Title:
An Efficient Feature Point Extraction Algorithm for Noisy Point Cloud

Authors:
NanHua Huang, hnh0774@163.com, Guangxi Normal University
Ming Chen*， hustcm@hotmail.com, Guangxi Normal University
ZhengQin Zhang, 1565493549@qq.com, Guangxi Normal University
Shenglian Lu, lsl@gxnu.edu.cn, Guangxi Normal University

Keywords:
Feature Extraction, Feature Detection, Point Cloud Data Processing, Reverse Engineering

DOI: 10.14733/cadconfP.2022.267-270

Introduction

Point cloud data is more and more widely used in reverse engineering, cultural relic restoration, architecture and many other fields. [4,5]. In practice, the obtained data is often non-uniformly sampled and of noises. Feature exaction is a key step for the subsequent processing of point clouds such as matching, segmentation and recognition. How to identify point features for noisy point cloud and improve the efficiency are challenging at present.

In previous studies, Nie*et al.* [2] used a surface smooth shrinkage index (SSI) to measure the degree of surface change, and judged feature points according to the absolute value of SSI of each point. This method has a good anti-noise ability and can extract sharp feature points, but it cannot work for smooth features. In practical applications, we find that the regions of smooth features (such as fillets) have a higher density than other non-featured places, as these interested regions are usually scanned multiple times or the scanning orientation is adjusted to obtain a relatively larger scanning point density at these places. Considering this fact, a combined index of density and SSI is proposed so as to recognize smooth features, and the recognition is also accelerated through octree data structure.

Main Idea

Surface variation

Surface variation $L(i)$ is calculated by the method in the literature [2]. More explicitly, $L(i)$ is the projected distance along the normal vector between the current point p_i and its neighboring gravity points. In order to enhance the anti-noise performance, the average coordinate of p_i's neighboring points is in place of p_i. $L(i)$ is evaluated as below:

$$L(i) = ||(\overline{p_i} - \overline{p_i^w}) \bullet n_i|| \tag{1}$$

Where n_i is the normal vector of $\overline{p_i}$, $\overline{p_i}$ and $\overline{p_i^w}$ are calculated by Eq. (2):

$$\overline{p_i} = \frac{\sum_{j=1}^{k} m(i)p_j}{\sum_{j=1}^{k} m(i)}$$

$$\overline{p_i^w} = \frac{\sum_{j=1}^{n} g(i)p_j}{\sum_{j=1}^{n} g(i)} \tag{2}$$

Where $\{p_j\}$ is the set of neighborhood points of p_i, $m(i)$ is the average distance r-neighborhood of p_j and $g(i)$ is the Gaussian weight, which is calculated as below:

$$g(i) = e^{-\left(\frac{||p_j - p_i||}{r}\right)^2}$$

Where *r* is preset search radius. The larger the value of *L*(*i*) is, the larger the surface change at the point p_i, and the more likely the point is to be a feature point. The sharp feature points can be recognized by this method. However, it cannot recognize the smooth features.

Density

As mentioned before, a feature region has a larger local density in practical applications, the density index *D*(*i*) is calculated using the following formula:

$$D(i) = \frac{1}{\left(\frac{1}{m}\sum_{j=1}^{m} ||p_i - p_j||\right)} \tag{3}$$

Where p_j is the neighboring point of p_i, and *m* is the *m* nearest points of p_i. As some $||p_i - p_j||$have been calculated when calculating SSI, it will not be computed for the acceleration consideration.

The Combined index

The combined index *F*(*i*) is proposed in this paper, in which *D*(*i*) and *L*(*i*) are combined to let the algorithm recognize both sharp features and smooth features. The final feature determination index in this paper is calculated by Eq.(4):

$$F(i) = \alpha L(i) + (1-\alpha)D(i) \tag{4}$$

Where α is the adjusting weight. When the noise is large and the shape change is significant, α should be relatively larger, otherwise a smaller value will be assigned.

