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 present our algorithm that generates a manifold triangular mesh F from a set of unorganized points and compare it with various existing algorithms for Surface Reconstruction.

Contents

List of Figures

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.

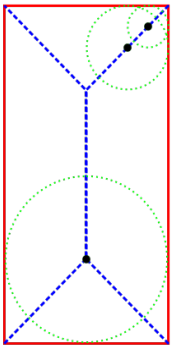
1.5 Euclidean minimum spanning tree

The Euclidean minimum spanning tree or EMST is a minimum spanning tree of a set of n points in the plane (or more generally in d), where the weight of the edge between each pair of points is the Euclidean distance between those two points.

1.6 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.

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



(a)

Figure 1: (a) MAT (Original Shape, Tangent Circles, MAT).

2 Literature Survey

An early, and influential, attempt to characterize the shape of a set of points is due to [**?**], 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 [**?**]. 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 [**?**], 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 [**?**].

Two rather different constructs, $r - shape$ and $s - shape$, were defined in [**?**] 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 [**?**]. 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 [**?**]. 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 [**?**] 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 Algorithm

We start with Delaunay tetrahedralization DT of available points in $\mathbb{R}^3$. We use the 3D Delaunay to obtain Euclidean Minimum Spanning Tree of DT using Kruskal's algorithm.

3.1 Definitions

3.1.1 Hole Graph

Graph induced by the edges of the surface whose edge degree is 1 and of EMST whose edge degree is less than or equal to 1.

3.1.2 Hole

A connected component in hole graph. A hole is a cycle.

3.1.3 Stitching Edges

A subset of edges of all the possible edges in a hole which satisfies the following conditions:

- Length of the edge is less than 2 times the minimum of the maximum length edge incident on the vertices of given edge
- The edge does not overlap with the existing edges and adjacent faces

3.1.4 Score

Score of a face is defined as the minimum angle of the face.

3.1.5 Possible Faces

Given an edge (u,v) the possible faces are each (u,v,w) for w $\in$ (Adjacent vertices of u in $Surface \cup$ Adjacent vertices of v in $Surface$)

3.1.6 PQ

PQ is a priority queue with key as score of a face and value as the edge and point pair representing the face. The priority queue is sorted in descending order.

3.1.7 Face Acceptance Condition

A face satisfying following conditions are accepted into the surface:

- Either all the edges of the face must have edge degrees less than 2 or each edge with edge degree 2 must be able to replace a face adjacent to it
- Number of edges in the face with edge length less than max length of EMST edge should be greater than or equal to edge condition integer

Algorithm 1: Surface Reconstruction using Euclidean MST

Function *reconstruct(P)* `// P: Point Set`
  $DT \leftarrow$ Delaunay triangulation of P
  $G \leftarrow$ graph of MST of DT
  $S \leftarrow$ surface initialized with ϕ
  $PQ \leftarrow maxHeap(score, edge, point)$
  `//` *score* `is the minimum angle of a` $\triangle$
  $edgeCondition \leftarrow 2$
  repeat
    $G' \leftarrow G$
    $stitchEdges \leftarrow getFacesFromEdges(G, DT)$
    $PQ \leftarrow PQ \cup getFacesFromEdges(G, sticthEdges)$
    $PQ \leftarrow addToSurface(S, G, PQ, edgeCondition)$
    $edgeCondition \leftarrow edgeCondition - 1$
  until $G' = G$;
  $PostProcess(S, G)$

Function *isEdgeOverlap(e,G)*
  return *true* if e has projection on any of the faces adjacent to vertices of e in G
  return $false$ otherwise

Function *getFacesFromEdges(G, edgeSet)*
  $faces \leftarrow \phi$
  foreach $e(u, v) \in edgeSet$ **do**
    $pointSet \leftarrow N_G[u] \cup N_G[u] \setminus \{u, v\}$
    foreach $w \in pointSet$ **do**
      $faces \leftarrow faces \cup \{scoreof\{u, v, w\}, e, w\}$
    end
  end
  return $faces$

