Surface Reconstruction

Ashok Dhungana, Prabhav Adhikari, Suraj Yadav

National Institute of Technology Calicut

Abstract

Given a finite sampling S of an unknown surface F, Surface Reconstruction is concerned with the calculation of a model of F from S. In this report we outline various existing algorithms for Surface Reconstruction and basics required for these algorithms. In general the basic idea, the algorithm and the complexities for each methods have been discussed.

Keywords: Surface reconstruction, Delaunay Triangulation, Voronoi Diagram, Point sets

1. Preliminaries

1.1. Convex Hull

Given a set of points P, a closed convex polygon that includes all the points in P with the least area.

1.2. Triangulation

Triangulation of a set of points P is the subdivision of the convex hull of the points into simplices such that any two simplices intersect in a common face of any dimension or not at all and such that the set of vertices of the simplices are contained in P. Every point set has a triangulation.

1.3. Delaunay Triangulation

Delaunay triangulation of P is a triangulation of P in which every triangle is Delaunay. A triangle is delaunay if its vertices are in P and its open circumdisk is empty. For a given point set, there always exists a delaunay triangulation, and is unique if and only if no four points in P lie on a common empty circle.

1.4. Voronoi Diagram

The partitioning of a plane with point set P, into convex polygons such that each polygon contains exactly one generating point in P and every point in a given polygon is closer to its generating point than to any other. Voronoi Diagram is dual to the Delaunay triagulation of the point set.

1.5. Combinatorial Map

A (2-dimensional) oriented combinatorial map, or just map, M, is a triple D, θ_0 , θ_1 , where D is a finite set of elements, called darts, θ_0 is an involutory bijection on D, and θ_1 is a bijection on D. We may also assume that θ_0 has no fixed points.

1.6. Gabriel graph

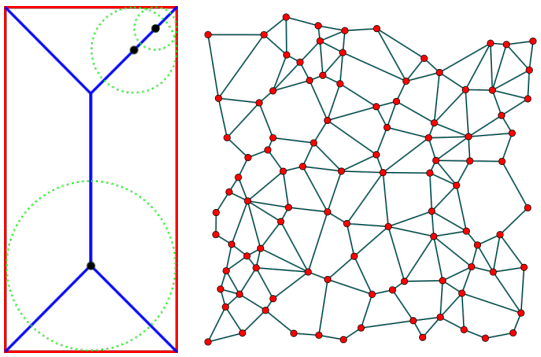
Gabriel graph is the graph with vertex set S in which any points P and Q in S are adjacent precisely if they are distinct and the closed disc (or sphere in $3D$) of which line segment PQ is a diameter contains no other elements of S.

1.7. Medial Axis

The Medial Axis (MA) of subset S bounded by curve C, is defined as the locus of the centers of circles (in $\mathbb{R}^2$), spheres (in $\mathbb{R}^3$), that are tangent to curve C in two or more points, where all such circles (or spheres) are contained in S.

1.8. Medial Axis Transform

The medial axis together with the associated radius of the inscribed discs is called the Medial Axis Transform (MAT).



(a) (b)

Figure 1: (a) MAT (Original Shape-Red, Tangent Circles-Green, MAT-Blue). (b) Gabriel Graph of Random Points.

2. Literature Survey

An early, and influential, attempt to characterize the shape of a set of points is due to [1], which introduced a construction known as "$\alpha - shape$" as a generalization of the convex hull. For a finite set P of points in the plane, the "$\alpha-hull$" for "$\alpha = 0$" is the intersection of all closed discs of radius $1/\alpha$ containing all the points of P (where for negative values of α a closed disk of radius, $1/\alpha$ is interpreted as the complement of an open disk of radius $1/\alpha$). As α approaches 0, the $\alpha - hull$ approaches the ordinary convex hull, and therefore the $0 - hull$ is stipulated

to be the convex hull. The $\alpha - shape$ is a straight-line graph (usually a polygon) derived in a straightforward manner from the $\alpha - hull$. When $\alpha = 0$, this is the convex hull, and for large negative values of α it is P itself.

