Polygonal Surface Remeshing with Delaunay Refinement

T. K. Dey[1], G. Li[1], and T. Ray[1]

The Ohio State University, Columbus, Ohio, USA
{tamaldey,ligan,rayt}@cse.ohio-state.edu

Summary. Polygonal meshes are used to model smooth surfaces in many applications. Often these meshes need to be remeshed for improving the quality, density or gradedness. We apply the Delaunay refinement paradigm to design a provable algorithm for isotropic remeshing of a polygonal mesh that approximates a smooth surface. The proofs provide new insights and our experimental results corroborate the theory.

Key words: surface meshing, computational geometry, computational topology, Delaunay refinement.

1 Introduction

Polygonal meshes including the triangular ones are often used in many applications of science and engineering to model a smooth surface. Such a mesh is usually designed by some modeling software (CAD software), or is generated by some reconstruction algorithm from a set of sample points provided by a scanning device. These meshes often lack properties that are useful for subsequent processing. For example, the triangle shapes may be poor for subsequent numerical methods [PB01], or the application may need a graded mesh with different levels of density without sacrificing the shape quality. To address these requirements, the input mesh needs to be remeshed, that is, they need to be sampled and triangulated appropriately.

Because of its application needs, the problem of remeshing has been a topic of research in many areas. We refer the readers to [Alliez03, Sifri] and the references therein. In this work, we apply the Delaunay refinement technique to design a provable algorithm for remeshing polygonal surfaces. The Delaunay refinement, originally pioneered by Chew [Chew89] is a powerful paradigm for meshing. First, it produces a Delaunay mesh as output which is often favored over other meshes due to its isotropic nature. Second, the paradigm offers a very simple mechanism to guarantee the quality of the mesh. It works on the following “furthest-point” principle. Whenever some desired criterion of the mesh is not satisfied, the algorithm inserts a point within the domain which is locally furthest from all other existing points. Then, the desired condition is automatically satisfied when the algorithm terminates. The main challenge entails to guarantee the termination. The Delaunay refinement paradigm has been successfully used for meshing two and three dimensional domains [CDRR204, PW04, Rup95, Shew98].

Researchers have also explored the Delaunay refinement technique for surface meshing. Chew [Chew93] proposed the first surface meshing algorithm with this technique though without any guarantee. Cheng et al. [CDES01] combined the sampling theory of Amenta and Bern [AB98] for surface reconstruction with the Delaunay refinement for producing a mesh for skin surfaces. Boissonnat and Oudat [BO03] gave an elegant algorithm for general surfaces assuming that the feature sizes of the surface can be computed. Cheng, Dey, Ramos and Ray [CDRR04] gave a different algorithm for surface meshing which replaced the feature size computations with critical point computations. All these algorithms are meant for smooth surfaces and not for polygonal surfaces. Although the authors of [BO03] and [CDRR04] indicate that their algorithms work well for polygonal surfaces in practice, no guarantee is proved.

Of course, treating a polygonal mesh as an input polyhedron one can use any of the polyhedra meshing algorithms [CDRR204, PW04] to obtain a meshing of the polygonal surfaces. Unfortunately, the output mesh produced with this approach may have some undesirable properties. These algorithms respect the edges and vertices of the input polygonal mesh so that the underlying space of the output is exactly the same as that of the input. As a result small input angles are not eliminated. In particular, if applied to a triangular mesh, the algorithm tries to compensate for acute angles invariably present in each triangle and produces too many sample points. What we are looking for is a remeshing of the input polygonal mesh where the new points are constrained to be on the input mesh though the underlying space of the output is allowed to differ from the input.