Function *getStitchEdges(G, DT)*
  $H \leftarrow G[\{u, v \mid faceDegree(u, v) \leq 1\}]$
  $sticthEdges \leftarrow \phi$
  foreach $e(u, v) \in DT \setminus H$ *(sorted by length)* **do**
    if $u, v \in$ *same component in* $H \wedge$
     $e.length \leq 2 \times \min(maxLength(u), maxLength(v)) \wedge \neg edgeOverlap(e) \wedge$
     $(\neg u.covered \vee \neg v.covered)$ **then**
      $sticthEdges \leftarrow sticthEdges \cup \{e\}$)
      $u.covered \leftarrow True$
      $v.covered \leftarrow True$
    end
  end
  return sticthEdges

Function *addToSurface(S,G,PQ,edgeCondition)*
  $tempPQ \leftarrow \phi$
  while $PQ \neq \phi$ **do**
    $score, edge_0, point_0 \leftarrow PQ.pop()$
    $face \leftarrow \{edge, point\}$
    if $face \in S$ **then**
      continue
    end
    $edge_1, edge_2 \leftarrow face \setminus \{edge_0\}$
    $point_1, point_2 \leftarrow$ opposite points of $edge_1$ and $edge_2$ in $face$
    Check if it is possible to add face to $edge_i's$ FaceAdjList such that either $faceEdgeDegree \leq 1$ or if $face$ can remove any face f_b in FaceAdjList such that f_a is the other face in $FaceAdjList$ and $angle(face, f_a) < angle(f_a, f_b)$
    if *false* **then**
      continue
    end
    $removedFaces \leftarrow$ faces removed
    if $\exists f` \in (N_G[face] \setminus removedFaces) \mid f`$ *overlaps with* $face$ **then**
      continue
    end
    if *count(edges of face with length <= max Edge Length in MST)* $\leq edgeCondition$ **then**
      $tempPQ \leftarrow tempPQ \cup \{score, edge_0, point_0\}$
      continue
    end
    $S \leftarrow S \cup face$
    $G \leftarrow G \cup face$
    $PQ \leftarrow PQ \cup getFacesFromEdges(G, face)$
  end
  return $tempPQ$

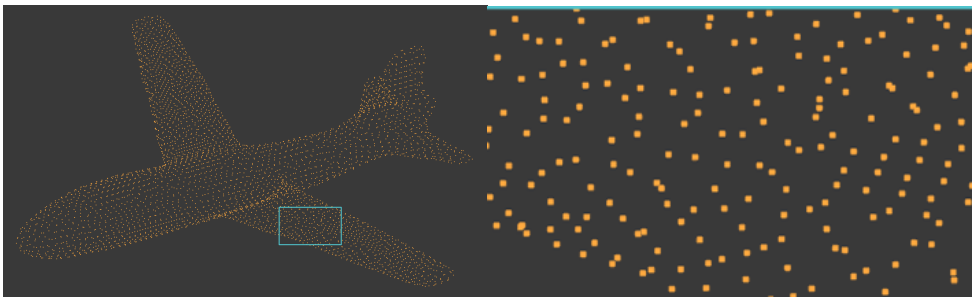
Function *PostProcess(G, S)*
  $H \leftarrow G[\{u, v \mid faceDegree(u, v) \leq 1\}]$
  foreach $cycle \in H$ **do**
    repeat
      $edge_0, edge_1 \leftarrow$ pair of adjacent Edges in $cycle$ with min angle
      $S \leftarrow S \cup face(edge_0, edge_1)$
      $G \leftarrow G \cup \{edge_0, edge_1\}$
      $cycle \leftarrow cycle \setminus \{edge_0 \cap edge_1\}$
    until $|cycle| \leq 3$;
    $S \leftarrow S \cup cycle$
    $G \leftarrow G \cup cycle$
  end