Voxel Acceleration

F(*i*) requires heavy computing when the point data is large, so the well-known voxel method is adopted to reduce the calculation scale. In the paper, a rough calculation will be first performed to exclude the points that are obviously not feature ones, and the left points will be next judged. A voxel data structure is constructed to subdivide the point data with the lattice edge length being the average distance between each pair of neighboring points. Once the number of points in one lattice is larger than that of any other neighboring lattices by a threshold, all points in the lattice together with its neighbors will be taken as candidate points for next judgment. After this treatment, the number of invalid points for next judgment by Eq.(4) will be significantly reduced (see Fig.1) and the computing cost will be reduced.

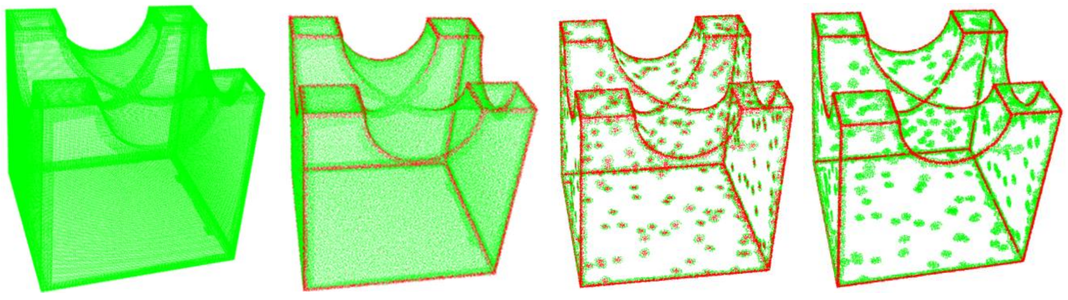

Fig. 1: (a) The input model with nearly 50% noisy points; (b) directly processing all points using Eq. (4), it will take 4.839 second; (c) use voxel acceleration strategy, the many points will be filtered out and the Eq. (4) will be calculated on these left points (all points will be red colored as candidate feature points); (d) the final result, which will take 3.196 second and the whole algorithm will be sped up by 33.95%.

Results

The algorithm is implemented by Visual Studio C++ 2019, using C++, and tested by the PC with 2.40GHz AMD processor, 8G DDR4 memory. The classical methods including PCA[3], SSI[2] and the method in [6] are adopted as benchmarks. In order to compare the accuracy of feature extraction, an evaluation index P_r is proposed in the paper, which is defined as follows:

$$P_r = N_e / N_g \tag{5}$$

Where N_e is the number of exacted feature points and N_g is the ground truth.

Two noisy point cloud models are tested with the proposed algorithm and the benchmarks, the results are shown in Figure2 and Table 1.