Results. Given an input polygonal mesh G, our algorithm samples G with a Delaunay refinement approach and produces an output mesh that has the same topology and approximate geometry of G. Moreover, the output triangles have bounded aspect ratios. These guarantees are proved assuming that G satisfies certain conditions. Specifically, we show that if G approximates a smooth surface both point-wise and normal-wise closely, then the Delaunay refinement running with the desired conditions terminates. It is only the proofs that use a hypothetical smooth surface Σ approximated by G, but Σ plays no role in the algorithm. In practice, there are many situations where such assumption is valid. For example, G might be a polygonal surface reconstructed from a dense sample of a smooth surface [Dey03], or a designed surface approximating a smooth surface closely.

Overview. Our algorithm has two distinct phases. In the first phase it recovers the topology of G. In the second phase, it refines further to recover geometry and ensures quality of the triangles. For topology recovery we follow the approach of Cheng et al. [CDRR04] to build the mesh as the restricted Delaunay triangulation of a set of points sampled from the input polygonal surface. This restricted Delaunay triangulation has the same topology of the input surface if a property called topological ball holds [ES97]. We prove that one can find a point on the input surface that is far-away from all existing sampled points if this property does not hold. Such a point is sampled to drive the Delaunay refinement. For geometry recovery we present similar results that lead to a new refinement algorithm. Section 2 and 3 describe the algorithms for topology and geometry recovery respectively. Section 4 and 5 build all necessary lemmas for the termination proof in section 6. Finally, in section 7 we show that meshes computed from dense point cloud data satisfy the conditions required for our proofs.

2 Delaunay refinement for topology

The Delaunay refinement algorithm first concentrates on getting the topology right. Before we describe the algorithm we briefly set up our notations for Delaunay and Voronoi diagrams. The Delaunay triangulation of a point set $P \subset \mathbb{R}^3$ is denoted as $\mathrm{Del}\, P$ and its dual Voronoi diagram as $\mathrm{Vor}\, P$. A Voronoi cell for a point $p \in P$ is denoted as V_p. Skipping the details we just mention that $\mathrm{Del}\, P$ is a simplicial complex where a k-simplex is dual to a Voronoi face of dimension $3-k$ which is the intersection of $3-k$ Voronoi cells. Zero-,one- and two- dimensional Voronoi faces are called Voronoi vertices, Voronoi edges and Voronoi facets respectively.

The topology is recovered in two phases. First, in the manifold recovery phase, a Delaunay mesh is computed which is guaranteed to be a 2-manifold. Then, further refinement is carried out so that the 2-manifold has the same topology as that of the input. In both phases we grow a sample Q, generated from G iteratively. The notion of restricted Delaunay triangulation plays an important role in the algorithms and the proofs.

Definition 1. *Let $v(\sigma)$ denote the set of vertices of a simplex σ and $V_q|_G = V_q \cap G$. The Delaunay complex*

$$\mathrm{Del}\, Q|_G = \{\sigma \in \mathrm{Del}\, Q \mid \bigcap_{q \in v(\sigma)} V_q|_G \neq \emptyset\}$$

is called the restricted Delaunay triangulation of Q with respect to G.

Basically, $\mathrm{Del}\, Q|_G$ contains a dual Delaunay simplex for every Voronoi face intersected by G. The triangulation $\mathrm{Del}\, Q|_G$ plays a key role in the topology recovery phase. However, we need only a subcomplex of $\mathrm{Del}\, Q|_G$ to guarantee the manifold property. Let T be the set of triangles in $\mathrm{Del}\, Q$ dual to the Voronoi edges that intersect G. The *edge restricted Delaunay triangulation* $\mathrm{EDel}\, Q|_G$ is the simplicial complex made by T and its edges and vertices. The manifold recovery phase MflRecov computes $\mathrm{EDel}\, Q|_G$ while the topology recovery phase TopoRecov computes $\mathrm{Del}\, Q|_G$.