4 Comparison

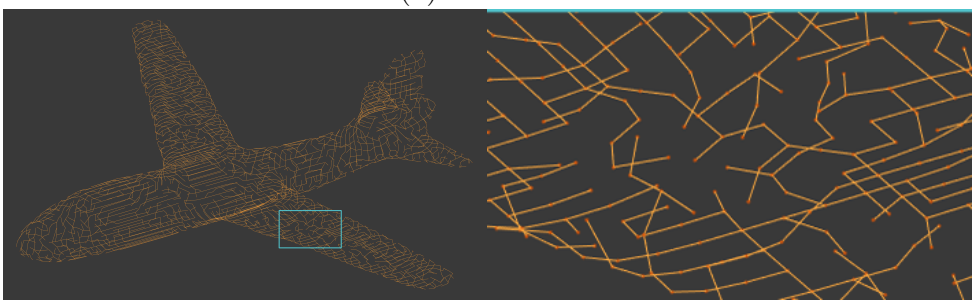
Model	# of Points	Our Algorithm	Screened Poisson	PowerCrust	Robust Cocone
Anchor	263286	43.412s	15.536s	2m 24.156s	3m 23.892s
Daratech	245836	49.316s	29.023s	2m 3.79s	3m 44.312s
Dancing Children	468022	1m 44.648s	47.642s	3m 36.024s	8m 16.432s
Gargoyle	481358	1m 45.236s	1m 6.947s	4m 46.892s	8m 16.808s
Lord Quasimoto	350000	1m 11.784s	29.944s	3m 21.192s	5m 51.044

