A Fast Algorithm for Delaunay based Surface Reconstruction

Shan Gao, Han-Qing Lu
National Laboratory of Pattern Recognition
Chinese Academy of Science, Beijing 100080, P.R. China
shgao@nlpr.ia.ac.cn

ABSTRACT

Advanced 3D scanning technologies enable us to obtain dense and accurate surface sample point sets. From sufficiently dense sample point set, Crust algorithm, which is based on Voronoi diagram and its dual Delaunay triangulation, can reconstruct a triangle mesh that is topologically valid and convergent to the original surface. However, the algorithm is restricted in the practical application because of its long running time. Based on the fact that we do not need dense sample in featureless area for successful reconstruction, we propose a non-uniformly sampling method to resample the input data set according to the local feature size before reconstruction. In this way, we increase the speed of reconstruction without losing the details we need.

Keywords

surface reconstruction, Delaunay triangulation, geometric modeling.

1. INTRODUCTION

With the development of 3D scanning technologies, we are now able to obtain dense, accurate samples of real objects' surface, and modeling complex objects from samples becomes a significant recent trend in geometric modeling [Rusink00]. As the sample points only have the information of their 3D position, surface reconstruction, which is to build a piecewise linear surface approximating the original surface, is one essential problem of this modeling method.

2. OVERVIEW

In recent years, people have proposed a lot of algorithms for the problem. These algorithms can be roughly divided into two completely different kinds: approximation and interpolation. The first kind of approach generally estimates an approximating surface that passes close by the original sample points and its typical work is the algorithm presented by Hoppe et al [Hoppe96]. The second kind of approach normally uses Voronoi diagram and Delaunay triangulation to find the topological connection of the sample points. Different to the result of the first kind, the surface reconstructed by the second kind of approach passes through the original sample points. The α-shape of Edelsbrunner et al [Edels94], the crust of Amenta and Bern [Ament99][Ament98a][Ament98b] are both included in the second kind.

Compare to other algorithms, the Crust algorithm is not only simple and direct in theory but also faithful to the original surface. In view of that the first kind of algorithms approximate rather than interpolate the original surface, they potentially do some low-pass filtering of the data. As we are considering the general surface reconstruction problem here, we actually use the same filter to get rid of types of noise. Apparently, it can't have good result to every input data. So, the best way is to filter noise before reconstruction. That is to say we don't consider noises in the input data. In this case, the result of the former kind algorithm is certainly more faithful than that of the latter kind.

Unlike α-shape algorithm, Crust doesn't need to choose any parameter, which is the major drawback of α-shape method. When the sample is sufficiently dense, it can automatically reconstruct a triangle mesh that is topologically valid and convergent to the original surface.

However, Crust algorithm is too slow for many practical applications with current computing resource. Unless we can improve its speed, it can't be used it in large data set. In this paper, we present a non-uniformly sampling method to decrease the complexity of reconstruction. The down-sampled point set is dense in detailed areas and sparse in featureless areas. The reconstructed surface has the same topology of the original surface, and the details are maintained well.

3. DEFINITIONS

Our approach is built on the Crust algorithm introduced in [Ament99][Ament98a][Ament98b].

This algorithm is based on the following definitions.

Definition 1. The medial axis of a surface F is the closure of all centers of the spheres touching the surface in more than one point.

Definition 2. To any point p on F, its local feature size $LFS(p)$ is the Euclidean distance from p to the nearest point on medial axis.

Definition 3. Let S be a sample set of F, if the Euclidean distance from any point p on F to the nearest sample point is within $r \cdot LFS(p)$, then S is an r-sample of F.

Definition 4. The positive pole of a sample s is the farthest vertex in Voronoi cell V_s, and its negative pole is the farthest vertex of V_s on the other side of the surface.

Definition 5. Let S be a sufficiently dense sample point set from a surface F, the Crust of S is composed by the triangles one of whose circumsphere is empty both of the samples and the medial axis.

It has been observed that an r-sample with $r = 0.5$ is generally dense enough for Crust to correctly reconstruct the surface [Ament99].

4. ALGORITHM

We assume that the input point set S is a sufficiently dense sample of a smooth surface.