The refinement routines check if the Voronoi diagram $\mathrm{Vor}\, Q$ satisfies certain conditions. If not, they insert a point from G into Q which is far away from its nearest neighbor and hence from all points in Q. Specifically, MfldRecov enforces a manifold property explicitly and TopoRecov enforces a topological ball property. The routines are similar in spirit with those used for smooth surface meshing by Cheng et al. [CDRR04]. However, they differ in crucial details. We take the liberty of computing the furthest point in $G \cap V_q$ from q since G is polygonal and these computations do not require any optimization steps as opposed to the smooth surface meshing case. Also, the critical point computations in various phases of the algorithm of Cheng et al. are completely eliminated which results into new subroutines like Vcell with its new justification.

2.1 Manifold Recovery

The manifold recovery phase explicitly enforces the manifold condition. Two subroutines, VEDGE and DISK are called by the manifold recovery algorithm. VEDGE checks if any Voronoi edge, say in a Voronoi cell V_q, intersects G in more than one point. If so, it outputs the intersection point, say p, furthest from q. Clearly, p cannot be closer to any point in Q than q. Let T be the set of triangles in $\mathrm{EDel}\, Q|_G$, i.e., the triangles dual to the Voronoi edges in $\mathrm{Vor}\, Q$ intersected by G. Denote the set of triangles incident to q in T as Δt_q. DISK checks if Δt_q is a topological disk. If not, it returns the point in $G \cap V_q$ furthest from q. MFLDRECOV inserts the points returned by VEDGE and DISK into Q and updates $\mathrm{Vor}\, Q$. It starts with a vertex from each component in G and finally returns T when no more points need to be inserted.

Observation 2.1 *When* MFLDRECOV *terminates, T is a 2-manifold as Δt_q for each vertex q is a 2-disk.*

VEDGE($e \in V_q$)

If e intersects G tangentially or at least in two points, return the point furthest from q among them, otherwise return `null`.

DISK(q)

If Δt_q is not a topological disk, return the furthest point in $G \cap V_q$.

MFLDRECOV(G)

1. Initialize Q with a vertex from each component of G.
2. Compute $\mathrm{Vor}\, Q$.
3. If there is a Voronoi edge $e \in \mathrm{Vor}\, Q$ so that VEDGE(e) returns a point p, insert p into Q and go back to step 2.
4. If there is a point $q \in Q$ so that DISK(q) returns a point p, insert p into Q and go back to step 2.
5. Output the set T of triangles dual to the Voronoi edges intersecting G.

2.2 Topology recovery

In the topology recovery phase we insert points into Q till $\mathrm{Vor}\, Q$ satisfies the following property. We say $\mathrm{Vor}\, Q$ satisfies the *topological ball property* if each Voronoi face of dimension k, $0 < k \leq 3$, intersects Σ in a topological $(k-1)$-ball if it intersects G at all. For $k = 0$, this intersection should be empty. The motivation for ensuring the topological ball property comes from the following result of Edelsbrunner and Shah [ES97].

Theorem 1. *The underlying space of the complex* $\mathrm{Del}\, Q|_G$ *is homeomorphic to* G *if* $\mathrm{Vor}\, Q$ *has the topological ball property.*

We use this theorem to guarantee the topology of the output. The VEDGE subroutine can be used to enforce the topological ball property for the Voronoi edges. A Voronoi facet $F \in V_q$ can violate this property by intersecting G in

(i) more than one topological interval, and/or
(ii) in one or more cycle.

If (i) happens, either a Voronoi edge of F intersects G more than once, or the dual Delaunay edge $\mathrm{dual}(F)$ has more than two triangles incident to it in $\mathrm{EDel}_Q|_G$. The topological disk condition for Δt_q will be violated where q is any of the end vertices of $\mathrm{dual}(F)$. So, this violation can be handled by the subroutine DISK. For (ii) we introduce a subroutine FCYCLE to check the condition and to identify the point in $F \cap G$ furthest from q. This furthest point will be a point of intersection between F and an edge of G.

FCYCLE($F \in V_q$)

If $F \cap G$ has a cycle, return the point in $F \cap G$ furthest from q.