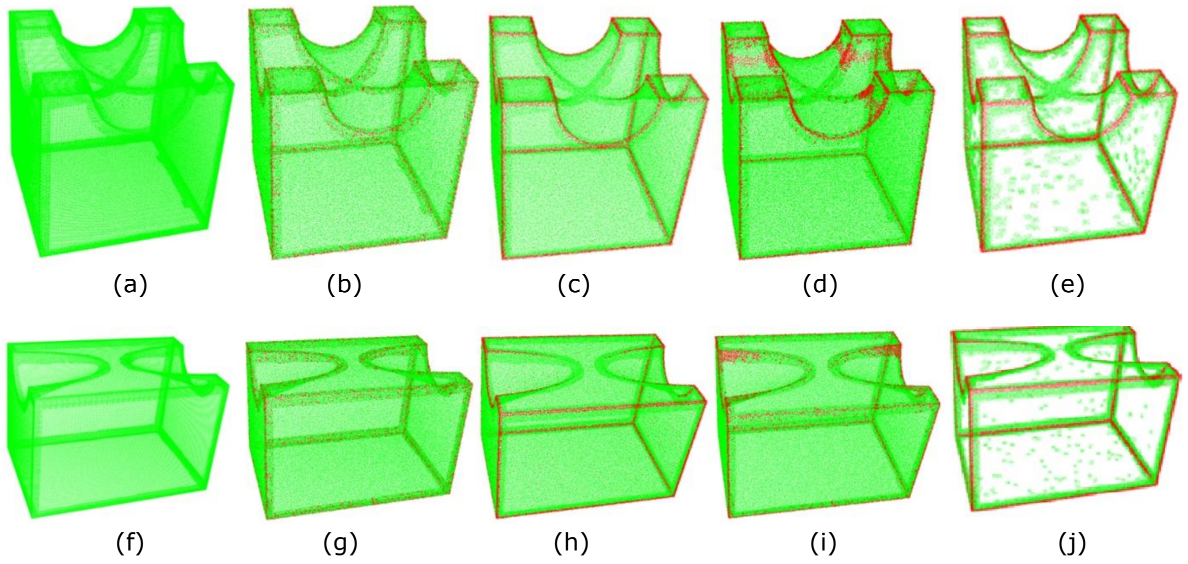

Fig. 2: The results of Model I and Model II by different methods: (a)The input Model I; (b) The result by PCA; (c) The result by SSI; (d) The result by the method in [6]; (e) The result by the proposed algorithm; (f) The input Model II; (g) The result by PCA; (h) The result by SSI; (i) The result by the method in [6]; (j) The result by the proposed algorithm.

Test model	*Methods*	*Point Number*	*P_r (%)*	*Time (sec.)*
Model I	PCA	170K	61.16	1.637
	SSI		86.32	4.479
	The method in [6]		86.78	1.961
	Ours		91.41	3.196
Model II	PCA	220K	57.58	2.128
	SSI		84.46	5.744
	The method in [6]		85.24	2.468
	Ours		90.16	3.932

Tab. 1: The statistics data on the test models, i.e., Model I and Model II.

Through Fig.2, it can be found that PCA and the method in[6] are both sensitive to noises and some noises are mistakenly regarded as feature points. SSI method has an anti-noise ability and sharp features at the upper and middle and bottom edges can be detected. The proposed algorithm is efficient and the feature recognition ability are the best with the highest P_r.

.

Conclusions

One novel algorithm is proposed in this paper to detect feature points for scanned point cloud data of noises. This method combines the density index and SSI index and use the voxel data structure for acceleration. The classical method including PCA, SSI and the method in [6] are adopted as benchmarks, and the comparing results show that it has a good anti- noise ability and the recognition ability is the best. In the future work, efforts will be taken to how to connect the feature points to be parametric features.

References:

[1] Li Q, Huang X, Li S, et al.: Feature extraction from point clouds for rigid aircraft part inspection usi ng an improved Harris algorithm, Measurement Science and Technology, 2018, 29(11): 115202. htt ps://doi.org/10.1088/1361-6501/aadff6

[2] Nie J.: Extracting feature lines from point clouds based on smooth shrink and iterative thinning. Graphical Models, 2016, 84: 38-49. https://doi.org/10.1016/j.gmod.2016.04.001

[3] Pauly M, Keiser R, Gross M.: Multi - scale feature extraction on point - sampled surfaces, Computer graphics forum, Oxford, UK: Blackwell Publishing, Inc, 2003, 22(3): 281-289. https://doi.org/10.111 1/1467-8659.00675

[4] X Lai, et al.: A Building Extraction Approach Based on the Fusion of LiDAR Point Cloud and Elevatio n Map Texture Features, Remote Sensing 11.14(2019):1636-. https://doi.org/10.3390/rs11141636

[5] Yi, Y. A. et al.:Three-dimensional point cloud data subtle feature extraction algorithm for laser sca nning measurement of large-scale irregular surface in reverse engineering –ScienceDirect, Measure ment 151,(2020). https://doi.org/10.1016/j.measurement.2019.107220

[6] Wang, L.; Yuan, B.: Curvature and density-based feature point detection for point cloud data, IET International Conference on Wireless. IET, 2011. https://doi.org/10.1049/cp.2010.0694