In Crust algorithm, we first compute the Voronoi diagram of the sample and select the poles in the Voronoi vertices to estimate the medial axis, then we compute the Delaunay triangulation of the combined point set of the samples and poles, in the end we choose the triangles whose vertices are all samples. From the process of the algorithm, we can see that the most time-wasting step of Crust algorithm is the computation of 3D Voronoi Diagram and Delaunay triangulation. Notice that the number of sample and poles is at most 3n, the time complexity of the algorithm is about $O(n^2) + O(9n^2)$, where n is the number of input points. Therefore, there are two ways to reduce the complexity: improve the efficiency of the computation of 3D Voronoi Diagram, or decrease the number of points. Voronoi diagram and its dual Delaunay triangulation have been studied widely since it was presented in 1936. It is difficult to improve efficiency of algorithm in advance. Thus we try the second way.

Notice that the local feature size is big in featureless area and small in detailed area, Crust does not require dense sample everywhere. However, as the surface is unknown, sample device can't know the local feature size of the area it is sampling, it is almost impossible to realize r-sample. If we do it manually, on the one hand the sampling process will be quite troublesome, on the other hand people can only evaluate how detail the surface is so that the sample can't be very well coincident to the r-sample's requirement. In order to maintain the detail information in the reconstructed model, people usually desire the sample as dense as possible. The result is that the input point set is often with a great deal of points that are not necessary to correct reconstruction. If we discard these points, we can still correctly reconstruct the surface without losing details. In addition, the running time of reconstruction will be reduced.

5.1. Local feature size

As the sample is assumed dense enough, the poles are approximate to the medial axis. According to the definition of pole, the nearest pole of a sample s is its negative pole. Thus we can use the distance between them to approximate the sample's local feature size.

5.2. Non-uniformly down sampling

If S is an r-sample of F and p is a point on F, then the distance between p and its nearest sample point s is within $r \cdot LFS(p)$. Since every sample is also a point on F, the distance between s and s_1 is no more than $r \cdot LFS(s)$, where s_1 is the nearest point of s in S.

As show in figure 1, s is a point in S, v is the negative pole of s, s_1 is another point in S that $d(s, s_1) = r_s \cdot LFS(s)$. Let s be the center and $r_s \cdot LFS(s)$ be the radius, we have the ball B_1. Let v be the center, $LFS(s)$ be the radius, we have another ball B_2. In accordance with the definition of local feature size, s_1 is outside ball B_2. Passing through s we make a plane L tangent to F. Because of the assumption that the surface is smooth, s_1 and B_2 must be located the same side of L. From the above discussion, we can see that F must be in the shaded region of figure 1 if it is in B_1.

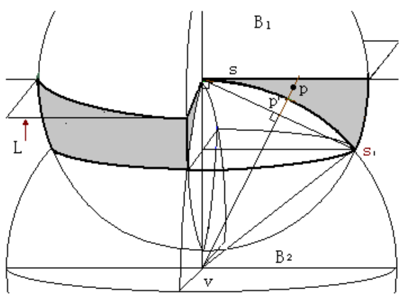


Figure1. If the surface is in the ball B_1, it must be in the shaded region.

There are two factors influencing local feature size – the curvature and proximity of the other parts of the surface[Ament98b]. However, the second factor can't affect the local feature size in a small region, so we need not take into account the factor in a local area. That is to say, the local feature size is inversely proportional to the curvature in the shaded region when r is small enough.

Let p be a point on the surface in the shaded region, and p' is the intersection of the line pv and B_2. As we all known, the more flat the surface is, the lower the curvature is. It is apparent that the curvature of point p is smaller than that of point p'. Since point p' and s are both on the ball B_2, their curvatures are the same. Thus, we have $LFS(p) \geq LFS(s)$. In addition, on account of that p is in the shaded region, we have $d(s,p) \leq d(s,s_1)$. As a result, we get $d(s,p) \leq r_s \cdot LFS(p)$. As S satisfies the requirement for r-sample, r_s is less than r. So, we have $d(s,p) \leq r \cdot LFS(p)$.

Then, we can make the following conclusion: if we can find another point $s' \in S$ that satisfied equation $d(s,s') \leq r \cdot LFS(s)$, S is an r-sample of a surface F.

Therefore, if we delete all the points in the shaded area excepting the farthest one and s itself, the down-sampled point set S' is still an r-sample of F In [Ament99] it is written that, an r-sample point set is sufficiently dense for correctly reconstruction if r is no more than 0.5. Thus, r should be less than 0.5 here. In fact, we obtain good result when $r = 0.5$.

