A Constrained Delaunay Triangle Mesh Method for Three-Dimensional Unstructured Boundary Point Cloud

Qian Huang[1], Thomas Wischgoll[2]
[1,2]Computer Science and Engineering Department,
Wright State University, Dayton, OH, 45435
huang.11@wright.edu

Abstract
A constrained Delaunay triangle mesh method is presented to recover the surface from the three-dimensional unstructured boundary point cloud. The surfaces of different three-dimensional object models are recovered by this triangle mesh method. The radius of the tested cylinder model can be accurately estimated from the derived maximum principle curvature. The derived minimum principle curvature of the tested cylinder model displays a zero value, which is the analytical value of a standard cylinder model's minimum principle curvature.

Keyword: *constrained Delaunay triangle mesh, advancing front, unstructured point cloud in three dimensions*

I. INTRODUCTION

The surface of a three-dimensional object discussed in this work is a two-dimensional manifold. It is also a surface that is compacted, connected, orientable and can be triangulated. The object's surface can be approximated with piecewise connected triangles. A real object surface with triangulation rendering is faster than ray tracing rendering with a modern graphics card. Delaunay triangulation connects points with empty circumcircle or circumsphere property, it is often applied to recover a surface from a three-dimensional point cloud of an object [1], [2], [3], [4]. In Delaunay triangulation [5], a simplex is a tetrahedron, a triangle, an edge or a vertex. All the simplices are Delaunay. There exists a circumcircle or circumsphere of each simplex, no other vertices are in inside this circumcircle or circumsphere. To respect every input boundary point and segment, Steiner points are often to be added with Delaunay triangulation. In a constrained Delaunay triangulation, a simplex must be a constrained Delaunay that is a relax definition of Delaunay, the input set of boundary points and segments must be respected, and no Steiner points are added.

The advancing front technique is often used to generate unstructured mesh [6], [7]. The advantages of the advancing front technique are that a new triangle is formed locally from an existing edge, the third point is optimally located from all nearby front points, boundary integrity is guaranteed since the boundary discretization forms the initial front.

A constrained Delaunay triangulation (CDT) meshing procedure with advancing front scheme is presented in this work to recover a surface from the unstructured point cloud in three dimensions without Steiner points.

II. RULES OF THE CDT MESH METHOD

A. *The Definition of CDT in Three Dimensions*

In this work, the CDT mesh method is to recover a surface from the surface boundary point cloud of real data and without new points being interpolated. Based on the advancing front technique, from triangles' edges, the constrained Delaunay triangles are built among the three-dimensional unstructured boundary point cloud of real data. The definitions of CDTs in [5], [8], [9], [10] mean that the input set of boundary points and segments must be respected, no Steiner points are added, there exists a circumcircle or circumsphere for each simplex that enclose no other points or segments that are visible from any point in the interior of the simplex. That is a "truncated empty circumcircle or circumsphere" criterion. A modified "truncated empty circumsphere" criterion is presented here. Any neighboring points or existing triangle edges cannot be projected inside any triangles. The boundary points' normals and their reverse normals are taken as the projection directions onto the plane defined by the triangle. A point in proximity of an existing triangle edge's midpoint can be a possible third vertex of the new triangles. The proximity region is a sphere with a user defined radius. Then from all the possible triangles satisfying this definition of CDT, the third vertex is chosen as the one that is the closest to the midpoint of the existing triangle edge. Thus this new triangle can be a CDT. Each edge of a triangle connects at most two triangles. The following describes several situations to decide a projected edge intersecting a triangle or not.

B. *Identification of a Projected Edge Intersecting a Triangle or not*

To an existing edge, its end points are projected onto the new triangle $\Delta p1p2p3$ plane along their normal rays, respectively. Let $p4$ and $p5$ be the two projected end points of a computing edge on $\Delta p1p2p3$ plane. The Fig. 1 delineates the positions of $p4$ and $p5$ between $\Delta p1p2p3$, for each case, line segment $p4p5$ intersecting $\Delta p1p2p3$ or not is defined. Edge $p2p3$ of $\Delta p1p2p3$ is as the general case for all the figures in Fig 1.

