# A New Region Growing Algorithm for Triangular Mesh Recovery from Scattered 3D Points

Chengjiang Long[1], Jianhui Zhao[1,*], Ravindra S. Goonetilleke[2], Shuping Xiong[3], Yihua Ding[1], Zhiyong Yuan[1], and Yuanyuan Zhang[1]

[1] Computer School, Wuhan University, Wuhan, Hubei, 430072, PR China
jianhuizhao@whu.edu.cn
[2] Dept. Industrial Engineering and Logistics Management, Hong Kong University of Science and Technology (HKUST), Hong Kong
[3] School of Design and Human Engineering, Ulsan National Institute of Science and Technology (UNIST), Ulsan, 689-798, Republic of Korea

**Abstract.** A novel region growing algorithm is proposed for triangular mesh recovery from scattered 3D points. In our method, the new principle is used to determine the seed triangle considering both maximum angle and minimum length; the open influence region is defined for the active edge under processing; positional element is added into the criterion to choose the most suitable active point; geometric integrity is maintained by analyzing different situations of the selected active point and their corresponding treatments. Our approach has been tested with various unorganized point clouds, and the experimental results proved its efficiency in both accuracy and speed. Compared with the existing similar techniques, our algorithm has the ability to recover triangular meshes while preserving better topological coherence with the original 3D points.

**Keywords:** surface recovery, triangular mesh, region growing, point cloud.

## 1 Introduction

Currently there are three typical approaches for surface reconstruction from scattered 3D points, and they are sculpting-based approach [1-3], implicit surface approach [4-6] and region growing approach [7-12]. The drawbacks of sculpting-based approach are its expensive computation and complex extraction. The limitation of implicit surface approach is that it only approximates rather than interpolates the scattered points, correspondingly it cannot guarantee the obtained surface passing through the original sample points. Different from the foregoing two approaches, region growing approach starts with a seed unit and then continues until all the points have been considered. Comparatively, the efficiency of this approach is very computationally high, thus has received more attentions.

The key technique of region growing approach lies in the selection of a point to form the new triangle with an active edge. In BPA algorithm [7], a sphere with user

* Corresponding author.

Z. Pan et al. (Eds.): Transactions on Edutainment VI, LNCS 6758, pp. 237–246, 2011.

specified radius pivots around an active edge until it contacts another point, and the contacted point is chosen to construct the new triangle. Petitjean presented their method based on regular interpolation [8], and the different properties are compared between the regular and irregular point sets. Huang [9] projected the K nearest points of each endpoint of an active edge onto the plane defined by the triangle adjacent to the active edge, and then a point is selected among the K points based on the minimal length criterion to form a new triangle. From a seed triangle, the algorithm of Lin grows a partially recovered triangular mesh by selecting a new point based on the intrinsic property of the point cloud, i.e. degree of sampling uniformity [10].

However, there are still four problems not well dealt with in region growing approach. (1) Determination of the seed triangle: the existing methods use specific principles to choose its three points, and a more comprehensive strategy needs to be defined to guarantee the quality of seed triangle and avoid possible awful effects on the subsequently generated triangles. (2) Determination of the influence region: closed region is often used to reduce the computing cost, but may cause poor performance on data with sharp and long edges. (3) Determination of the best active point: different with the seed triangle, the positional information in influence region should be considered in the evaluation criterion to select the most suitable active point. (4) Determination of the geometric integrity: geometric integrity has to be maintained in construction of new triangle, thus it is very important to analyze different situations of the selected active point in details and then treat with them respectively.

To solve the aforementioned problems, a novel region growing algorithm is proposed in this paper. Contributions of our method include: (1) points of seed triangle are determined with the principle considering both maximum angle and minimum length; (2) one open influence region instead of the closed region is defined for the active edge; (3) the criterion for active point selection also takes the positional element into account; (4) different situations of the active point and their corresponding treatments are analyzed in order to maintain the geometric integrity. The rest of our paper is organized as follows: related definitions and point cloud pre-processing are described in Section 2, the proposed new algorithm is presented in Section 3, experimental results are illustrated, compared and analyzed in Section 4, and then the conclusion is given in Section 5.