A related notion, $A - shape$, was introduced in [2]. Given a finite set of points P, and a set A (which evidently needs to be disjoint from P, although the authors do not specify this), we can define the $A - shape$ of P by first constructing the Voronoi diagram for $A \cup P$ and then joining together any pair of points $p, q \in P$ whose Voronoi cells both border each other and border some common Voronoi cell containing a point of A. The edges $p-q$ belong to the Delaunay triangulation of $A \cup P$: they are the "$A - exposed$" edges of the triangulation. An important issue discussed in the paper is how to choose A so that the $A - shape$ of P is "adequate." In a later paper [3], the $A - shape$ is used as the basis for an "onion-peeling" method, by analogy with the popular convex onion-peeling method for organizing a set of points and extracting a "central" embedded convex shape from them [4].

Two rather different constructs, $r - shape$ and $s - shape$, were defined in [5] as follows. The initial set of points P is assumed to be a dot pattern, that is, a planar point set whose elements are clearly visible as well as fairly densely and more or less evenly distributed. To obtain the $s - shape$, the plane is partitioned into a lattice of square cells of side-length s. The $s - shape$ is simply the union of lattice cells containing points of P. The authors suggest a procedure for optimizing the value of s so that the $s - shape$ best approximates the perceived shape of the dot pattern. For the $r - shape$, they first construct the union of all disks of radius r centered on points of P. For points $p, q \in P$, the edge $p - q$ is selected if and only if the boundaries of the disks centered on p and q intersect in a point which lies on the boundary of the union of all the disks. The $r - shape$ of P is the union of the selected edges, and the authors show that this can be computed in time $O(n)$, where n is the cardinality of P. They note that the $r - shape$ is a subgraph of the $\alpha - shape$ in the sense of [1]. Regarding the selection of r, they note that "to get a perceptually acceptable shape, a suitable value of r should be chosen, and there is no closed form solution to this problem," and that moreover "perceptual structure of P ... will vary from one person to another to a small extent."

The use of Voronoi diagrams for constructing regions from point-sets has also been advocated in the context of GIS [6]. In this context, the set P consists of points known to be in a certain region, for which an approximation to the boundary is required. It is assumed that in addition to P another point-set P' is given, consisting of points known to lie outside the region to be approximated. From the Voronoi diagram for $P \cup P'$, the method simply selects the union of the Voronoi cells containing points of P. The resulting shape differs from the characteristic shapes constructed in this paper in that the original point-set lies entirely in its interior. Depending on ones purposes, this feature may either be desirable or undesirable.

A similar method [7] is based on Delaunay triangulations. Given sets P and P' as before, the Delaunay triangulation of $P \cup P'$ is constructed, and then the midpoint of every edge which joins a point in P to a point in P' is selected. The final region is produced by joining all pairs of selected midpoints belonging to edges of the same triangle.

3. Detailed Study of Some papers

3.1. Recursive Voids for Identifying a Nonconvex Boundary of a Set of Points in the Plane [8]

A method that identifies the boundary of a nonconvex shape of a set of points,P in the plane is introduced in this paper. The boundary of the shape is explored through finding empty regions recursively within a shell that encapsulates all of the points. This algorithm is output sensitive and runs in linear $O(ln)$ time determined by the output parameter l, which is proportional to the length of the nonconvex boundary measured by a threshold unit distance. The recursive nature of the algorithm allows a tree structure that characterizes the empty regions, and by complementarity, the nonconvex shape itself.

This work falls under the non-Delaunay methods.The algorithm focuses on identifying the empty regions within a shell H that encapsulates P, such as the convex hull of P. In exploring these empty regions, a unique boundary is defined between the points in P and a given empty region, with respect to an edge E, such that this boundary together with edge E define an empty simple polygon, which is called a void. Then, recursively new voids are created by exploring edges on the boundary that are greater than or equal to some threshold t.

3.2. Efficient generation of simple polygons for characterizing the shape of a set of points in the plane [9]

For a finite set of at least three points in the Cartesian plane $P \subset \mathbb{R}^2$, the characteristic shape algorithm yields a possibly non-convex area with a shape that characterizes the distribution of the input point set. All the sets under consideration in this paper are sets of points in the Cartesian plane R^2, and these sets are assumed to be finite.