Down-sampling:

```
Initial every point in S as unmarked
For (i=0; i<n; i++){
  if s_i is unmarked {
    d_max = 0; m = 0;
    for (j=0; j<n; j++) {
      if s_j is unmarked {
        if d(s_i, s_j) < r · LFS(s_i) {
          marked s_j ;
          if d(s_i, s_j) > d_max update d_max and m
}}
unmarked s_m;}
select all the unmarked points as the down-
sampled point set
```

5. SURFACE RECONSTRUCTION

In view of the fact that the poles of denser sample approximate the medial axis better than that of sparser one, we use the poles evaluated in the previous steps. Just like Crust algorithm, we combine the down-sampled point set and its corresponding poles to a new point set. The following steps are the same to Crust: we compute Delaunay triangulation for the new point set and select the triangles in which the three vertices are all sample points as the simplices of the reconstructed surface.

5.1. Experimental Result

We experiment with the two data sets ---- Mannequin and Stanford Bunny. Here, Voronoi diagram and Delaunay triangulation are implemented by the free qhull code [Qhull99] from Geometry center, and the parameter r is chosen as 0.5. The result is show in figure 2 and figure 3.

5.2. The reduction of data

Just as our expectation, the density of down sampled point set is varied according to the surface's detail. The samples are still very dense in the region like the eyes, mouth and ears of mannequin. But in the featureless region, such as the jaw and forehead, they are very sparse compare with the original dataset. In the example of Stanford bunny the points are reduced relatively uniformly. It is because that the surface of bunny does not change very quickly.

Form the result we also can see that the reduction of data is varied with the different dataset. It is relied on the density of the input points: the denser the input data set is, the more points we can delete. In the example of Mannequin, the size of new data set is reduced to about 1/3, however it is about 1/4 in Bunny.

5.3. Complexity

Now let us compare the complexity of the algorithm. The running time is dominated by the following steps: computing the Voronoi diagram of the input point set, down sampling, computing the Delaunay triangulation of the down-sampled point set. The core operation of down sampling is the computation of two points' distance, and the amount of the operation is within $(n-1) * n_{new}$, so the asymptotic complexity of down sampling is $O(n^2)$. Therefore, the total complexity of our approach is about $O(n^2) + O(n^2) + O(9n_{new}^2)$. Comparing with using the input dataset directly, the complexity is decreased about $O(9n^2) - O(n^2) - O(9n_{new}^2)$, here $n_{new} \approx \frac{1}{4}n$ in the Bunny, $n_{new} \approx \frac{1}{3}n$ in Mannequin. In these two examples, the computation is both decreased more than 50%.

5.4. Smooth rendering

Since the points are sparse in featureless region, the triangles approximating the surface are comparatively large there. That makes the reconstructed surface look very coarse. As we know that the result of Gouraud shading look much smoother than that of flat shading. We try Gouraud shading to solve the problem. It is a very simple and effective method. We first calculate average normal of all the triangles sharing one vertex, following that we use the value as the normal of that vertex. Next we bi-linearly interpolate the normal of the vertices as the normal of the surface inside the triangle [Gouraud71]. From figure 2 (e) and figure 3 (e), we can see that the result of that method is satisfying --- the reconstructed surfaces are acceptable now.

6. CONCLUSION

We have presented a non-uniform down sampling method for dense and unorganized point set before surface reconstruction according to the local feature size. Guaranteeing the topological shape, we use a smaller point set to reconstruct the original surface. As the result, the speed of reconstruction is improved. This method also can be applied in mesh simplification. In fact we can use r-sample to define the level of detail for mesh. With the increasing of r, the mesh's level of detail is decreasing. So we can realize mesh simplification by using this down-sampling method to build an r-sample model with bigger r.

As the triangles in the plat areas are relatively large, the whole model looks coarse. However, Gouraud shading can give us a tolerable visual effect when r is not very big. In addition, if we want a more elaborate visual effect, subdivision can be used to get smooth surface.

7. REFERENCES

[Ament98a] N. Amenta, M. Bern and D. Eppstein. The Crust and the β-skeleton: Combinatorial curve reconstruction. Graphical Models and Image Processing, 60(1998), pp. 125-135.

[Ament98b] N. Amenta, M. Bern and M. Kamvysselis. A new Voronoi-based surface reconstruction algorithm. In SIGGRAPH'98 Conference Proceedings (August 1998), pp. 415-421.

