Three Dimensional Surface Reconstruction for Shape Reforming

Kyi Soe, Myint Myint Sein
University of Computer Studies, Yangon
kyisoemisc@gmail.com, chuchu0218@gmail.com

Abstract

Surface reconstruction is needed for almost all modeling and visualization applications, but unfortunately 3D data from a passive vision system are often insufficient for a traditional surface reconstruction technique from 2D images. In this paper, we develop a method for surface reconstruction from 3D data set. This gives a more robust approach than existing methods using only pure 2D information. Our method associates 3D Delaunay triangulation data structure with the geometric convection algorithm. Moreover, we also use the 3D Voronoi Diagram for reconstructing the surface. Given a good program for this fundamental subroutine, the algorithm is quite easy to simple. Our method may also be used to simplify shapes by removing small features which would require an excessive number of elements to preserve them in the output mesh. In our experiment, a tsubo(pot) 3D data set is used.

1. Introduction

Surface reconstruction is a natural extension of the point-based geometric methods. The recent advances in 3D scanning technologies have led to an increasing need for techniques capable of processing massive discrete geometric data. In the last year, a great deal of work has been carried out on surface reconstruction from datasets with millions of sample points, including unorganized point sets [3, 6, 10] and set of range images [9]. These methods are often used to produce a triangulated mesh surface, which is standard representation for fast visualization and geometry processing algorithms. However, the data used to generate these meshes are generally overly dense, due to uniform grid sampling patterns, and a mesh simplification step is required for use in common applications.

Point set simplification techniques offer an alternative to the standard pipeline by introducing a simplification step before the reconstruction process. The former aim at reducing the redundancy of the input data in order to accelerate subsequent reconstruction or visualization.

Several algorithms that perform reconstruction and simplification in a single framework have been recently studied. Boissonnat and Cazals [2] have proposed a Delaunay-based coarase-to-fine reconstruction algorithm controlled by a signed distance function to implicit surface. Ohtake et al. [10] have developed an algorithm that resamples a point set using a quadric error metric, coupled with a specific fast local triangulation procedure. In both cases, the resulting sampling remains static, and the reconstructed surface cannot be easily updated, especially if the level of detail needs to be modified afterwards, or if additional data become available later (e.g. when streaming data on a network, or during a digital acquisition project). Starting from a dense unorganized input point set, the authors reconstruct a simplified triangulated surface by means of a Delaunay-based surface reconstruction algorithm called geometric convection [5] coupled with a local point set subsampling procedure. The Delaunay triangulation is constructed only for the retained sample points in order to maintain some history of the reconstruction process. The reconstructed surface can then be easily updated by inserting or removing sample points without restarting the reconstruction process from scratch. However, the method lacks an efficient data-structure to handle large data.

The algorithm is based on the three-dimensional Voronoi diagram and Delaunay triangulation; it produces a set of triangles. All vertices of triangles are sample points; in fact, all triangles appear in the Delaunay triangulation of the sample points. We demonstrate the effectiveness of our method. Our algorithm inherits the robustness properties of the original geometric convection algorithm [5].

2. Geometric convection

In this section, we briefly review the geometric convection algorithm as described by Chaine in [5]. This algorithm serves as the basic for our surface reconstruction method.

The geometric convection algorithm is a surface reconstruction algorithm that proceeds by filtering the Delaunay triangulation of an input point set sampled from a smooth surface [4]. This method has some similarities with the Wrap [7] and Flow Complex [8] techniques. The filtration is guided by a convection scheme related to level set methods [11] that consists in shrinking an enclosing surface under the influence of the gradient field of a distance function to the closest sample point. This process results in a closed, oriented triangulated surface embedded in the Delaunay triangulation of the point set, and characterized by an *oriented Gabriel property* [5]. This means that for every facet, the diametral half-ball located inside the surface, or *Gabriel half-ball*, contains no sample point.