## 2 Definitions and Data Preprocess

### 2.1 Related Definitions

Following terms are defined for the proposed reconstruction algorithm:

(1) Active edge: each newly reconstructed edge prior to processing;
(2) Influence region: the region of searching for a new point to form a new triangle with an active edge, which may or may not contain a new point;
(3) Inner edge: the edge with 2 adjacent faces;
(4) Fixed point: a point from the point cloud whose incident edges are all inner edges;
(5) Bounded edge: an edge whose influence region contains only fixed points, i.e. the edge has only one face adjacent to it;

(6) Fixed edge: the edge whose adjacent triangles have been determined completely, thus both inner edge and bounded edge are fixed edges;
(7) Active point: the point with no edges or an active edge incident to it.

### 2.2 Preprocessing of Point Cloud

In region growing based surface recovery, searching for the best active point is very time consuming. For the unorganized $n$ scattered 3D points, the time complexity of global search on the entire point cloud is $O(n^2)$. The best active point can only lie in the neighbor region of the related active edge, i.e. the local regions of its two vertexes. Therefore, to avoid the global search, the preprocessing techniques are applied to determine the local region of each scattered point, i.e. its K nearest neighbors [13]. Based on our experiments, parameter K is assigned with a value which varies from 20 to 50.

## 3 Our Algorithm

### 3.1 Determination of the Seed Triangle

For region growing approach, quality of the reconstructed triangular mesh can be improved with a better seed triangle, and it is determined in our approach as follows:

Step 1. Search for the point $P$ whose z-value is the maximum in the point cloud;
Step 2. Search for the point $Q$ that is the nearest to $P$ and forms a line between them;
Step 3. Search for the third point $R$ ;

For point $R^*$ from $N_P^K$ (K nearest neighbors of $P$ ) or $N_Q^K$ (K nearest neighbors of $Q$ ), its energy value can be evaluated by the following functions:

$$E_s(R^*) = \mathrm{E}_L(R^*) \bullet \mathrm{E}_A(R^*) \tag{1}$$

$$\mathrm{E}_L(R^*) = d_{PQ}^2 + d_{PR^*}^2 + d_{QR^*}^2 \tag{2}$$

$$\mathrm{E}_A(R^*) = ctg\angle PR^*Q \tag{3}$$

where $d_{PQ}$ , $d_{PR^*}$ and $d_{QR^*}$ are the lengths of segment $PQ$ , $PR^*$ and $QR^*$ respectively, while $\angle PR^*Q$ is the angle between $PR^*$ and $QR^*$ . Therefore, considering the principle of both “minimum length” [14] and “maximum angle” [15], point $R$ is determined by

$$R = \min_{R^* \in (N_P^K \cup N_Q^K)} (E_s(R^*)) \tag{4}$$

Step 4. Take $\Delta PQR$ as the seed triangle.

Thereafter, normal vector of the seed triangle is adjusted to be outward as the base for the other triangles to be produced. If the inner product of the normal vector of the seed triangle and the constant vector (0, 0, 1) is positive, it points outward; otherwise the direction of the normal vector is reversed. Once outward normal vector of the seed triangle is determined, normal vectors of the subsequently generated triangles can be adjusted to be outward in the same way.

### 3.2 Influence Region for Active Edge

To maintain the topological consistency, a suitable active point for the active edge should be in front of the existing triangle, i.e. not all the active points satisfy with the necessary geometric constraints. As illustrated in Fig. 1, a new defined open influence region instead of the usually used close region is determined in our method to help filter the active points.