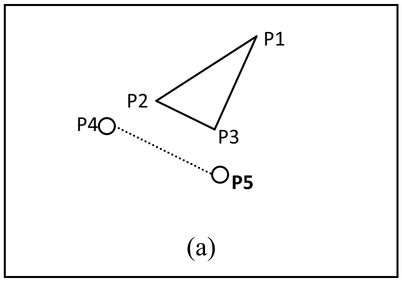


Fig 1. Identify a projected edge intersects a triangle or not (a)

1) According to the normal of $\Delta p1p2p3$, Fig. 1 (a) is the case that $p4$ and $p5$ are both on the side of one triangle edge $p2p3$, which has no $\Delta p1p2p3$, then line segment $p4p5$ does not intersect $\Delta p1p2p3$.

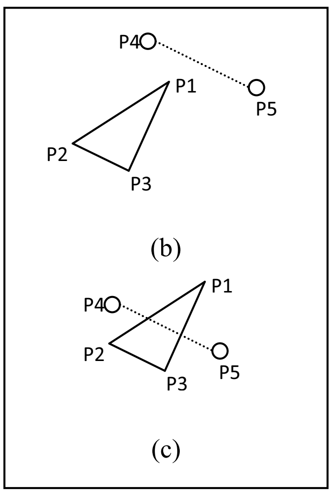


Fig 1. Identify a projected edge intersects a triangle or not (b) and (c)

2) In Fig. 1 (b) and (c), $p4$ and $p5$ are both on the side of edge $p2p3$, which includes $\Delta p1p2p3$, they are on the different side of the other two triangle edges, respectively. The normal of $\Delta p1p2p3$ is $\vec{n} = (p2 - p1) \times (p3 - p1)$, edge $p1p2$ is the first item of this vector product. $p4$ is at the side of edge $p1p2$, which has no $\Delta p1p2p3$, $p5$ is on the side of edge $p1p2$, which has $\Delta p1p2p3$. A virtual triangle $\Delta p1p4p5$ is composed, its normal is $\vec{n}_{virtual} = (p4 - p1) \times (p5 - p1)$. $p4$ substitutes the position of $p2$ and $p5$ substitutes the position of $p3$. If $\vec{n} \cdot \vec{n}_{virtual} \leq 0$ as Fig. 1(b), line segment $p4p5$ is not intersecting $\Delta p1p2p3$. If $\vec{n} \cdot \vec{n}_{virtual} > 0$ as Fig. 1(c), line segment $p4p5$ is intersecting $\Delta p1p2p3$.

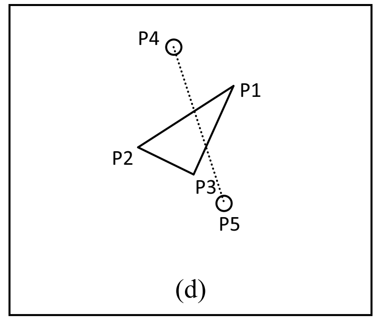


Fig 1. Identify a projected edge intersects a triangle or not (d)

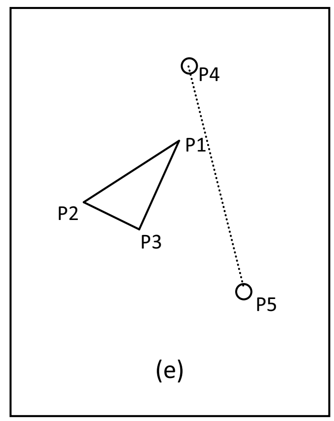


Fig 1. Identify a projected edge intersects a triangle or not (e)

