

**Title:****Fast Voxel-based Surface Propagation Method for Outlier Removal****Authors:**Yu Chen, [cheny39@rpi.edu](mailto:cheny39@rpi.edu), Rensselaer Polytechnic Institute, U.S.A.Hao Chen, [pschenhao@sues.edu.cn](mailto:pschenhao@sues.edu.cn), Shanghai University of Engineering Science, ChinaJie Shen, [shen@umich.edu](mailto:shen@umich.edu), University of Michigan-Dearborn, U.S.A.**Keywords:**

Surface Propagation, Laser Scanning, Data Outlier, Discrete Data Point

DOI: 10.14733/cadconfP.2016.284-289

**Introduction:**

Outlier removal is an important problem in modern society. Related techniques have been widely used in a variety of fields: reverse engineering, rapid prototyping, biomedicine, architecture, entertainment industry, financial data analysis, etc.

Many studies have been conducted to remove outliers effectively and efficiently. However, most of these methods perform well only in specific situations. Some methods [1, 2] are more suited to handling isolated outliers, and others [3] are good in dealing with surfaces without sharp features. Shen, et al. [4] proposed a surface propagation method combined with minimal variance and normalized histogram which performs well in both non-isolated outlier clusters and sharp edges. But, Shen's method encounters an efficiency issue when dealing with large-scale data models. Wang and Feng [5] further advanced Shen's method by introducing a new method: majority voting.

According to our analysis, non-isolated outlier clusters and sharp featured outlier clusters are two types of the most difficult outlier clusters. As for non-isolated outlier clusters, outlier clusters are so close to a main surface that distance-based criteria are not effective to remove outliers. With respect to sharp featured outlier clusters, it is difficult to preserve these sharp features in a data model because geometric non-smoothness at these features invalidates many analysis arsenals in calculus and differential geometry. In some cases such as laser scanning data, we might have to handle surfaces with sharp features and non-isolated outlier clusters. This becomes the most difficult type of models. The main objective of this paper is to propose a fast algorithm to remove different types of clusters, especially the most difficult types of outlier clusters, in an accurate and efficient way.

**Main Idea:**

The domain of a point cloud model is subdivided into a limited number of voxels (volume elements) through a uniform partition in a 3D space. We use a voxel as the minimal unit of analysis and display. Specifically, all the numerical analyses are aimed at each single voxel, and all the data points in a voxel are treated simultaneously as an outlier or true data point. We adopt this approach for three reasons:

- (a) Principle of locality: as for the tasks of surface outlier removal, data points in a small voxel are highly likely to have similar properties.
- (b) A reasonable partition of voxels provides us an effective way to analysis data clusters statistically and locally.
- (c) It is cheap to partition data clusters into voxels and work on them.

The propagation is based on voxel and starts from an initial voxel. Then, it looks around the neighboring ring of the current voxel, select the voxels that are considered to be inside the main surface, and mark them as true data according to driving force for propagation. Herein, the driving force is defined as a mechanism to guide the surface propagation. Other voxels in the ring are

Proceedings of CAD'16, Vancouver, Canada, June 27-29, 2016, 284-289

© 2016 CAD Solutions, LLC, <http://www.cad-conference.net>

temporarily marked as outliers. For each voxel that is inside the main surface, the program looks around and repeats the above procedure. The propagation terminates until we have no new voxel to visit. After the propagation, the program treats each data point in the voxels, which are marked as outliers or untouched (initial status), as outlier; each data point in the voxels that are marked as true data as a true data point.

Notice that because some voxels will be visited several times, there are different tricks to take in the propagation. Here, we finally mark the voxel as true data as long as it has been treated as true data for at least one time. But, with respect to those voxels that are treated as true data after being treated as outliers, they are not included into next iteration of propagation.

The driving force is crucial for surface propagation. It is used to determine whether a voxel should be included into the main surface. The key principles to select the driving force for surface propagation include

- (a) The driving force should be direct or indirect measurements or statistics,
- (b) The driving force should imply the levels of intimacy between voxels,
- (c) The driving force should imply the difference between true data and outliers.

We define two kinds of driving forces in our method: distance and angle. As for distance, it refers to the Euclidean distance between the centroid of data points in a neighboring voxel and the plane determined by the two largest eigenvectors of the current voxel. It is a priori assumed that the smaller the distance is, the larger the probability, with which the neighboring voxel is actually inside the main surface, is. In other words, outliers have larger distances than true data points.