So far we have subroutines for checking the topological ball property for Voronoi edges and facets. The Voronoi cell also needs a separate check. For a Voronoi cell V_q, the subroutine VCELL checks if $W = V_q \cap G$ is a topological disk. By the time VCELL is called in the algorithm, W is ensured to be a 2-manifold with a single boundary. Also, W has a single component as Q is initialized with a vertex from each component of G. It is easy to check if such a surface is a disk by computing the Euler characteristic that is given by the alternating sums of the number of vertices, edges and polygons in W. Of course, this requires one to compute the boundary edges and vertices of W. However, we can simply ignore those vertices and edges as they cancel out in the alternating sum.

VCELL(q)

Determine the number $\#v$ of vertices, $\#e$ of edges and $\#g$ of polygons in G intersecting V_q. If $\#v - \#e + \#g \neq 1$, return the point in $G \cap V_q$ furthest from q.

Now we have all ingredients to recover the topology of G into the new mesh T.

TOPORECOV(G, Q)

1. If $Q = \emptyset$ initialize Q with a vertex from each component of G.
2. Compute $\mathrm{Vor}\, Q$.
3. If any of VEDGE, DISK, FCYCLE and VCELL, necessarily in this order, succeeds and returns a point p, insert p into Q and go back to step 2.
5. Output the set T of triangles dual to the Voronoi edges intersecting G.

3 Delaunay refinement for geometry

It is often not enough to recover only the topology of G. The remeshed surface T should also follow the geometry of G. For this, we need to sample G more densely

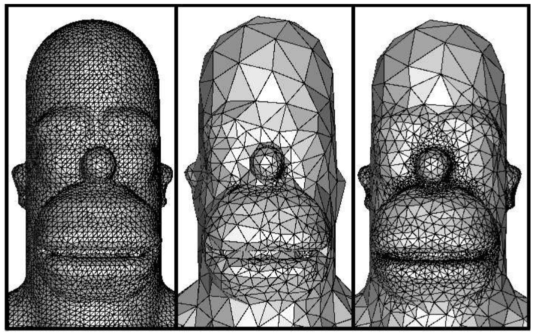

Fig. 1. The input mesh is uniformly dense everywhere. The output is a graded mesh at two different levels of density.

than the topology recovery requires. The level of density is controlled by an user parameter λ. Figure 1 shows an example of remeshing at two different levels of density.

The algorithm for geometry recovery uses the structure of the Voronoi cells to determine if the mesh should be refined locally. For a point $q \in Q$, the set $W = V_q \cap G$ is a 2-disk after the topology recovery phase. This disk separates V_q into two subsets one on each side of W. Let V_q^+ and V_q^- denote these subsets and let q^+ and q^- be the Voronoi vertices in V_q^+ and V_q^- respectively furthest from q. If any of V_q^+ and V_q^- is unbounded, the corresponding furthest vertex is taken at infinity. The points q^+ and q^- are like poles of the Voronoi cell V_q as defined by Amenta, Bern [AB98] for smooth surfaces. When points are sampled from a smooth surface densely, it is known that the poles of a point remain far away from it. Although we deal with a non-smooth surface G, we can claim (Lemma 15) a similar property if G approximates a smooth surface closely enough.

Geometry recovery checks if all triangles Δt_q incident to a vertex q are small enough compared to the pole distance $h_q = \min\{\|q - q^+\|, \|q - q^-\|\}$. Let $r(t)$ denote the circumradius of a triangle in Δt_q. For an user parameter λ, if $r(t)/h_q$ is larger than 12λ, we insert the point c where the dual Voronoi edge of t, dual(t), intersects G. Clearly, the value of λ denotes the level of refinement.

A similar procedure can be used to guarantee the quality of the triangles. Let $\rho(t)$ denote the ratio $r(t)/\ell(t)$ where $\ell(t)$ is the length of the shortest edge of t. It is known that triangles with bounded ρ value have bounded aspect ratios. In the algorithm, we check if a triangle t has $\rho(t)$ more than $(1 + 8\lambda)$, and, if so, we insert the point dual$(t) \cap G$ into Q.