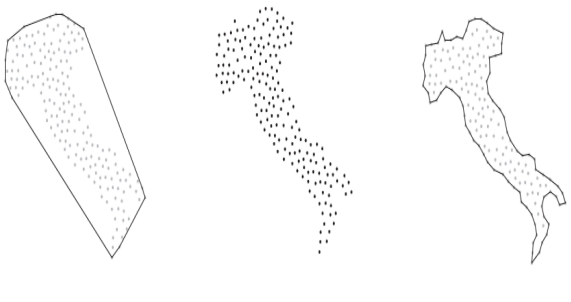
The χ-shape produced by the algorithm has the properties that:

1. it is a simple polygon;
2. it contains all the points of P ; and
3. it bounds an area contained within and possibly equal to the convex hull of the points of P.

- **Idea** The χ-shape algorithm is based on "shaving" exterior edges (edges that bound only one triangle) from a triangulation of the input point set in order of the length of edges and subject to a regularity constraint. The algorithm itself has a time complexity of $O(n \log n)$, where n is the number of input points.

- **Algorithm** The algorithm can be summarized as comprising the following steps for an input point set P and a length parameter l:

    1. Generate the **Delaunay triangulation**, τ of the set of input points P;



(a) (b) (c)

Figure 2: (a) Convex Hull of P. (b) Point Set P. (c) Characteristic χ shape of P.

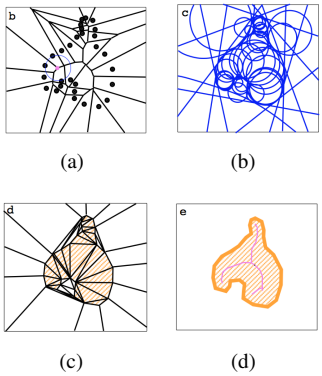
2. Construct a priority list of boundary edges B. The checking for boundary edge can be done using idea from **Combinatorial Map**.
3. Remove the longest boundary edge from the triangulation, τ such that:
   (a) the edge to be removed is longer than the length parameter l; and
   (b) the exterior edges of the resulting triangulation form the boundary of a simple polygon and the triangulation, τ is regular;
4. Repeat 2. as long as there are more edges to be removed
5. Return the polygon formed by the exterior edges of the triangulation

3.3. Power Crust

Power Crust [10] is a construction which takes a sample of points from the surface of a three-dimensional object and produces a surface mesh and an approximate medial axis.

- **Idea** The approach is to first approximate the medial axis transform (MAT) of the object, then use an inverse transform to produce the surface representation from the MAT.
- **Algorithm**
  Let F be the original surface and S the sampled points from the surface.
  1. Compute the Voronoi diagram of the sample points S.
  2. For each sample point s, compute its poles. Poles of a sample point $s \in S$ are the farthest vertex of its Voronoi cell in the interior and the of exterior of F.
  3. Compute the power diagram (a weighted Voronoi diagram using input parameter ρ) of the poles.
  4. Label each pole either inside or outside F.
  5. Output the power diagram faces separating the cells of inside and outside poles as the **Power Crust**.
  6. Output the regular triangulation faces connecting inside poles as the **Power Shape**.

  The **Power Shape** approximates the **MAT** of F, while **Power Crust** approximates the F.



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

Figure 3: (a) The Voronoi diagram of S, with the net ball surrounding one pole shown. (b) The inner and outer polar balls. (c) The power diagram cells of the poles. (d) The power crust and the power shape.

3.4. Shape-Hull

Shape-Hull[11] is a non-parametric, single stage Delaunay sculpting algorithm which reconstructs topologically correct piece-wise linear approximation for divergent concave surfaces.

- **Idea** For a given set of Points $S \in \mathbb{R}^2$ S is sampled from a divergent concave curve $\sum_D$, a proximity graph called as shape-hull graph $SHG(S)$ is defined that contains all Gabriel edges and a few non-Gabriel edges of the Delaunay triangulation of S. Shape-hull $SH(S)$ is defined as the boundary of Shape-hull graph $SHG(S)$. Under dense sampling assumption it can be shown that $SH(S)$ represents a piece-wise linear approximation of $\sum_D$.
- **Algorithm**
  We define a tetrahedral T to be **deletable** if either
  1. It has only one face f on the boundary and the vertex opposite f is not on the boundary
  2. It has exactly two face $f_1 and f_2$ on the boundary and the edge opposite to common edge of $f_1 and f_2$ is not on the boundary.