3) In Fig. 1(d), $p4$ and $p5$ are on the different side of the three triangle edges, respectively. As the method in Fig. 1(b) and Fig. 1(c), line segment $p4p5$ composes three virtual triangles with three vertices of $\Delta p1p2p3$, since three normals of the new triangle at three vertices are $\vec{n} = (p2 - p1) \times (p3 - p1)$, $\vec{n} = (p3 - p2) \times (p1 - p2)$ and $\vec{n} = (p1 - p3) \times (p2 - p3)$. In the normal equations above, positions of vertices are arranged in circular shift manner. Three normals of its virtual triangles are $\vec{n}_{virtual,1} = (p4 - p1) \times (p5 - p1)$, $\vec{n}_{virtual,2} = (p5 - p2) \times (p4 - p2)$ and $\vec{n}_{virtual,3} = (p5 - p3) \times (p4 - p3)$. If all three dot products are greater than zero, $\vec{n} \cdot \vec{n}_{virtual,1} > 0$, $\vec{n} \cdot \vec{n}_{virtual,2} > 0$ and $\vec{n} \cdot \vec{n}_{virtual,3} > 0$ at the same time as Fig. 1(d), line segment $p4p5$ intersects $\Delta p1p2p3$. If one of three dot products is less than zero, $\vec{n} \cdot \vec{n}_{virtual,1} < 0$, or $\vec{n} \cdot \vec{n}_{virtual,2} < 0$, or $\vec{n} \cdot \vec{n}_{virtual,3} < 0$ as Fig. 1(e), line segment $p4p5$ does not intersect $\Delta p1p2p3$.

4) One of three dot products can be equal to zero, $\vec{n} \cdot \vec{n}_{virtual,1} = 0$, or $\vec{n} \cdot \vec{n}_{virtual,2} = 0$, or $\vec{n} \cdot \vec{n}_{virtual,3} = 0$ as Fig. 1(f) and Fig. 1(g). For example, if $\vec{n} \cdot \vec{n}_{virtual,3} = 0$, and $p4$ or $p5$ resides at the side of edge $p2p3$ with $\Delta p1p2p3$, and the side of edge $p1p3$ with $\Delta p1p2p3$ at the same time, line segment $p4p5$ will intersect $\Delta p1p2p3$, as Fig. 1(f). If none of $p4$ and $p5$ resides at the side of edge $p2p3$ with $\Delta p1p2p3$, and the side of edge $p1p3$ with $\Delta p1p2p3$ at the same time, line segment $p4p5$ will not intersect $\Delta p1p2p3$, as Fig. 1(g).

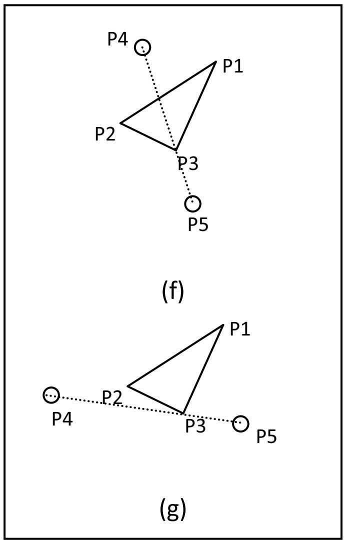


Fig 1. Identify a projected edge intersects a triangle or not (f) and (g)

C. Requirements of the Third Point to a CDT

Since the triangle is only a small surface patch, the normals' directions of the triangle's vertices are very similar, and they have to be conformed to the triangle's normal direction. The following conditions have to be satisfied.

1) For a given edge of a triangle, dot products of its endpoints' normals with that of the new triangle's third point are required to be positive.

2) The unit normal vectors of the new triangle's three vertices are dot product with the new triangle's unit normal, and the obtained absolute values are all greater than a positive threshold value to avoid dangling triangle which does not closely approximate a surface patch.

3) Each triangle edge can only connect two triangles. The edge plane is defined as the plane perpendicular to a plane containing at least one of the two triangles and passing the edge which is forming a new triangle. The triangles' third vertices of these two triangles must be on the two sides of this edge plane separately except that the normals of both the third vertices are in reverse direction as in Fig. 2 (b). Fig. 2(a) and Fig. 2(b) are possible CDTs, the new triangle in Fig. 2(c) does not satisfy CDT requirements.