[Ament99] N. Amenta and M. Bern. Surface Reconstruction by Voronoi filtering. Discrete and Computational. Geometry, 22(1999), pp. 481-504.

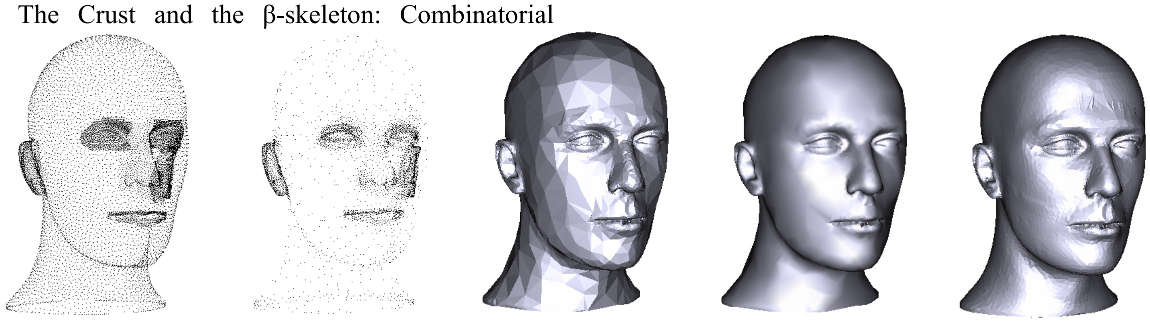
[Edels94] H. Edelsbrunner and E. P. Mucke, Three-Dimensional alpha shapes, ACM Transactions on Graphics 13, 43-72, Jan. 1994.

[Gouraud71] H. Gouraud, Computer Display of Curved surfaces, IEEE Trans. Computers, vol. 20, No. 6, pp. 311-317,1971.

[Hoppe96] H. Hoppe, T. DeRose, T. Duchamp, J. MeDonald and W. Stuetzle. Surface reconstruction from unorganized points. In SIGGRAPH'92 Conference Proceedings (August 1996), pp. 71-78.

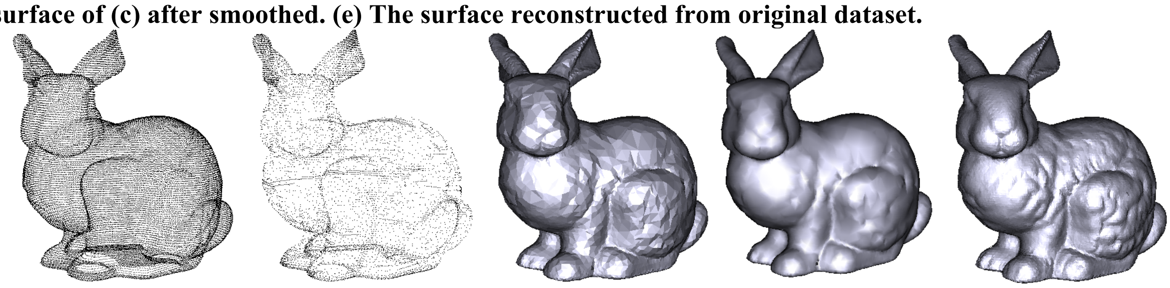
[Qhull99] C. B. Barber and H. T. Huhdanpaa. Manual for Qhull and Rbox. http://www.geom.umn.edu/locate/qhull/manual.

[Rusink00] Syzmon Rusinkiewicz, Marc Levoy. QSplat: A Multiresolution Point Rendering System for Large Meshes. In SIGGRAPH'00 Conference Proceedings (Auguest 2000), pp. 343-252.



(a) (b) (c) (d) (e)

Figure2: Dataset Mannequin (a) The Point cloud of original dataset. (12772 points) (b) The point cloud of down-sampled dataset. (4820 points) (c) The surface reconstructed from down-sampled dataset. (d) The surface of (c) after smoothed. (e) The surface reconstructed from original dataset.



(a) (b) (c) (d) (e)

Figure 3: Dataset Stanford bunny (a) The Point cloud of original dataset. (35947 points) (b) The point cloud of down-sampled dataset. (8845 points) (c) The surface reconstructed from down-sampled dataset. (d) The surface of (c) after smoothed. (e) The surface reconstructed from original dataset.