes. Delaunay triangular meshes are newly generated by connecting center samples of supervoxels. To indicate local information, sharp features of point clouds are extracted in voxel scale. Additionally, smaller meshes are generated in edge regions to illustrate the accurate edges of original surfaces. Experimental results on large noisy indoor scenes show that the proposed method is fast and reliable for robot motion planning. There are two further tasks remain, one is to optimal normals estimation to improve real-time processing performance. Another one is to apply this method into robotics application such as unknown environments motion planning, ladder and stair climbing.

References

- [1] 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.
- [2] 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.
- [3] 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.
- [4] 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.
- [5] Ricard C, Rafael G, Pierre A, and Mariette Y. "A surface reconstruction method for in-detail underwater 3D optical mapping," The International Journal of Robotics Research(IJRR), January 2015 34: 64-89, doi:10.1177/0278364914544531
- [6] Hoppe H, DeRose T, Duchamp T, et al. "Surface reconstruction from unorganized points,". ACM, 1992.
- [7] Rusu R B. "Semantic 3D object maps for everyday manipulation in human living environments," KI-Kunstliche Intelligenz, 2010, 24(4): 345-348.
- [8] 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.
- [9] A. Moore, S. Prince, J. Warrell, U. Mohammed, and G. Jones. "Supervoxel lattices," Computer Vision and Pattern Recognition (CVPR). IEEE Conference on, pages 18, june 2008. 2
- [10] 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.
- [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] Kevin Lai, Liefeng Bo, and Dieter Fox, "Unsupervised Feature Learning for 3D Scene Labeling," IEEE International Conference on Robotics and Automation (ICRA), May 2014.
- [13] Amenta N, Choi S, Dey T K, et al. "A simple algorithm for homeomorphic surface reconstruction," Proceedings of the sixteenth annual symposium on Computational geometry. ACM, 2000: 213-222.
- [14] Amenta N, Choi S, Kolluri R K. "The power crust, unions of balls, and the medial axis transform," Computational Geometry, 2001, 19(2): 127-153.
- [15] Weikersdorfer D, Schick A, Cremers D. "Depth-adaptive supervoxels for RGB-D video segmentation," ICIIP. 2013: 2708-2712.