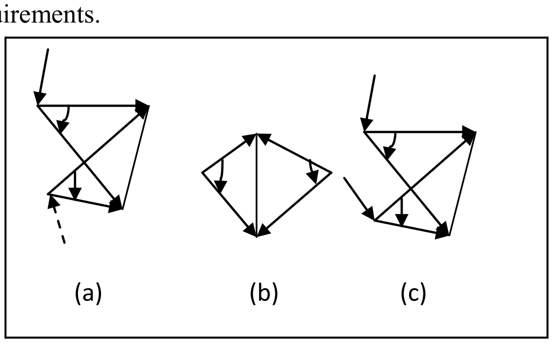


Fig. 2 Cases of the third point of the new triangle

D. Determinations of the Neighboring Points or Edges Intersecting the New Triangle

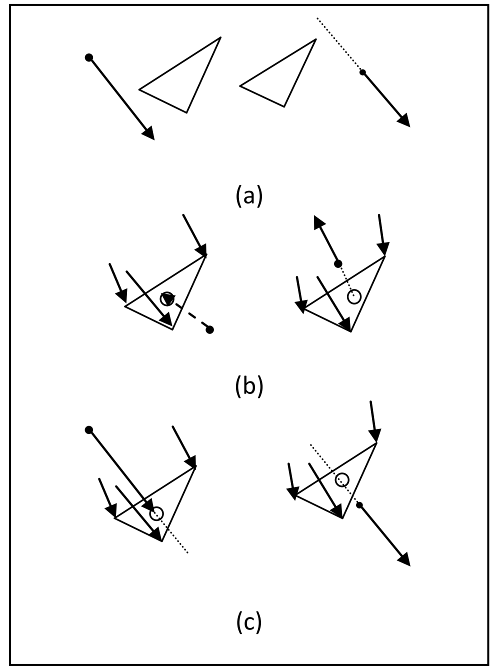


Fig. 3 Cases of neighboring points intersecting a triangle or not (a), (b) and (c)

The CDT mesh method looks at whether the normal ray of a boundary point intersects the new triangle or not. The following procedures are the implementations of the CDT definition's requirements, and must be satisfied at the same time to allow a CDT to be built.

1) The projection of a neighboring point is along its normal direction onto the plane of the new triangle, if inside the new triangle, the new triangle is not a CDT. If it is outside the new triangle, this triangle is a possible CDT, two instances are shown in Fig 3 (a). Fig. 3 (b) is a possible CDT since the dot product of the solid neighboring point's normal with one of the triangle vertices' normal is negative. Fig. 3 (c) is not a valid CDT since the dot product of the solid point's normal with all the triangle vertices' normals are positive. The small circles are indicating intersections of normal rays with triangles.

2) Edge projection is that its endpoints' normals or the reverse direction of the normals intersect on the plane of a formed or a new triangle. If no projected neighboring edge intersects a new triangle, that new triangle is a possible CDT. Three examples are shown in Fig. 4 (a). Fig. 4 (b) provides cases of intersections between the projected neighboring edge and the possible CDT since the dot product of the average edge endpoints' normals with the average of triangle vertices' normals is negative. Fig. 4 (c) lists cases of invalid CDT, since the dot product of the average edge endpoints' normals with the average of triangle's vertices normals is positive, The dashed line segment with two circles is the projected neighboring edge on the triangle plane. The solid arrow lines are normals of the solid endpoints of a neighboring triangle edge. The dash lines attached with the solid points are extensions of the normal directions reversely. Also the normal ray propagation distance cannot exceed a user defined threshold.