The particular choice of 12λ and $(1 + 8\lambda)$ comes from our proofs.

GeomRecov(G, λ)

1. Q:=$\emptyset$
2. T:= TopoRecov(G, Q), Q := vertices of T
3. For each $q \in Q$, if there is a $t \in \Delta t_q$ with $c = \mathrm{dual}(t) \cap G$ so that either
 (i) $\rho(t) > (1 + 8\lambda)$, or
 (ii) $r(t)/h_q > 12\lambda$
 then insert c into Q and go back to step 2.
4. Output T.

Notice that, although GeomRecov requires an user supplied λ, MfldRecov and TopoRecov require no such λ. We choose $\lambda = 0.02$ in GeomRecov for our experiments. Figure 5 shows some results of this experiment.

4 Prelude to proofs

The refinement routines may not terminate for arbitrary polygonal meshes but we prove that when the input mesh G approximates a smooth surface closely enough, they necessarily terminate. We need several definitions to state this approximation precisely.

Let Σ be a smooth, compact surface without boundary which is approximated by G. In general, G and hence Σ may have more than one connected component. The output is a triangulation T whose vertex set lies in G. Abusing the notations slightly we will use G and T to mean their underlying complexes and spaces as well.

4.1 Definitions

Distances: For a point $x \in \mathbb{R}^3$ and a set $X \subseteq \mathbb{R}^3$, let $d(x, X)$ denote the Euclidean distance of x from X, i.e.,

$$d(x, X) = \inf_{y \in X} \|x - y\|.$$

A ball $B_{c,r}$ is the set of points whose distance to c is no more than r.

Medial axis and feature size: The *medial axis* M of Σ is the closure of the set $X \subset \mathbb{R}^3$ so that, for each point $x \in X$, $d(x, \Sigma)$ is realized by two or more points. Alternatively, M is the loci of the centers of the maximal balls whose interiors are empty of points from Σ. These balls, called *medial balls*, are tangent to Σ at one or more points. At each point $x \in \Sigma$, there are two such tangent medial balls. Define a function $f : \Sigma \to \mathbb{R}$ where $f(x) = d(x, M)$. The value $f(x)$ is called the *local feature size* of Σ at x [ABE98].

Projection map: Often we will use a projection map $\nu : \mathbb{R}^3 \setminus M \to \Sigma$ where $\tilde{x} = \nu(x)$ is the closest point in Σ, i.e., $d(x, \Sigma) = \|x - \tilde{x}\|$.

Oriented normals: The normals of the three spaces, Σ, G and T play an important role in the proofs. These normals need to be oriented. We denote the normal of Σ at a point x as $\tilde{\mathbf{n}}_x$. These normals are oriented, i.e., the normal $\tilde{\mathbf{n}}_x$ points to the bounded component of $\mathbb{R}^3 \setminus \Sigma'$ where x is in the connected component $\Sigma' \subseteq \Sigma$. Similarly, we define an oriented normal $\mathbf{n}_g$ for each polygon g in G. Let g be in the connected component $G' \subseteq G$. The normal $\mathbf{n}_g$ points to the bounded component of $\mathbb{R}^3 \setminus G'$. We will also orient the normals of the Delaunay triangles computed by our algorithms. For a Delaunay triangle t, we orient its normal $\mathbf{n}_t$ so that $\angle \mathbf{n}_t, \tilde{\mathbf{n}}_{\tilde{p}} < \pi/2$ where p is a vertex subtending the largest angle in t.

Angle notation: The angle between two oriented normals $\mathbf{a}$ and $\mathbf{b}$ is denoted as $\angle \mathbf{a}, \mathbf{b}$.