  Let S be the sampled points set.
  1. Set $D = Del(S)$ (Delaunay Triangulation of S).
  2. Set PQ = Priority Queue, containing **deletable** boundary tetrahedral of D, sorted in the descending order of their circumradii.
  3. While PQ is not empty
     (a) $T = pop(PQ)$.
     (b) If T is **deletable** and circumcenter of T lies outside the boundary of D **then**
     – delete T from D
     – add the **deletable** neighbors of T to PQ.
  4. Finally D is reconstructed surface.
- **Complexity**
  In $\mathbb{R}^3$, Delaunay triangulation takes $O(n^2 \log n)$ in the worst case[12]. Let t_c = number of Delaunay tetrahedral in the Convex hull of S and t_d = number of tetrahedral in

the reconstructed surface. Each push/pop operation take $O(\log n)$ time, while the boundary checking and deletion of tetrahedral can be done in constant time. Thus for a given $Del(S)$, it takes $kO(\log n)$ time to reconstruct the surface. For surfaces having large concave portion, k tends to be large and as a result, the reconstruction time will be substantially increased. However, the worst case time complexity of shape-hull algorithm is $O(n^2 \log n)$, dominated by $Del(S)$ construction.

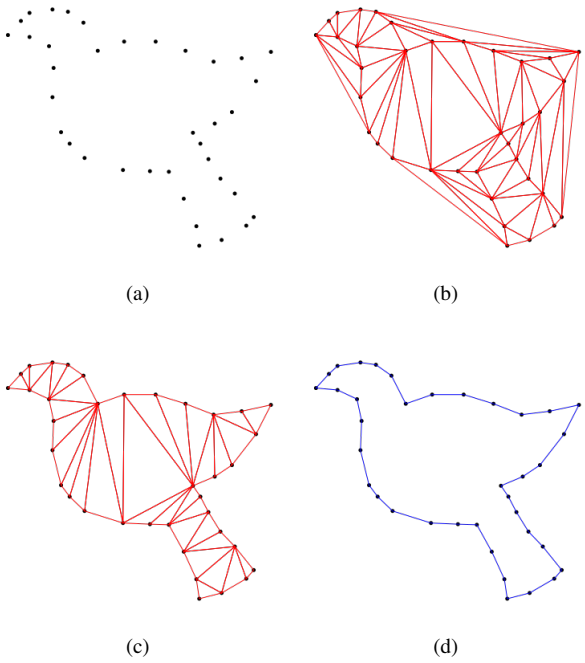


Figure 4: (a) Sample Points. (b) Delaunay Triangulation. (c) Shape-hull graph. (d) Shape-hull.

4. Description of Algorithm

We start with Delaunay tetrahedralization of available points in $\mathbb{R}^3$. We use the 3D Delaunay to obtain graph DT, with edges as sides of tetrahedrals in Delaunay. Minimum spanning tree for DT is obtained using Kruskal's algorithm.

Score of a triangle is max of cosine of each angle in a triangle

Algorithm 1: updatePQ

for all $edge[u,v] \in edgesCovered$ **do**
  $commonPoints \leftarrow$ {adj verts of u} $\cup$ {adj verts of v} $\setminus\{u,v\}$
  for all $points \in commonPoints$ **do**
    $score \leftarrow getScore(edge, point)$
    if $[u,v,point] \in DT \wedge$ satisfies edge condition **then**
      $pq.insert(score, edge, point)$
    end if
  end for
end for

Algorithm 2: validToAdd(face)