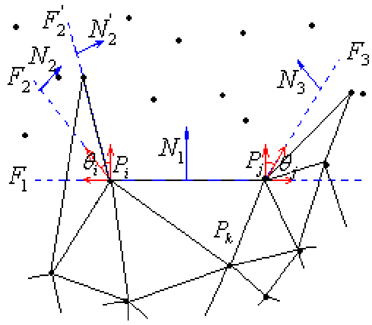


**Fig. 1.** The open influence region

The open influence region is determined with three faces, $F_1$, $F_2$ and $F_3$, and angle $\theta_i$, $\theta_j$ are used to control the influence region. Suppose $N_{ij}$ is the normal vector of triangle $\Delta P_i P_j P_k$, each face of the region can be represented by one point in the face and the normal vector of the face as

$$F_1(P_i, N_1 = norm(N_{ij} \times P_i P_j)) \tag{5}$$

$$F_2(P_i, N_2 = norm((N_1 + tg\theta_i \bullet norm(P_j P_i)) \times N_{ij}) \tag{6}$$

$$F_3(P_j, N_3 = norm(N_{ij} \times (N_1 + tg\theta_j \bullet norm(P_i P_j))) \tag{7}$$

To maintain the geometrical integrity of the reconstructed triangular mesh, the intersection between the triangle to be generated and the existing triangles should be null or the existing active edges. Thus, among the edges incident to point $P_i$ ( $P_j$ ), if there are some edges lying in the same side of face $F_2$ ( $F_3$ ) with the edge $P_i P_j$, face $F_2$ ( $F_3$ ) should be renewed. As shown in Fig. 1, face $F_2$ is changed in this case to be

$$F_2^{'}(P_i, N_2 = norm(P_i P_h \times N_{ij})) \tag{8}$$

### 3.3 Select the Most Suitable Active Point

There are some active points in the influence region of the active edge, one of them should be selected and used to construct a new triangle with the active edge in each step of the region growing method. Often used criteria to select the most suitable active point are minimum length or maximum angle. Besides them, the positional information is also considered in our algorithm to obtain better triangles.

Similar with selection of the third vertex of the seed triangle, the best active point is selected from K neighbors of two vertexes ( $N_{P_i}^K$ and $N_{P_j}^K$ ) of the active edge, and the difference is that the candidates are further filtered by the influence region. Suppose the set of active points in the influence region of active edge $P_i P_j$ is $Inf(P_i P_j)$, the most suitable active point only needs to be chosen from the point set of $(N_{P_i}^K \cup N_{P_j}^K) \cap Inf(P_i P_j)$.

It is possible to avoid more sharp triangles if the selected active point is not too close to any of the three faces of the open influence region, so positional element of the candidate point is also taken into account. Suppose $O$ is the geometric centre of $(N_{P_i}^K \cup N_{P_j}^K) \cap Inf(P_i P_j)$, the active point $P_{g^*}$ can be evaluated by the following functions:

$$\xi(P_{g^*}) = \mathrm{E}_L(P_{g^*}) \bullet \mathrm{E}_A(P_{g^*}) + \mathrm{E}_{Inf}(P_{g^*}) \tag{9}$$

$$\mathrm{E}_L(P_{g^*}) = \| P_i - P_j \|^2 + \| P_i - P_{g^*} \|^2 + \| P_j - P_{g^*} \|^2 \tag{10}$$

$$\mathrm{E}_A(P_{g^*}) = ctg\angle P_i P_{g^*} P_j \tag{11}$$

$$\mathrm{E}_{Inf}(P_{g^*}) = \eta \| P_{g^*} - O \| \tag{12}$$

where $\eta$ is the adjusting parameter. Its value usually varies with different 3D scattered point cloud. Thus the best active point $P_g$ is determined by

$$P_g = \min_{P_{g^*} \in (N_{P_i}^K \cup N_{P_j}^K) \cap \mathrm{I}nf(P_i P_j)} (\xi(P_{g^*})) \tag{13}$$

### 3.4 Constraints of Geometric Integrity

After the best active point is selected, it can be used to construct a new triangle with the active edge. However, generation of the new triangle may bring a sequence of effects, e.g. producing the other triangles, changing the status of some related edges or points. Therefore, this step is very important for the region growing approach. In our method, different situations of the active point are considered, and they are divided into ten situations as shown in Fig. 2 ( $P_g$ is the selected active point, $P_i$, $P_j$, $P_m$, $P_n$, $P_{m1}$, $P_{m2}$, $P_{m3}$, $P_{n1}$, $P_{n2}$, $P_{n3}$ are the points connected to $P_g$ ).