Thickening and approximation: We require that G approximate Σ pointwise, i.e., it resides within a small thickening of Σ. In particular, for $0 \le \delta < 1$, we introduce a thickened space

$$\delta\Sigma = \{x \in \mathbb{R}^3 | d(x, \Sigma) \le \delta f(\tilde{x})\}.$$

We will also require that G approximates Σ normal-wise. To specify this pointwise and normal-wise approximation we define:

Definition 2. *G is (δ, μ)-flat with respect to Σ if the two conditions hold:*

(i) For $\delta < 1$, the closest point $\tilde{x} \in \Sigma$ of each point $x \in G$ is within $\delta f(\tilde{x})$ distance, and conversely, each point $x \in \Sigma$ has a point of G within $\delta f(x)$ distance.
(ii) For any point x in a polygon $g \in G$, the angle $\angle \mathbf{n}_g, \mathbf{n}_{\tilde{x}}$ is at most μ for some $\mu < 1$.

4.2 Consequences

We assume G to be (δ, μ)-flat with respect to Σ. The following lemmas are direct consequences of this assumption.

Lemma 1. *(i) $G \subset \delta\Sigma$. (ii) Let $\mathbf{n}_g$ and $\mathbf{n}'_g$ be the normals of two adjacent polygons $g, g' \in G$. We have $\angle \mathbf{n}_g, \mathbf{n}'_g \le 2\mu$.*

Lemma 2. *Let p and q be two points in $\delta\Sigma$ with $\|p - q\| \le \lambda f(\tilde{p})$. Then, $\|\tilde{p} - \tilde{q}\| \le 2(\lambda + \delta) f(\tilde{p})$.*

Proof. We have

$$\begin{aligned} \|\tilde{p} - \tilde{q}\| &\le \|\tilde{p} - p\| + \|p - q\| + \|q - \tilde{q}\| \\ &\le 2(\|\tilde{p} - p\| + \|p - q\|) \le 2(\lambda + \delta) f(\tilde{p}). \end{aligned}$$

The medial balls of Σ are empty of points from Σ but not from G. This is a reason why the proofs for Σ as detailed in [CDRR04] do not extend to G. First, we clear this obstacle by showing that the medial balls after appropriate shrinking can be made empty of points from $\delta\Sigma$ and hence G. This claim is formalized in the following lemma. Let L_x^+ and L_x^- denote the rays originating at $x \in \Sigma$ in the directions of $\tilde{\mathbf{n}}_x$ and $-\tilde{\mathbf{n}}_x$ respectively, see Figure 2 (proof in the extended version [Dey05]).

Lemma 3. *Let $0 < \delta < 1/4$. For each point $x \in \Sigma$ there are two balls $B = B_{c,r}$ and $B' = B_{c',r'}$ with*

(i) $c \in L_x^+$ and $c' \in L_x^-$,
(ii) $r = r' = (1 - 4\delta) f(x)$,
(iii) $d(x, B) = d(x, B') = 4\delta f(x)$,
(iv) $\delta\Sigma \cap B = \emptyset$ and $\delta\Sigma \cap B' = \emptyset$.

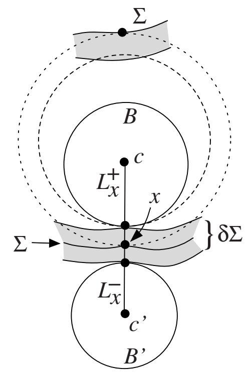

Fig. 2. A medial ball tangent to Σ at x (shown with dotted boundary) is shrunk radially first to the ball with dashed boundary. Then it is shrunk further to be empty of $\delta\Sigma$ to the ball with solid boundary.

5 Normals and conditions

The normals to the triangles and edges that the Delaunay refinement produces play an important role in the analysis. For convenience we define the following two functions for any $\lambda > 0$.

$$\alpha(\lambda) = \frac{\lambda}{1-4\lambda} \text{ and}$$
$$\beta(\lambda) = \arcsin\lambda + \arcsin\left(\frac{2}{\sqrt{3}}\sin(2\arcsin\lambda)\right).$$

For smooth surfaces it is known that the triangles with small circumradius lie almost parallel to the surface. This follows from the following lemma proved by Amenta, Choi, Dey and Leekha [ACDL00] and the fact that medial balls incident to a point on the surface are relatively large.