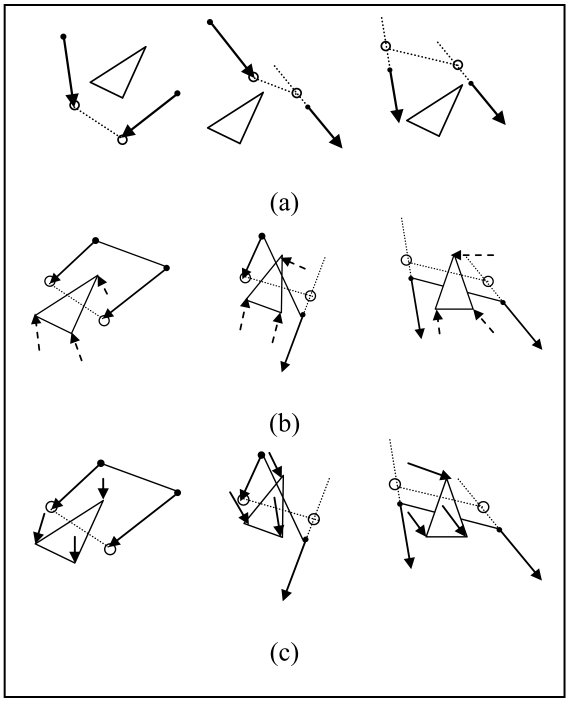


Fig. 4 Cases of neighboring edge intersecting a triangle or not (a) , (b) and (c)

E. Avoidance the Degenerate Triangle

The triangle area cannot be too small. The triangle area is required to be greater than a user defined small positive value. If the triangle area is too small, it perhaps implies a degenerate case that three vertices of the triangle are on the same line.

III. CDT MESH METHOD STEPS

A. Seed Triangles' Generation

Starting with a seed point from the boundary point cloud if it has no triangle connected, find the closest point to this point to build one edge of a seed triangle. Then find the third point that is closest to the midpoint of this edge just built. The resulted triangle must satisfy all the requirements in this CDT mesh method. This step will continue until all the boundary points are computed.

B. Advancing Front Step

For every triangle formed, each of its three edges must be investigated. If an edge has two triangles connected, continue to test the next edge. If the edge has only one triangle, it belongs to the front elements, the third point of a new triangle is required to select among the neighboring points of this edge's midpoint. Then from all the triangles satisfying all requirements of a CDT, the one that the third vertex is closest to the midpoint of the current triangle edge is selected. Then the current edge is removed from front. This step will continue until all formed triangles' edges are inspected.

C. Hole filling step

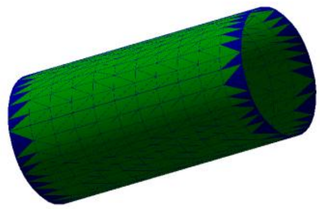
Sometimes holes in the mesh occur. A hole triangle means at least one edge of the triangle connects only one triangle. The holes produced in the mesh are filled by triangles with less strict criteria of CDT described above; reduce the proximity neighboring region and remove the requirement of II.C.2) that is to remove the normal requirement of the new triangle to avoid the occurrence of dangling triangle, or fill a hole by forming a triangle with the adjacent edges at a common vertex which form a hole.

IV. THE PERFORMANCES OF THE CDT MESH METHOD ON THREE-DIMENSIONAL MODELS

A. Render a Three-Dimensional Surface with the CDT Mesh

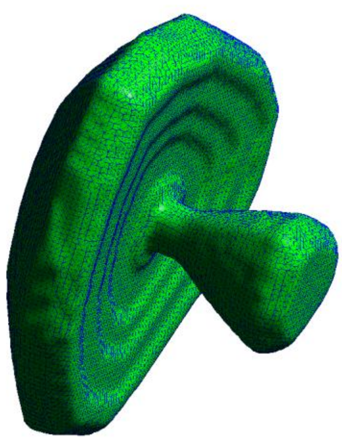
The surface recoveries are performed on three-dimensional point clouds from several different objects of real data: a cylinder, a mushroom, a monster, and a heart's artery tree. The cylinder model is a stack of a same circle, the boundary points of the circle is extracted from a real data. Fig. 5(a), Fig. 5(b), Fig. 5(c) and Fig. 5(d) display the surface recovered by this CDT mesh method on objects cylinder, mushroom, monster and an artery tree, and no Steiner points are added.