For the above ten situations, the corresponding treatments are different but their principle is similar. First, add the newly generated triangles (one, two or three) to the list of triangles, label the active edge just processed as fixed edge and then remove it from the active edge list. Second, check each newly generated edge: if the edge is active, add it into the active edge list and label it as active edge, otherwise label it as fixed edge. Third, check the states of the directly connected edges and points; change the state label if it has been converted from active to fixed while the edges should be removed from the active edge list if their states were converted from active to fixed.

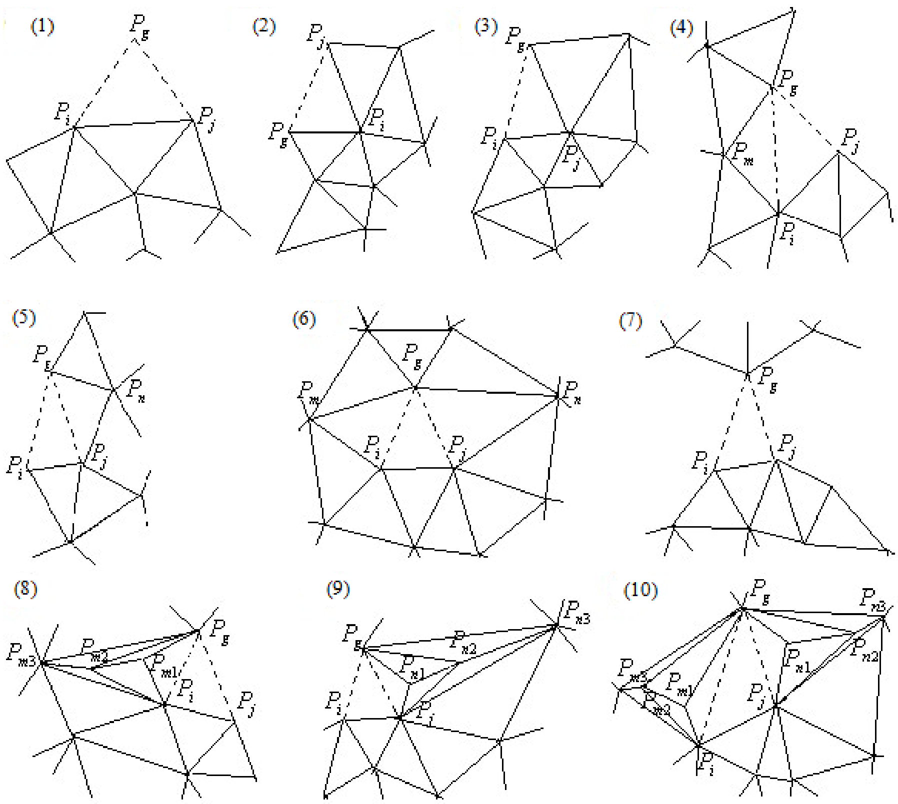


**Fig. 2.** Different situations of the active point

### 3.5 Description of the Algorithm

The pseudo-code for our recovery algorithm is described as follows:

```
Our recovery algorithm [in-file, out-file]
Input: in-file is the point cloud file.
Output: out-file is the triangle list file.
```

```
   (*APList: Active point list*/)
   (/*AEList: Active edge list*/)
   (/*TList: triangle list*/)
1. read in the point cloud from in-file and store them in
   APList.
2. Search K neighbors for each point.
3. determine the seed triangle and add 3 edges in AEList.
4. while (AElist is not empty) do
5.    determine influence region for one edge from AEList.
6.    if no active point in the influence region
         set the edge as boundary edge, update AEList, and then
         return 4.
7.    end if
8.    select the best point from active points in the influence
     region considering Eq. 13. add the newly generated
     triangles in TList.
9.    check and update APlist and AEList according to the
      contrain of geometric integrity.
10.  end while
11.  write out TList into out-file.
12. return.
```

## 4 Experimental Results

Our algorithm has been tested by experiments on a PC with Intel(R) Core(TM) i3 CPU, 2.27GHz, 1.92GB RAM. As shown in Fig. 3, seven sets of scanned 3D point cloud (1st row) including Bonze, Eight, Fandisk, Hand, Manequin, Rockeram and ThreePeaks are used as the experimental data, and the numbers of points in them are 32570, 776, 11984, 39231, 11703, 10043 and 1907 respectively. The recovered triangular meshes of the 7 models (2nd row) show that the proposed algorithm can generate surfaces with good quality from scattered 3D points.