Lemma 4. *Let $B = B_{c,r}$ and $B' = B_{c',r'}$ be two balls meeting at a single point p of a smooth surface Σ. Let $t = pqr$ be a triangle where*

(i) p subtends the largest angle of t,
(ii) the vertices of t lie outside of $B \cup B'$, and
(iii) the circumradius of t is no more than $\lambda \min\{r, r'\}$ where $\lambda < \frac{1}{\sqrt{2}}$.[1]

Then the acute angle between the line of the normal $\mathbf{n}_t$ of t and the line joining c, c' is no more than $\beta(\lambda)$.

[1] In [ACDL00], the constant is stated smaller, but $\frac{1}{\sqrt{2}}$ is also a valid choice.

This lemma implies that the small Delaunay triangles lie almost parallel to the surface, a key fact used in the Delaunay refinement algorithm of Cheng et al. [CDRR04]. Lemma 7 is a version of this fact for the non-smooth surface G which we prove with our assumption that G follows Σ point-wise.

Another key ingredient used in smooth case is that the normals of a smooth surface cannot vary too abruptly. Precisely, Amenta and Bern [AB98] proved the following lemma.

Lemma 5. *Let x and y be any two points in Σ with $\|x - y\| \leq \lambda f(x)$ and $\lambda < 1/4$. Then, $\angle \tilde{\mathbf{n}}_x, \tilde{\mathbf{n}}_y \leq \alpha(\lambda)$.*

Lemma 1 is the non-smooth version of the above lemma which holds because G follows Σ normal-wise.

5.1 Triangle and edge normals

Now we focus on deriving the flatness property of the triangles and edges that are produced by the Delaunay refinement of G. The Delaunay refinement of G generates a point set Q on G. Thus, necessarily $Q \subset \delta\Sigma$ as $G \subset \delta\Sigma$. It is easy to see that the triangles and edges with such points as vertices may have normals in any direction no matter how small they are. A key observation we make and formalize is that when vertices have a suitable sparsity condition, i.e., a lower bound on their mutual distances, the triangles lie almost parallel to the surface Σ. This is proved in Lemma 7 with the help of large empty balls guaranteed by Lemma 3 and the technical Lemma 6 (proof in [Dey05]).

Lemma 6. *Let $\ell > \sqrt{\delta} > 0$ and $\delta \leq 1/4$. Let $B = B_{a,R}$ and $B' = B_{b,r}$ be two balls whose boundaries ∂B and $\partial B'$ intersect in a circle C with the following conditions.*

(i) Let p be the point where the line joining a and b intersects $\partial B'$ outside B. The distance of p from any point on C is at least ℓR.
(ii) The distance of p from ∂B is no more than δR.

Then,

$$r \geq \frac{R}{9}.$$

Lemma 7. *For $0 \leq \delta < \lambda < 1/48$ and $\ell > \sqrt{6\delta}$, let t be a triangle and q be any of its vertices where*

(i) vertices of t lie in $\delta\Sigma$,
(ii) q is at least $\ell f(\tilde{q})$ distance away from all other vertices of t,
(iii) the circumradius of t is at most $\lambda f(\tilde{q})$.

Then, $\angle \mathbf{n}_t, \tilde{\mathbf{n}}_{\tilde{q}} \leq \beta(12\lambda) + \alpha(6\lambda)$.

Proof. Let p be the vertex of t subtending the largest angle. First, we prove that there are two balls of radius at least $\frac{f(\tilde{p})}{12}$ being tangent at p and with centers on $L^+_{\tilde{p}}$ and $L^-_{\tilde{p}}$ respectively.