After the CDT mesh operation, the hole occurrences can be identified by testing each built triangle. If a triangle is adjacent to a hole, at least one of its edges connects only this current triangle. Such a triangle is called a hole triangle. If a triangle is not adjacent to a hole, its three edges must connect to other triangles beside the current one. If the surface of the object has no boundary, no hole triangle should appear on the recovered surface. The number of hole triangles indicates how many boundaries are generated by this CDT mesh method. The number of boundary points, the total triangles, triangles created from hole filling step and hole triangles are listed in the Table 1 for each object. For the cylinder model, 72 hole triangles are at the boundary of the cylinder's surface, in Fig. 5(a). The CDT mesh produces no hole triangles with mushroom, monster model. It produces false boundaries with the artery tree model with nine hole triangles occurring. Fig. 5 (e) displays a large view of several hole triangles on CDT meshed surface of the artery tree model.



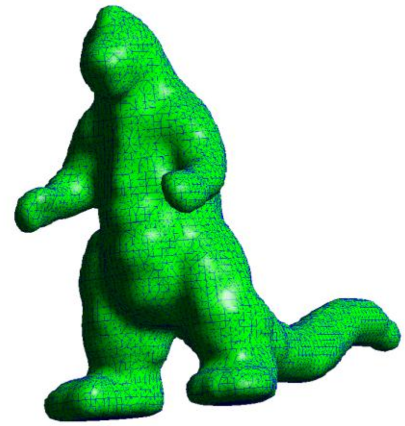
(a) cylinder

Fig. 5 CDT Meshed Surfaces on Three-Dimensional Models (a) cylinder



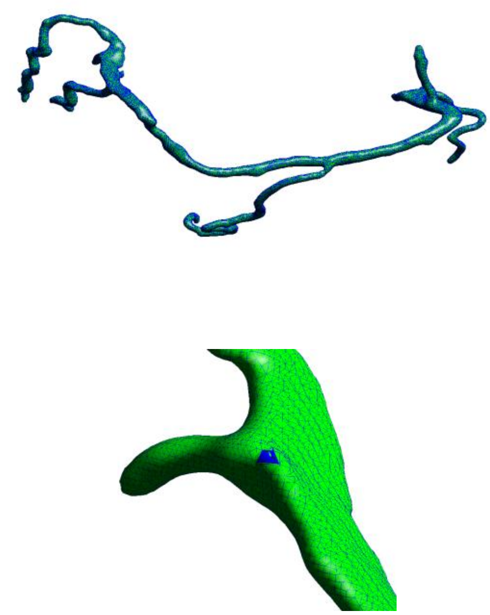
(b) mushroom

Fig. 5 CDT Meshed Surfaces on Three-Dimensional Models (b) mushroom



(c) monster

Fig. 5 CDT Meshed Surfaces on Three-Dimensional Models (c) monster



(d) artery tree (e) large view of the hole triangles in (d).

Fig. 5 CDT Meshed Surfaces on Three-Dimensional Models (d) artery tree (e) large view of the hole triangles in (d).

Table 1. The Performances of the CDT mesh on the three-dimensional models

3D model	point number	Total triangle number	Triangle number by hole filling	Hole triangle number
cylinder	432	792	0	72
mushroom	16008	32012	0	0
monster	7259	14514	31	0
artery tree	27008	54008	300	9

B. Estimations of the Recovered Surface's Normal Angle on Different Models

From the CDT meshed object's surface, the angle between the boundary point's normal and the surface triangle mesh can be estimated through computing the average angle between the normal at each boundary point and the triangle planes this point attached. The statistics of the angle values of different three-dimensional models are given in Table 2. The angle more approaches to 90^0 indicates the triangles are averagely more close to the tangent planes defined by the normals of the objects' boundary points. The mean normal angle of the cylinder model is more approach to 90^0, compared with other models.