With respect to angle, it represents an angle between two planes each of which is determined by the two largest eigenvectors of the current and neighboring voxels, respectively. It is a priori assumed that the smaller the angle is, the larger the probability, with which the neighboring voxel is actually inside the main surface, is. In most cases, outliers have larger angles than true data points.

Notice that each single driving force mentioned above is not effective enough to handle all cases. For example, in a scenario of removing non-isolated surface outliers, which is regarded as the most difficult case, outliers are often quite close to the main surface. Thus, the distance approach is likely to fail. In other scenarios, the angle approach is not effective enough to differentiate true data from outliers. A comprehensive way is to combine these two criteria so that we can handle different cases.

The initial voxel is another crucial factor for surface propagation. If an incorrect initial voxel is selected, the propagation cannot help us remove outliers. The key principle to select an initial voxel of surface propagation is to find the voxel that is most likely to be inside the main surface. Here, we adapt connectivity flags and histogram bins of eigenvalues to determine a hlevel variable for each voxel which indicates the probability for a voxel to be inside the main surface. Then, the voxel with the highest probability is selected as an initial voxel.

A general flowchart of our fast voxel-based propagation (FVBSP) for outlier removal is illustrated in Figure 1. It consists of three main technical components: connectivity, eigenvalue, and core propagation. The detailed algorithm of FVBSP is provided in Figure 2, while two important routines are outlined in Figures 3 and 4. The eigenvalue of a covariance matrix is computed on the basis of codes in [6].



Fig. 1: A general flowchart of our fast voxel-based propagation for outlier removal.

#### Name: Fast Voxel-Based Surface Propagation

1. VoxelSet = GetVoxelSet(data\_points);
2. InitialVoxel = GetInitialVoxel(VoxelSet);
3. Add InitialVoxel in MainSurface;
4. Loop:
5. foreach current\_voxel in MainSurface:

6. if current\_voxel is marked as untouched or unprocessed:
7.     Mark current\_voxel as true\_data;
8.     Foreach neighboring\_voxel in GetNeighboringVoxels(current\_voxel):
9.         if DrivingForceJudgement(neighboring\_voxel, current\_voxel) == true:
10.             if neighboring\_voxel is marked as untouched:
11.                 Add neighboring\_voxel in FrontWave;
12.                 Mark neighboring\_voxel as unprocessed;
13.             else if neighboring\_voxel is marked as unprocessed or true\_data:
14.                 do nothing;
15.             else:
16.                 Mark neighboring\_voxel as true\_data; // It varies on specific cases
17.         else:
18.             if neighboring\_voxel is marked as untouched:
19.                 Mark neighboring\_voxel as outlier;
20.     if number of voxel in FrontWave is zero:
21.         Jump out of loop;
22.     Copy FrontWave to MainSurface;
23. All data points in voxel marked as true\_data are true data;
24. All data points in voxel marked as outliers or untouched are outliers;

Fig. 2: Algorithm of fast voxel-based surface propagation.

**Name: GetInitialVoxel Input: Voxel**

Set Output: initial voxel

1. foreach voxel in Voxel Set:
2.     if GetConnectivityFlag(voxel) is true:
3.         error = CalcSmallestEigenValue(voxel);
4.         hlevel = Bin(error);
5.     else:
6.         hlevel = MaxBinNum - 1;
7.     Select a single voxel with smallest hlevel as initial voxel;

Fig. 3: Routine of GetInitialVoxel Input.

**Name: DrivingForceJudgement**

Input: neighboring\_voxel & current\_voxel

1. Calculate eigenvectors for the two voxels;
2. Calculate the two planes for the two voxel;
3. Calculate the distance between the centroid of data points in neighboring\_voxel and plane in current\_voxel;
4. Calculate the included angle between the two planes;
5. DrivingForce = ratio\*Normalized(distance) + (1-ratio)\*Normalized(angle);
6. if DrivingForce < Threshold:
7.     return true;
8. else:
9.     return false;

Fig. 4: Routine for computing the driving force.