Table 1: Running Time of Different Algorithms

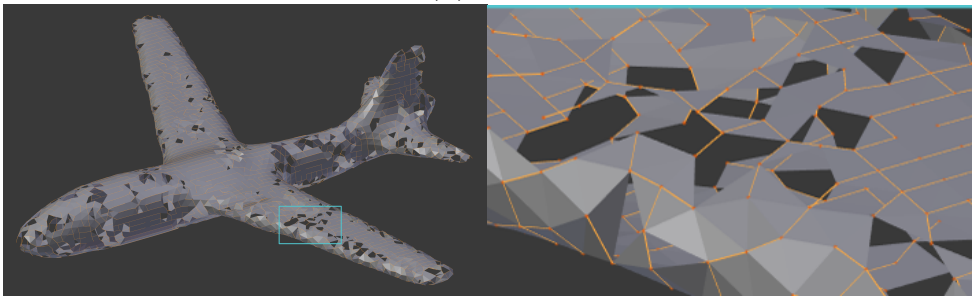
5 Illustrations



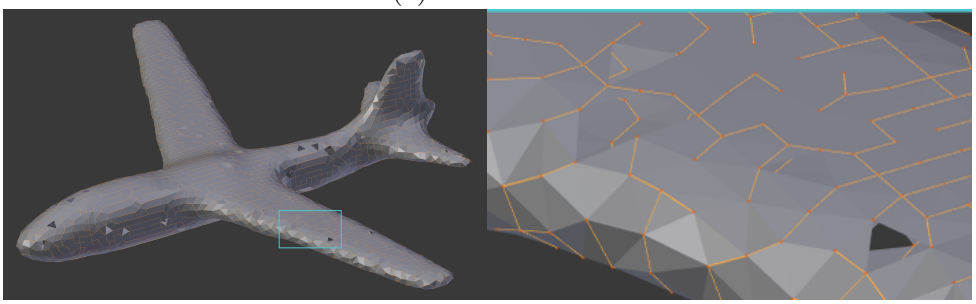
(a) Point Set



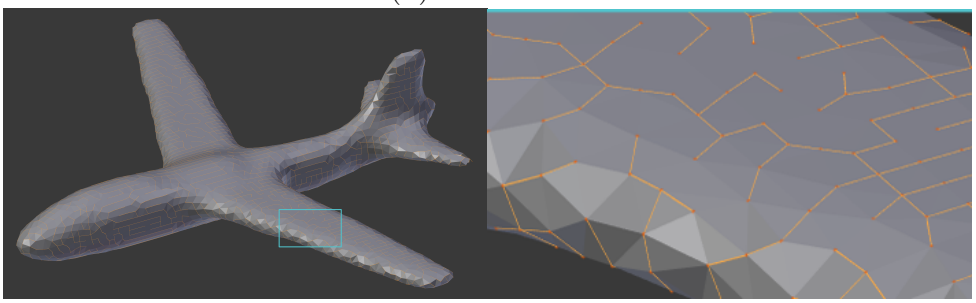
(b) MST



(c) Iteration 1



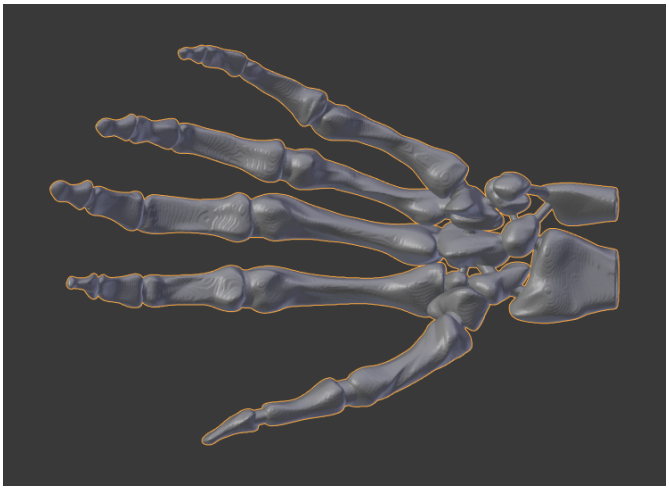
(d) Iteration 3



(e) After Post Processing

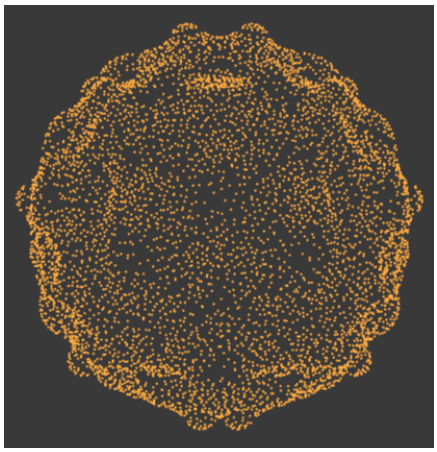
(a) (b)

(c) (d)

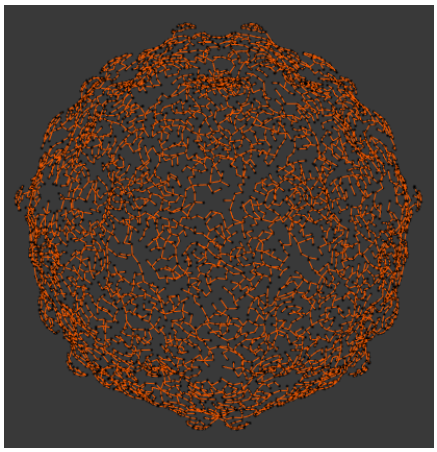


(e)

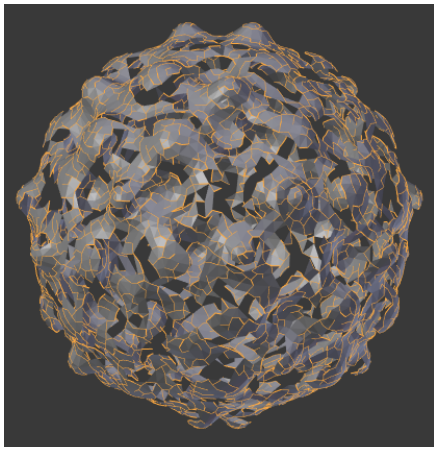
Figure 2: (a),(b) Reconstructed Model with 757490 vertices and (c),(d) with 172974 vertices. (e) Rendered Image of a reconstructed surface with 327323 vertices.