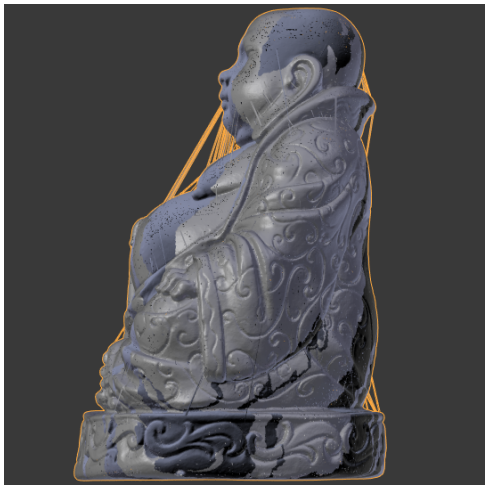
for all $edge \in face$ **do**
  if $deg(edge) = 2$ **then**
    return $false$
  end if
  if $deg(edge) = 1 \wedge$ angle between this face and another face with same edge < 90 **then**
    return $false$
  end if
end for
return $true$

Algorithm 3: reconstruct(points3D,DT3,MSTdges)

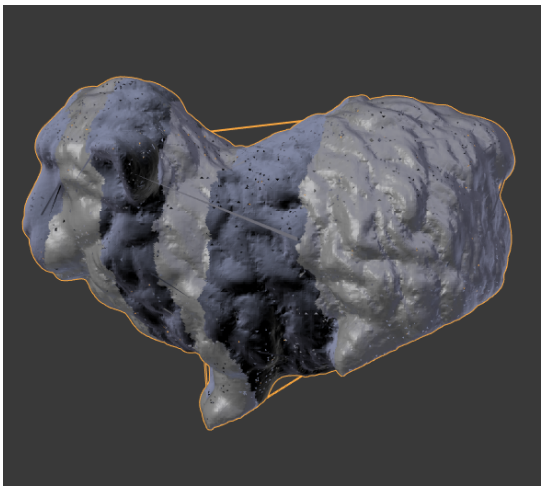
$adjList \leftarrow getAdjList(MSTedges)$
$edgeDegree \leftarrow$ adj list of each edge
$trianglesCovered \leftarrow$ set of covered faces
$edgesCovered \leftarrow MSTEdges$
$pq \leftarrow createPQ(key = score\_of\_face, value =$ $\{edge, oppositepoint\})$
$edgeCondition \leftarrow 2$
$updatePQ()$
while $edgeCondtition \geq 0$ **do**
  while $pq \neq \Phi$ **do**
    $face \leftarrow pq.pop()$
    if $deg(edge \in face) = 2$ **then**
      $continue$
    end if
    $count \leftarrow$ count of edges in $face$ longer than maxlength in current graph
    if $face \notin trianglesCovered \wedge validToAdd(face) \wedge$ $count > edgeCondition$ **then**
      add two new edges to $adjList$ and update $edgeDegree$
      add new face to $trianglesCovered$
      update pq by adding possible faces obtained by adding new edges
    end if
  end while
  $edgeCondition \leftarrow edgeCondition - 1$
  $holes \leftarrow$ subgraph of current surface with degree 1 edges
  for all $vertex \in holes$ **do**
    $newEdges.add($minimum length edge(in holes) from vertex that is a delaunay edge$)$
  end for
  $updatePQ(newEdges)$
end while
return $trianglesCovered$

5. Current Output

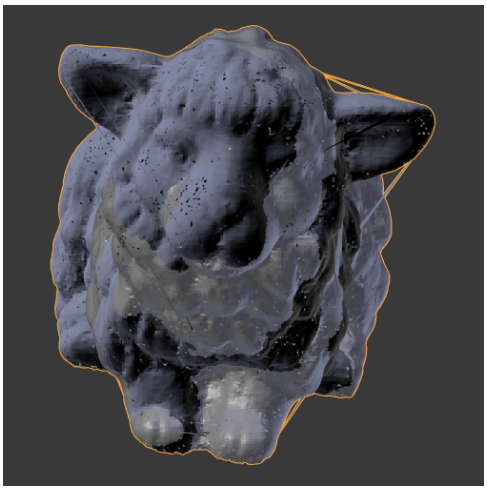
(a)