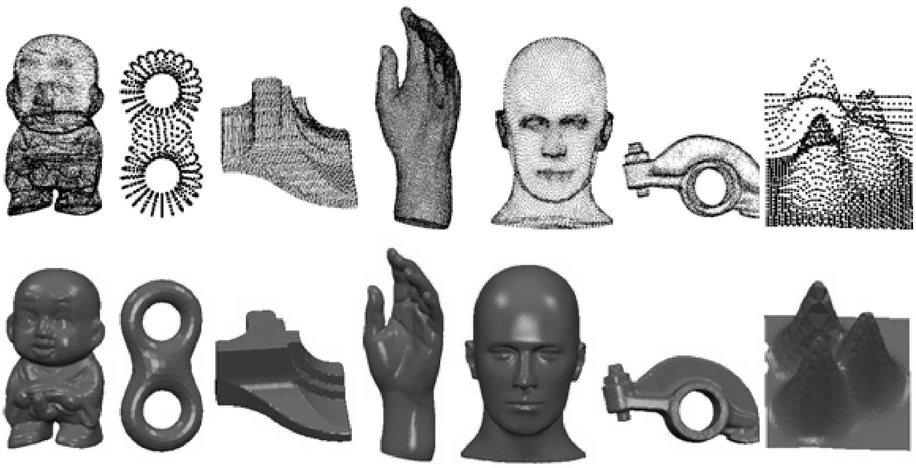

**Fig. 3.** Triangular mesh recovery from 7 point clouds

To illustrate the performance of our method for more complicated point clouds, three data, i.e. Buster with sharp edges, Dino with long tails, Frog with bottom holes are tested, as shown in Fig. 4. It can be found that fine details are preserved in the recovered surface.

Ref. [10] is also a region growing based approach for triangular mesh recovery, so it is implemented and compared with our algorithm in both speed and accuracy. Computational cost of 7 point clouds in Fig. 3 are listed in Table 1, where K is the number of nearest neighbors, KNN is the time spent on searching for K neighbors, Ref1 and Ref2 are spent time from Ref. [10] for surface recovery without and with KNN, Our1 and Our2 are spent time from our algorithm without and with KNN. It can be found that our approach has better performance in time complexity.