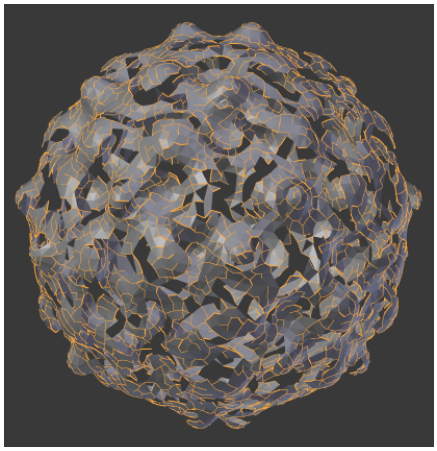
(a)



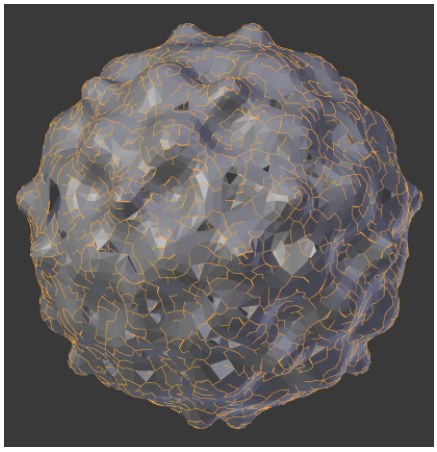
(b)



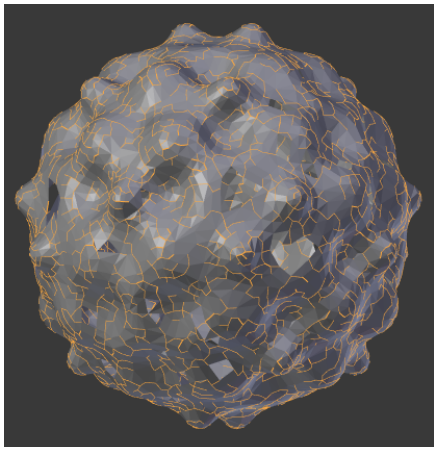
(c)



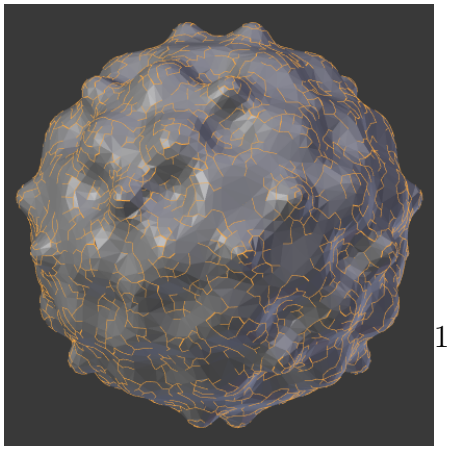
(d)



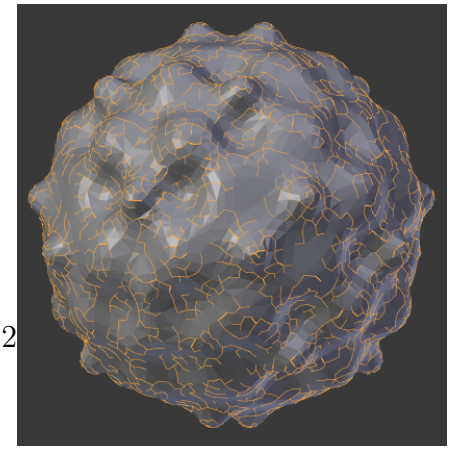
(e)