Table 2 Normal angles' statistical properties on the objects' CDT meshed surfaces

3D model	Range, degree	Statistics, degree
	max, min	mean, deviation
Cylinder	86.10, 83.35	85.0, 0.79
Mushroom	90.0, 52.66	78.85, 7.18
Monster	90.0, 28.04	75.64, 6.59
Artery tree	89.43, 7.17	71.96, 6.97

C. Estimations of the Recovered Surface's Principle Curvatures of a Cylinder Model

From the CDT meshed object's surface, principle curvatures can be estimated. According to the curvature computation in [11], the principal curvatures can be derived from the local hessian matrix at each boundary points. In Table 3, the computed minimum principle curvature from the CDT mesh is equal to the analytical value zero at each boundary point on the cylinder model. The analytical value of the maximum principle curvature of cylinder is a constant all over the cylinder's surface, which equals to the reciprocal of the cylinder's radius. The mean maximum principal curvature is $0.812mm^{-1}$, standard deviation is $0.0012mm^{-1}$. Thus, the mean radius of this cylinder is 1.231mm, the real radius of this cylinder measured from the sampled boundary points is 1.223mm, and so the absolute error is 0.0078mm.

Table 3.Estimated principle curvatures of cylinder model

Principle curvature	Range, mm^{-1}	Statistics, mm^{-1}
	Max, min	Mean, deviation
Maximum	0.814, 0.811	0.812, 0.0012
Minimum	0.0, 0.0	0.0, 0.0

CONCLUSION

The CDT mesh method presented in this work can successfully recover the surfaces of different three-dimensional object models. The principle curvatures can be derived from the triangulated surfaces. The derived maximum principle curvature can be used to accurately derive the cylinder radius. The derived minimum principle curvature of the tested cylinder model displays a zero value, which is the analytical value of a standard cylinder model's minimum principle curvature.

REFERENCES

[1] Herbert Edelsbrunner, Ernst P. Mücke, Three Dimensional Alpha Shapes, *ACM Transactions on Graphics*, 13(1), 1994, pp. 43–72.

[2] Nina Amenta, Sunghee Choi, Ravi Krishna Kolluri, The Power Crust, *Proceedings of the sixth ACM symposium on Solid modeling and applications*, 2001, pp.249-266.

[3] Tamal K. Dey, Samrat Goswami, Tight cocone: A Water-tight Surface Reconstructor, *Journal of Computing and Information Science in Engineering*, 3(4), 2003, pp.302-307.

[4] M Gopi, S. Krishnan, C. T. Silva, Surface reconstruction based on lower dimensional localized Delaunay triangulation, *Eurographics 2000/M.* Gross and F.R.A. Hopgood, 19(3), 2000, pp. 467-478.

[5] Jonathan Richard Shewchuk: General-Dimensional Constrained Delaunay and Constrained Regular Triangulations, I: Combinatorial Properties. *Discrete & Computational Geometry* 39(1-3): 580-637 (2008)

[6] F. Bernardini, J.Mittleman, H. Rushmeier, C. Silva, and G. Taubin, The ball-pivoting algorithm for surface reconstruction, *IEEE Transactions on Visualization and Computer Graphics*, 5 (4), 1999, pp. 349-359

[7] Dimitri J. Mavriplis. An Advancing Front Delaunay Triangulation Algorithm Designed for Robustness. *Journal of Computational Physics*, 117:90-101, 1995.

[8] Der-Tsai Lee and A. K. Lin, Generalized Delaunay Triangulations for Planar Graphs, *Discrete & Computational Geometry*, 1, 1986, pp. 201-217.

[9] L. Paul Chew, Constrained Delaunay Triangulations, *Algorithmica*, 4(1) , 1989, pp. 97-108.

[10] Thomas C. Kao, David M. Mount, Incremental Construction and Dynamic Maintenance of Constrained Delaunay Triangulations, *Proceedings of the Fourth Canadian Conference on Computational Geometry*, 1992.

[11] Gordon Kindlmann, Ross Whitaker, Tolga Tasdizen and Torsten Moller, Curvature-based transfer functions for direct volume rendering: methods and applications, *Proceedings of the 14th IEEE Visualization* 2003, pp. 513-520.