Compared with Shen's surface propagation method, some unique features of our FVBSP method include

- The FVBSP method is based on voxels instead of kd-tree. This makes it less expensive and more efficient than Shen's method.
- The FVBSP method combines not only the distance factor but also the angle factor, which makes it well-suited to different types of outlier clusters.
- The FVBSP method starts from one initial voxel and runs one round of propagation while Shen's surface propagation method starts from a number of initial points and runs several rounds of propagation. The experiment results show that as long as the driving force for surface propagation is effective and the initial voxel is good, one round of propagation is sufficient.

In summary, the FVBSP method simplifies the complexity of surface propagation while keeping a high performance of removing outliers which are difficult to deal with.

We compared eigenvalues method, connectivity method, Shen's surface propagation method and our FVBSP method on two data models. Both data model 1 (Fig. 1(a)) and data model 2 (Fig. 1(b)) have sharp edges and non-isolated surface outlier clusters that are extremely difficult to handle. In fact, data model 2 is more difficult than data model 1, because data model 2 has the non-isolated surface outlier clusters in the sharp edge area while data model 1 does not combine the two difficulties.

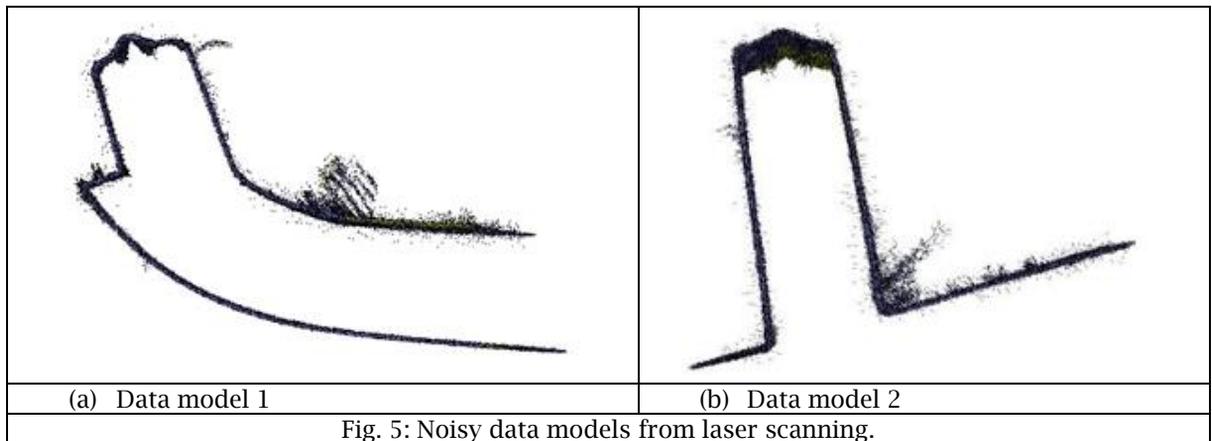
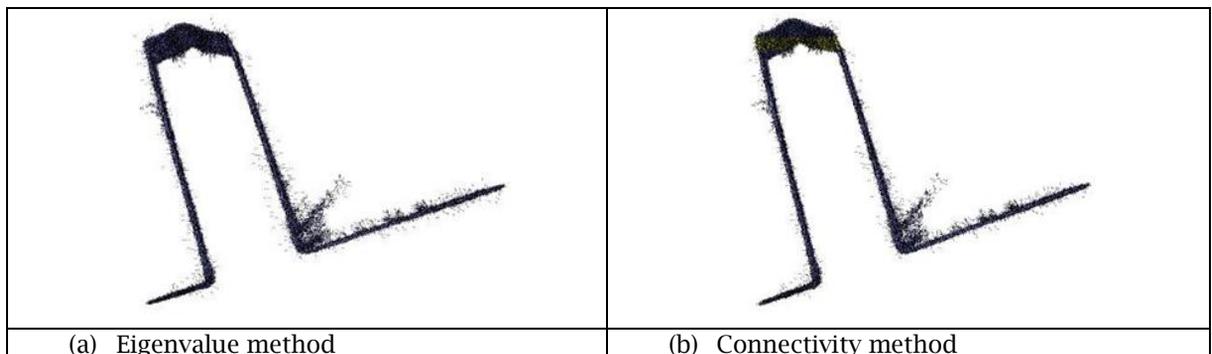


Fig. 5: Noisy data models from laser scanning.