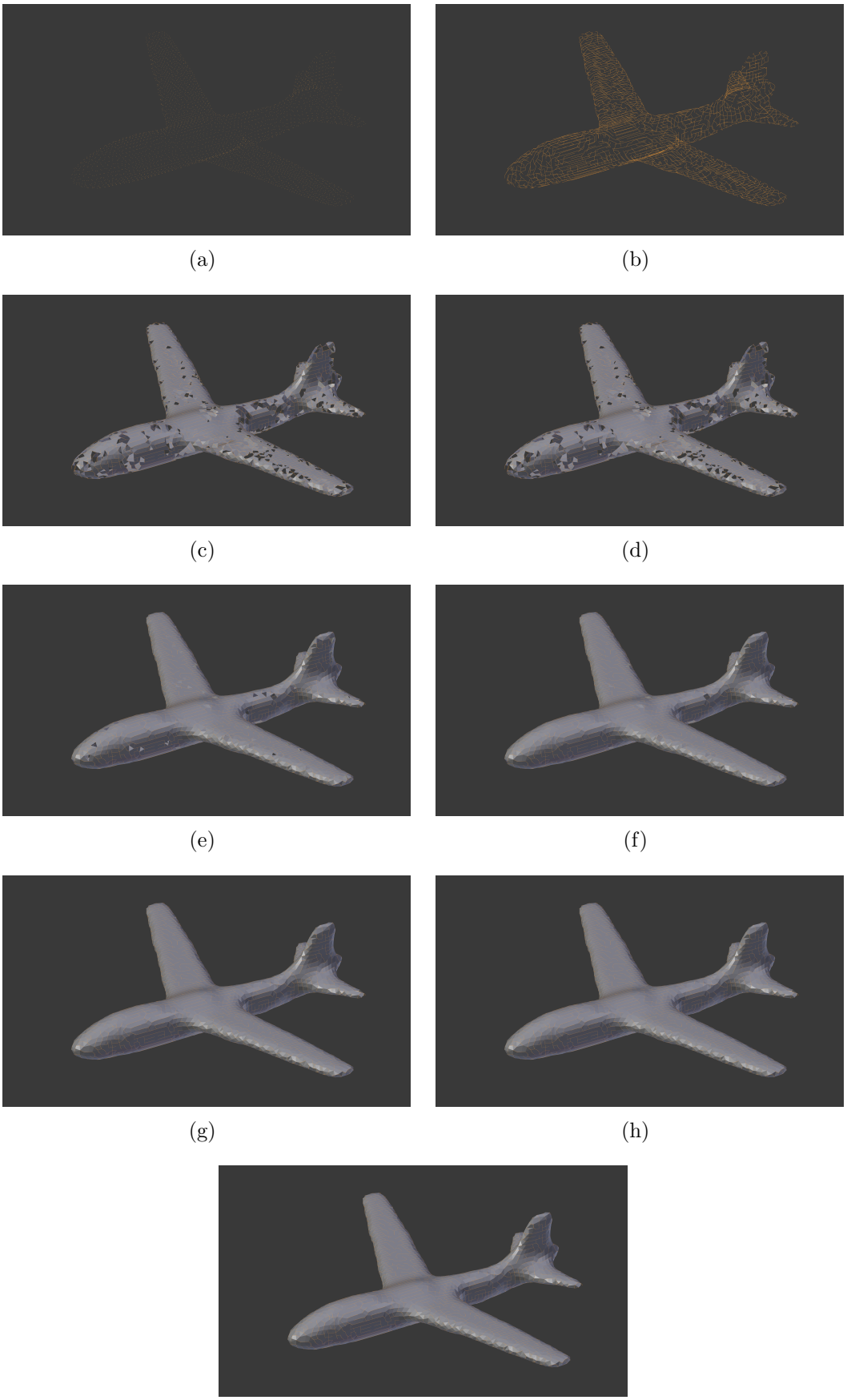
(f)



(g)



(h)



(a) (b) (c) (d) (e) (f) (g) (h) (i)

Figure 4: Surface during Iteration of Plane.