Let $P \subset \Re^3$ denote the input point set and $\hat{S}$ the surface in convection. The convection scheme can be completely achieved through the Delaunay triangulation of P by removing the facets that do not meet the oriented Gabriel property through an iterative sculpting process that starts from the convex hull. The $\hat{S}$ surface is a closed triangulated surface is maintained at every step, all the facets oriented inwards. Two facets with opposite orientations can meet – they are said to be coupled. Coupled facets can collapse locally, which may change the topology of $\hat{S}$. A local study (or a more global solution) is required to dig into packets that may locally block the convection scheme, e.g., based on local granularity. The order in which the facets of the evolving surface are processed does not influence the result. This is a reason why this algorithm is a good candidate to be translated into this version.

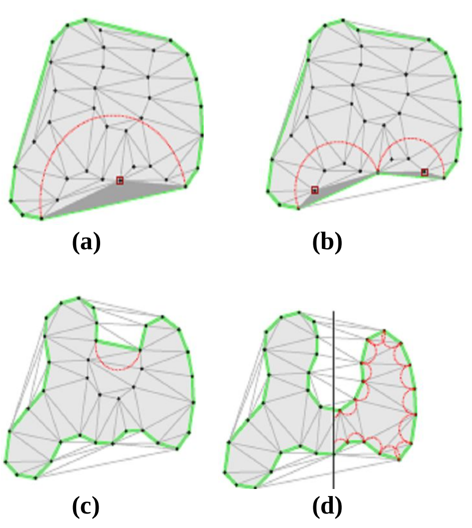

Figure 1: Geometric convection towards a 2D point set. In (a), an enclosing curve is initialized on the convex hull of the point set. The current edge, enclosed by a non-empty Gabriel half-ball, forms a Delaunay triangle (dark gray) with the square point. This triangle becomes external, the curve is updated (b), and it continues to shrink. In (c), an edge is found to block a pocket; it will be forced. The final result is shown in (d) with some empty Gabriel half-balls.

3. Geometry

We start by reviewing some standard geometric constructions. Given a discrete set S of sample points in $\Re^3$, the *Voronoi cell*of a sample point is that part of $\Re^3$ closer to it than to any other sample. The *Voronoi diagram* is the decomposition of $\Re^3$ induced by the Voronoi cells. Each Voronoi cell is a convex polytope, and its vertices are the *Voronoi vertices*; when S is nondegenerate, each Voronoi vertex is equidistant from exactly $d + 1$ points of S. These $d + 1$ points are the vertices of the *Delaunay simplex*, dual to the Voronoi vertex. A Delaunay simplex, and hence each of its faces, has a circumsphere empty of other points of S. The set of Delaunay simplices form the *Delaunay triangulation* of S. Computing the Delaunay triangulation essentially computes the Voronoi diagram as well.

3.1 Voronoi Diagarm

This simple Voronoi filtering algorithm runs into a snag in three dimensions. The nice property that all the Voronoi vertices of a sufficiently dense sample lie near the medial axis is no longer true. No matter how densely we sample, Voronoi vertices can appear arbitrarily close to the surface.

On the other hand, many of the three-dimensional Voronoi vertices *do* lie near the medial axis. Consider the Voronoi cell V_s of a sample S. The sample S is surrounded on F by other samples, and V_s is bounded by bisecting planes separating from its neighbors, each plane nearly perpendicular to F at S. So the Voronoi cell V_s is long, thin and roughly perpendicular F at S. V_s extends perpendicularly out to the medial axis. Near the medial axis, other samples on F become closer than S, and V_s is cut off. This guarantees that some vertices of V_s lie near the medial axis. We give a precise and quantitative version of this rough argument in [12]. This leads to the following algorithm. Instead of using all of the Voronoi vertices in the Voronoi filtering step, for each sample S we use only the two vertices of V_s farthest from S, one on either side of the surface F. We call these the poles of S, and denote them p^+ and p^-. It is easy to find one pole, say p^+: the farthest vertex of V_s from S. The observation that V_s is long and thin implies that the other pole p^- must lie roughly in the opposite direction. Thus in the basic algorithm below, we simply choose p^- to be farthest vertex from S such that sp^- and sp^+ have negative dot-product. Here is the basic algorithm:

1. Compute the Voronoi diagram of the sample points S.
2. For each sample point S do:
   (a) If S does not lie on the convex hull, let p^+ be the farthest Voronoi vertex of V_s from S. Let n^+ be the vector sp^+.
   (b) If S lies on the convex hull, let n^+ be the average of the outer normal of the adjacent triangles.
   (c) Let p^- be the Voronoi vertex of V_s with negative projection on n^+ that is farthest from S.
3. Let P be the set of all poles p^+ and p^-. Compute the Delaunay triangulation of $S \cup P$.
4. Keep only those triangles for which all three vertices are sample points in S.

3.2 Delaunay Triangulation

In the following paragraphs, we describe the Delaunay triangulation data-structure and the reconstruction algorithm in detail.

A tetrahedron is said to be in *conflict* with a point **p** if **p** is contained in its circumsphere. At a given time, a tetrahedron is said to be *final* if it cannot be in conflict with any further inserted point. A subset of tetrahedral of the triangulation is said to be *finalized* if all are final. Suppose that we have successively loaded and triangulated slices S_0, S_1, ….., S_k, with k < n. Reconstructing a coherent surface using geometric convection requires that every tetrahedron to be traversed by the shrinking surface is certified as final. Let t denote a tetrahedron with a vertex **p** in $\cup_{i=0}^{k} S_i$. If the circumsphere of t does not overlap S_{k+1}, then t is final. Otherwise, there is no known upper bound on the number of slices that still have to be loaded before t is certified. To make the reconstruction process possible while controlling the number of loaded slices, our strategy consists in computing and inserting extra points in the triangulation through Delaunay refinement process, so that the set of tetrahedral intersecting a target slice can be finalized, and the extra points are far from the sampled surface in this slice. Before loading a new slice, extra points falling above the loaded slices are removed to prevent them from interfering with the reconstruction result. Finalized tetrahedral are preserved under these extra point removal steps. The algorithm refines the tetrahedral that do not satisfy the circumsphere-slice overlapping condition, called bad tetrahedral, by adding their circumferences as vertices. Furthermore, the algorithm guarantees that no extra point is inserted outside the bounding box *B*. The detailed refinement rule can be stated as follows:

Rule If there is a bad tetrahedron *t*:
  Compute the circumcenter $\mathbf{c}_t$ of *t*;
  if $\mathbf{c}_t$ is outside *B* then
    let *f* denote a facet of *t* visible by $\mathbf{c}_t$;
    Compute the circumcenter $\mathbf{c}_f$ of *f*;
    if $\mathbf{c}_f$ is outside *B* then
      let *e* denote the edge of *f* visible by $\mathbf{c}_f$;
      compute the midpoint $\mathbf{c}_e$ of e;
      insert $\mathbf{c}_e$;
    else
      insert $\mathbf{c}_f$;
    end if
  else
    insert $\mathbf{c}_t$;
  end if

4. Experimental Results

A tsubo (pot) 3D data set is used in our experiment. It is a less texture object and difficult to obtain the exact feature points from image. In the following Figure 2 shows original tsubo (pot) image. In the next Figure 3 display the plotting of tsubo (pot) 3D data set and the final Figure 5 depicts the 3D of tsubo (pot).