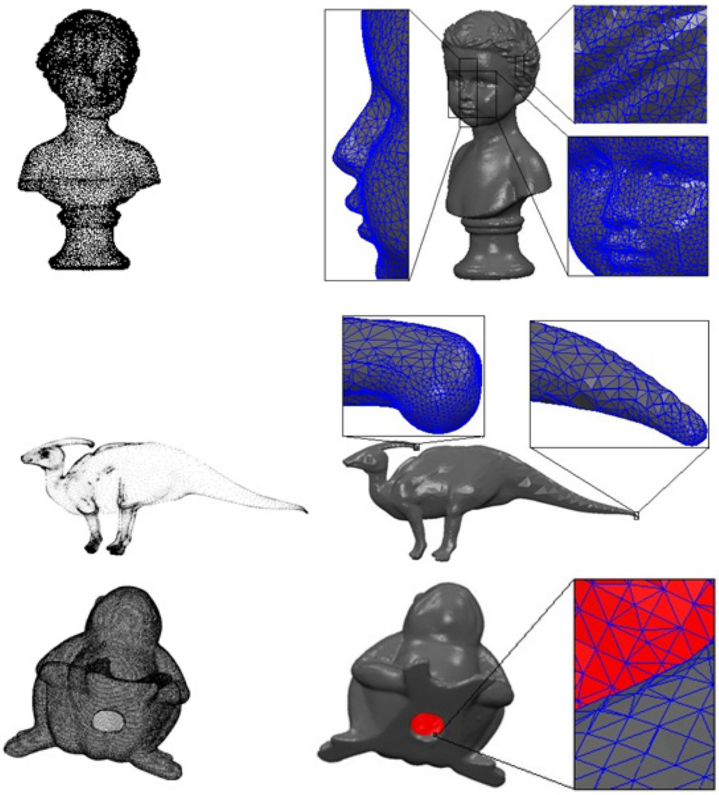

**Fig. 4.** Reconstructed results with enlarged details

Two sets of 3D scattered point clouds are selected to compare the quality of reconstructed surface, as shown in Fig. 5. The original point clouds are presented in the first row; the reconstructed surfaces from Ref. [10] are displayed in the second row, while the reconstructed triangular meshes from our algorithm are displayed in the third row. For point cloud on the left, result from Ref. [10] ignores many details on the border, but result from our method recovers the boundary shape much better. For point cloud on the right, result from Ref. [10] can only generate a rough approximation of the hole in the 3D point cloud, but result from our method has reconstructed the hole with much higher precision. It can be proved that our proposed approach has the ability to produce the triangular meshes with good quality from scattered 3D points while preserving the topological coherence from the original point cloud.

**Table 1.** Comparison of computational expense (Unit: second)

| Name | Vertices | K | KNN | Ref1 | Our1 | Ref2 | Our2 |
|---|---|---|---|---|---|---|---|
| Bonze | 32570 | 25 | 1.25 | 2.921 | 1.468 | 4.171 | 2.718 |
| Eight | 776 | 28 | 0.093 | 0.094 | 0.047 | 0.187 | 0.14 |
| Fandisk | 11984 | 30 | 0.5 | 1.297 | 0.64 | 1.797 | 1.14 |
| Hand | 39231 | 30 | 1.562 | 3.703 | 2.062 | 5.265 | 4.624 |
| Manequin | 11703 | 35 | 0.578 | 1.453 | 0.672 | 2.031 | 1.25 |
| Rockeram | 10043 | 35 | 0.546 | 1.203 | 0.563 | 1.749 | 1.109 |
| ThreePeaks | 1907 | 25 | 0.125 | 0.156 | 0.078 | 0.281 | 0.203 |

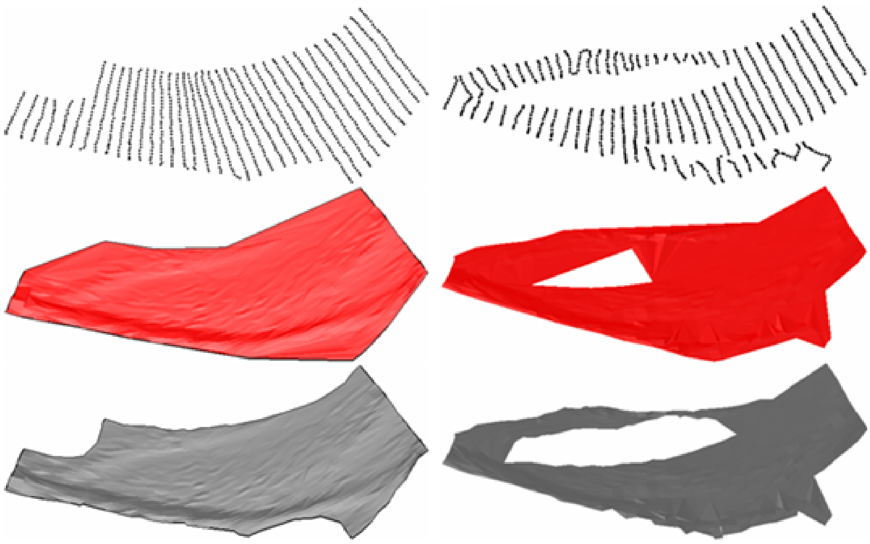

**Fig. 5.** Comparison of surface quality

## 5 Conclusion

A novel region growing based algorithm is presented in this paper for triangular surface reconstruction from unorganized 3D points. For the four key problems, i.e. determinations of seed triangle, influence region, the best active point, and the geometric integrity, related new techniques are proposed to deal with them.