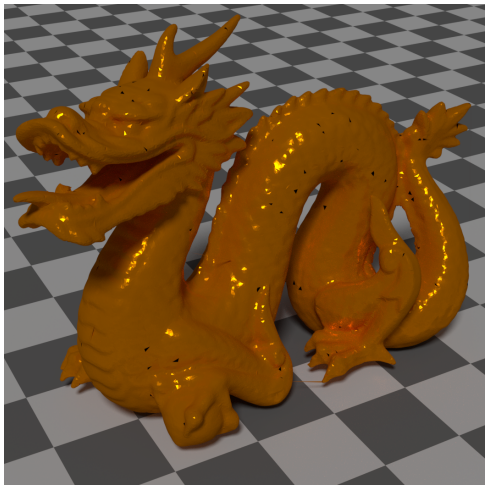
(b)



(c)



(d)



(e)

Figure 5: (a),(b) Reconstructed Model with 757490 vertices and (c),(d) with 159312 vertices. (e) Rendered Image of a reconstructed surface with 100250 vertices.

References

[1] H. Edelsbrunner, D. Kirkpatrick, R. Seidel, On the shape of a set of points in the plane, IEEE Trans. Inf. Theor. 29 (4) (2006) 551–559. doi:10.1109/TIT.1983.1056714.
URL http://dx.doi.org/10.1109/TIT.1983.1056714

[2] M. Melkemi, M. Djebali, Computing the shape of a planar points set, Pattern Recognition 33 (9) (2000) 1423–1436. doi:10.1016/s0031-3203(99)00124-7.
URL http://dx.doi.org/10.1016/s0031-3203(99)00124-7

[3] J. M. Fadili, M. Melkemi, A. Elmoataz, Non-convex onion peeling using a shape hull algorithm, Pattern Recognition Letters 25 (14) (2004) 1577–1585. doi:10.1016/j.patrec.2004.05.015.
URL https://hal.archives-ouvertes.fr/hal-01123869

[4] B. Chazelle, On the convex layers of a planar set, IEEE Transactions on Information Theory 31 (4) (1985) 509–517. doi:10.1109/TIT.1985.1057060.

[5] A. R. Chaudhuri, B. B. Chaudhuri, S. K. Parui, A novel approach to computation of the shape of a dot pattern and extraction of its perceptual border, Comput. Vis. Image Underst. 68 (3) (1997) 257–275. doi:10.1006/cviu.1997.0550.
URL http://dx.doi.org/10.1006/cviu.1997.0550

[6] H. Alani, C. B. Jones, D. Tudhope, Voronoi-based region approximation for geographical information retrieval with gazetteers, International Journal of Geographical Information Science 15 (4) (2001) 287–306.

[7] A. Arampatzis, M. van Kreveld, I. Reinbacher, C. B. Jones, S. Vaid, P. Clough, H. Joho, M. Sanderson, M. Benkert, A. Wolff, Web-Based Delineation of Imprecise Regions.

[8] O. Åeref, C. W. Zobel, Recursive voids for identifying a nonconvex boundary of a set of points in the plane, Pattern Recognition 46 (12) (2013) 3288 – 3299. doi:http://dx.doi.org/10.1016/j.patcog.2013.05.013.
URL http://www.sciencedirect.com/science/article/pii/S0031320313002173

[9] M. Duckham, L. Kulik, M. Worboys, A. Galton, Efficient generation of simple polygons for characterizing the shape of a set of points in the plane, Pattern Recogn. 41 (10) (2008) 3224–3236. doi:10.1016/j.patcog.2008.03.023.
URL http://dx.doi.org/10.1016/j.patcog.2008.03.023

[10] N. Amenta, S. Choi, R. K. Kolluri, The power crust, in: Proceedings of the Sixth ACM Symposium on Solid Modeling and Applications, SMA '01, ACM, New York, NY, USA, 2001, pp. 249–266. doi:10.1145/376957.376986.
URL http://doi.acm.org/10.1145/376957.376986

[11] J. Peethambaran, R. Muthuganapathy, Reconstruction of water-tight surfaces through delaunay sculpting, Comput. Aided Des. 58 (C) (2015) 62–72. doi:10.1016/j.cad.2014.08.021.
URL http://dx.doi.org/10.1016/j.cad.2014.08.021

[12] J.-D. Boissonnat, Geometric structures for three-dimensional shape representation, ACM Trans. Graph. 3 (4) (1984) 266–286. doi:10.1145/357346.357349.
URL http://doi.acm.org/10.1145/357346.357349