Figure 2 : Original tsubo (pot) image

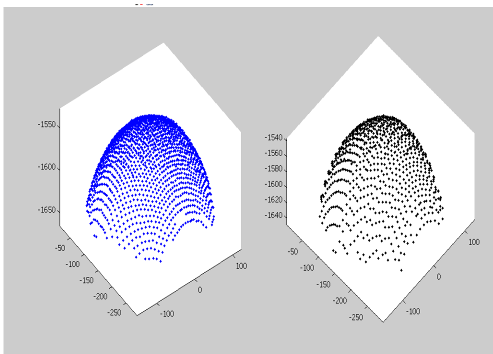

Figure 4 : Reducing the 3D Data Set

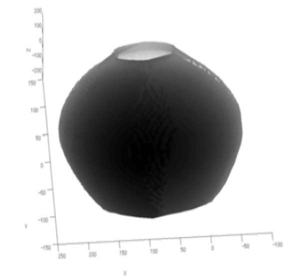

Figure 5 : Complete reconstructed of a tsubo

5. Conclusions

This paper introduces a method for 3D surface reconstruction from 3D data set. We have designed a Delaunay triangulation data-structure that fits the purpose of reconstructing surfaces from a stream of points without the need to compute the triangulation of the whole data set and to maintain it in memory. Our framework may also differ in interesting perspective for other Delaunay-based surface reconstruction algorithms. Our current method interpolates the input points, which can be an advantage of for some applications. But performance could be significantly improved in case the data need simplification. And our method is not incremental, and our implantation is too slow for real-time decompression, so this application motivates work in both directions.

6. References

[1] Allegre R., Chaine R., and Akkouche, "Convection-Driven Dynamic Surface Reconstruction", *In Proc. Shape Modeling International*, IEEE Computer Society Press, 2005, pp. 33-42.

[2] Boissonnat J.-D., and Cazals F., "Corase-to-fine surface simplification with geometric guarantees. *In Proc. Eugraphics*, 2001, pp. 490-499.

[3] Bernardini F., MIttleman J., Rushmeier H., Silva C., and Taubin G., "The Ball Pivoting Algorithm for Surface Reconstruction", *IEEE Transactions on Visualization and Computer Graphics 5, 4,* 1999, pp. 349-359.

[4] Cazals F., and Giesen J., "Delaunay Triangulation based Surface Reconstruction: Ideas and Algorithms", Tech. Rep 5393, INRA, November 2004.

[5] Chaine R., "A Geometric convection approach of 3D reconstruction", *In Proc. Euographics Symposium on Geometry Processing*, 2003, pp. 218-229.

[6] Dey T. K., Giesen J., and Hudson J., "Decimating samples for mesh simplification", *In Proc. Canadian Conference on Computational Geometry*, 2001, pp. 85-88.

[7] Edelsbrunner H., "Surface reconstruction by wrapping finite point sets in space", *In Ricky Pollack and Eli Goodman Festscrift*, 2002, Aronov B., Basu S., Pach J., M. Sharir S.-V., (Eds.), Springer-Verlag, pp. 379-404.

[8] Giesen J., and John M., "The Flow Complex: A Data Structure for Geometric Modeling", *In Proc. ACM-SIAM Symposium on Discrete Algorithms*, 2003, pp. 285-294.

[9] Levoy M., Pulli K., Curless B., Rusinkiewicz S., Koller D., Pereira L., Ginzton M., Anderson S., Davis J., Ginsberg J., Shade J., and Fulk D., "The Digital Michelangelo Project: 3D Scanning of Large Statues", *In Proc. SIG-GRAPH*, 2000, PP. 131-144.

[10] Ohtake Y., Belyaev A., and Seidel H.-P, "An integrating approach to meshing scattered point data", *In Proc. ACM Symposium on Solid and Physical Modeling*, 2005, pp. 61-69.

[11] Zhao H.-K., Osher S., and Fedkiw R., "Fast Surface Reconstruction using the Level Set Method", *In Proc. IEEE Workshop on Variational and Level Set Methods in Computer Vision (VLSM)*, 2001, pp. 194-202.

[12] Nina Amenta and Marshall Bern, "Surface reconstruction by Voronoi filtering", *In 14th ACM symposium on Computation Geometry*, June 1998.