Our method can recover a triangular mesh from the scattered 3D point cloud with good quality. Compared with the existing similar techniques, our algorithm can perform better in both accuracy and speed. Currently, the method is tested with more point clouds, and will be compared with the other approaches for triangular mesh reconstruction except for the ones of region growing type.

**Acknowledgments.** The work was supported by National Basic Research Program of China (973 Program, No. 2011CB707904), National Natural Science Foundation of China (No. 60603079, No. 61070079, No. 71001066), Research Grants Council of Hong Kong (No. 613008), Shanghai Pujiang Program (No. PJ[2009]00861), PhD Programs Foundation from The Ministry of Education of China (No. 20100073120046).

## References

1. Kuo, C.-C., Yau, H.-T.: Reconstruction of virtual parts from unorganized scanned data for automated dimensional inspection. J. Comput Inf. Sci. Eng. Trans. ASME 3(1), 76–86 (2003)
2. Kolingerova, I., Zalik, B.: Reconstructing domain boundaries within a given set of points using Delaunay triangulation. Computers & Geosciences 32(9), 1310–1319 (2006)
3. Wang, D., Hassan, O., Morgan, K., Weatherill, N.: Efficient surface reconstruction from contours based on two-dimensional delaunay triangulation. International Journal for Numerical Methods in Engineering 65(5), 734–751 (2006)
4. Hoppe, H., DeRose, T., Duchampy, T., McDonaldz, J., Stuetzlez, W.: Surface reconstruction from unorganized points. In: Proceedings of SIGGRAPH 1992, pp. 71–78 (1992)
5. Carr, J.-C., Beatson, R.-K., Cherrie, J.-B., Mitchell, T.-J., Fright, W.-R., McCallum, B.-C.: Reconstruction and representation of 3D objects with radial basis functions. In: Proceedings of SIGGRAPH 2001, pp. 67–76 (2001)
6. Schreiner, J., Scheiclegger, C.-E., Silva, C.-T.: High-quality extraction of Isosurfaces from regular and irregular grids. Visualization and Computer Graphics 12(5), 1205–1212 (2006)
7. Bernardini, F., Mittleman, J., Rushmeier, H., Silva, C., Taubin, G.: The ball pivoting algorithm for surface reconstruction. IEEE Trans. Vis. Comput. Graph. 5(4), 349–359 (1999)
8. Petitjean, S., Boyer, E.: Regular and non-regular point sets: properties and reconstruction. Comput. Geom. Theory 19, 101–126 (2001)
9. Huang, J., Menq, C.-H.: Combinatorial manifold mesh reconstruction and optimization from unorganized points with arbitrary topology. Comput-Aided Design 34(2), 149–165 (2002)
10. Lin, H., Tai, C., Wang, G.: A mesh reconstruction algorithm driven by an intrinsic property of a point cloud. Computer Aided Design 36(1), 1–9 (2004)
11. DoRego, R.-L., Araujo, A.-F., De, L.-N., Fernando, B.: Growing self-organizing maps for surface reconstruction from unstructured point clouds. In: IEEE International Conference on Neural Networks, pp. 1900–1905 (2007)
12. Lv, H., Wang, Y.: A heuristic approach to reconstruct triangle mesh from unorganized point cloud. In: The 6th International Conference on Fuzzy Systems and Knowledge Discovery, pp. 87–91 (2009)
13. Jagan, S., Hanan, S., Amitabh, V.: A fast all nearest neighbor algorithm for applications involving large point-clouds. Computers & Graphics 31, 157–174 (2007)
14. Boissonant, J.-D., Cazals, F.: Smooth surface reconstruction via natural neighbor interpolation of distance functions. In: Proceedings of 16th Annual Symposium On Computational Geometry (SCG 2000), Hong Kong, pp. 223–232 (2000)
15. Cheng, W., Sorguc, A.-G., Shinoda, J., Hagiwara, I.: MOAA and topology judgment for mesh construction. American Society of Mechanical Engineers 482, 227–238 (2004)