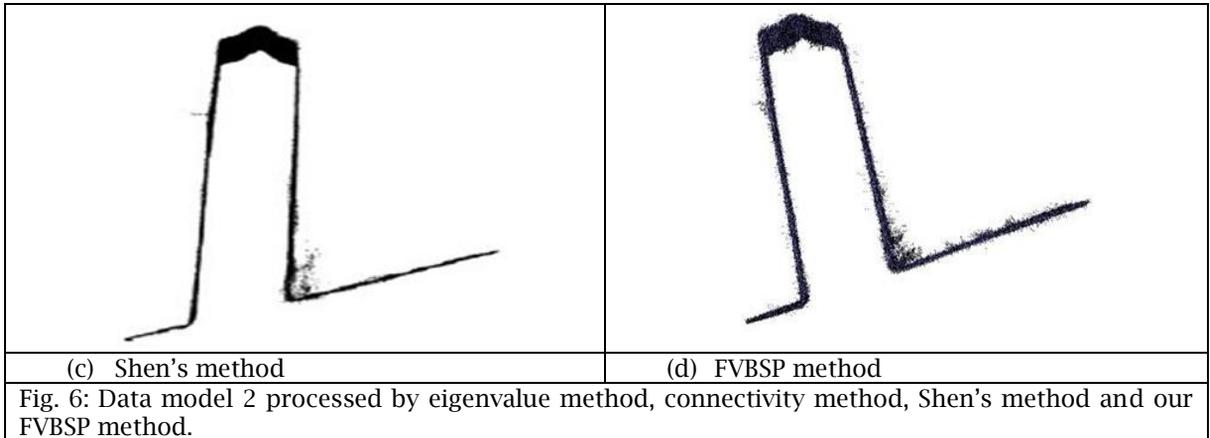


Fig. 6 is a typical case where non-isolated outlier clusters exist around a sharp corner. This case is more difficult than data model 1 since it combines non-isolated outlier clusters and sharp featured surface. Shen's method and our FVBSP method perform much better than the eigenvalue method and connectivity method.

Notice that the performance of our VBFSP method is very close to that of Shen's method, but the FVBSP method is much more efficient than Shen's method, as illustrated in Table 1. Since our propagation is voxel-based, its efficiency is close to the eigenvalue method and connectivity method. Shen's method is based on kd-tree that is much more time-consuming.

Method	# of reserved points	# of removed outliers	Execution time (sec)
Eigenvalue	89911	617	4.7
Connectivity	89642	886	3.4
Shen's method	89068	1460	34.0
FVBSP method	86176	4352	3.9

Tab. 1: Execution time and removed outlier count of data model 2.

#### Conclusions:

In summary, our algorithm is compared favorably to the existing methods because it can remove outliers even in the most difficult cases and it is not so time consuming compared with Shen's method that can also remove difficult outliers. The unique contribution of this paper is to propose a voxel-based surface propagation approach and a liner combination of distance and angle as the driving force of propagation that can deal with complex types of outlier clusters.

#### Acknowledges:

This work was in part supported by U.S. National Science Foundation DMI-0514900, CMMI-0721625, ECCS-1039563, and IIP-1445355.

#### References:

- [1] Fleishman, S.; Drori, I.; Cohen-Or, D.: Bilateral mesh denoising, ACM Transactions on Graphics, 22(3), 2003, 950-953. <http://dx.doi.org/10.1145/882262.882368>
- [2] Xie, H.; McDonnell, K.T.; Qin, H.: Surface reconstruction of noisy and defective data sets, IEEE Visualization, 2004, 259-266. <http://dx.doi.org/10.1109/VISUAL.2004.101>

- [3] Schall, O.; Belyaev, A.; Seidel, H.: Robust filtering of noisy scattered point data, Eurographics symposium on point-based graphics, 2005, 71-144. <http://dx.doi.org/10.1109/PBG.2005.194067>
- [4] Shen, J.; Yoon, D.; Shehu, D.; Chang, S.Y.: Spectral moving removal of non-isolated surface outlier clusters, Computer-Aided Design. 41(4), 2008, 256-267. <http://dx.doi.org/10.1016/j.cad.2008.09.003>
- [5] Wang, Y.; Feng, H.Y.: Outlier detection for scanned point clouds using majority voting, Computer-Aided Design, 62, 2015, 31-43. <http://dx.doi.org/10.1016/j.cad.2014.11.004>
- [6] Press, W. H.; Teukolsky, S.A.; Vetterling, W.T.; Flannery, B.P.: Numerical Recipes in C. Cambridge University Press. 1996.