If $\delta = 0$, the vertices of t lie on Σ and the two medial balls at p satisfy the condition. So, assume $\delta \neq 0$. Consider a ball $B = B_{c,(1-4\delta)f(\tilde{p})}$ as stated in Lemma 3 for the point $\tilde{p}$. This ball is empty of any point from $\delta\Sigma$ and therefore does not contain any vertex of t. Consider a ball D with the center p and radius $\ell f(\tilde{p})$ where $\ell > \sqrt{6\delta}$. This ball also does not contain any vertex of t by the condition (ii). Let C be the circle of intersection of the boundaries of B and D. Let $B' = B_{w,r}$ be the ball whose boundary passes through C and p (Figure 3). No vertex of t lies inside B' as $B' \subset B \cup D$ and both B and D are empty of the vertices of t. We claim that $r \geq f(\tilde{p})/12$.

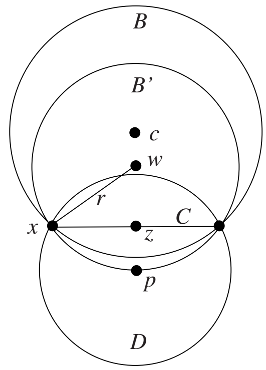

Fig. 3. Illustration for Lemma 7.

Let x be any point on the circle C whose center is z. The radius R of B is equal to $(1-4\delta)f(\tilde{p})$. So, $\|x-p\| = \frac{\ell}{1-4\delta}R$. The distance $d(p,B) \leq \|p-\tilde{p}\| + d(\tilde{p},B)$. Since p lies in $\delta\Sigma$, $\|p-\tilde{p}\| \leq \delta f(\tilde{p})$, and $d(\tilde{p},B) \leq 4\delta f(\tilde{p})$ by Lemma 3(iii). So,

$$d(p,B) \leq 5\delta f(\tilde{p}) \leq \frac{5\delta}{1-4\delta}R \leq 6\delta R \text{ for } \delta < \tfrac{1}{24}.$$

Since $\ell > \sqrt{6\delta}$ we can apply Lemma 6 to B and B' to get

$$r \geq \frac{R}{9} = \frac{f(\tilde{p})(1-4\delta)}{9} \geq \frac{f(\tilde{p})}{12}$$

using $(1-4\delta) > \frac{3}{4}$ for $\delta \leq \frac{1}{16}$.

Applying the above argument to the other ball $B_{c',(1-4\delta)f(\tilde{p})}$ as guaranteed by Lemma 3, we get another empty ball B'' with radius at least $\frac{f(\tilde{p})}{12}$ touching p. The centers of both B' and B'' lie on the line of the normal $\tilde{\mathbf{n}}_{\tilde{p}}$. Notice that both B' and B'' meet at a single point p. Shrink both B' and B'' keeping them tangent at p till their radius is equal to $f(\tilde{p})/12$. Now the balls B', B'' and the triangle t satisfy the conditions of Lemma 4. First, the vertices of t lie outside B' and B''. Secondly, the circumradius of t is at most $\lambda f(\tilde{p})$, i.e., 12λ times the radii of B' and B''. Therefore, the acute angle between the lines of $\mathbf{n}_t$ and $\tilde{\mathbf{n}}_{\tilde{p}}$ is at most $\beta(12\lambda)$. For $\lambda < 1/48$, $\beta(12\lambda) < \pi/2$. This implies that the upper bound of $\beta(12\lambda)$ also holds for the oriented normals, i.e.,

$$\angle \mathbf{n}_t, \tilde{\mathbf{n}}_{\tilde{p}} \leq \beta(12\lambda).$$

Now, consider any vertex q of t. Since the circumradius of t is no more than $\lambda f(\tilde{q})$, $\|p - q\| \leq 2\lambda f(\tilde{q})$. By Lemma 2, $\|\tilde{p} - \tilde{q}\| \leq 2(2\lambda + \delta) f(\tilde{q}) \leq 6\lambda f(\tilde{q})$ for $\delta < \lambda$. Then, by Lemma 5 $\angle \tilde{\mathbf{n}}_{\tilde{p}}, \tilde{\mathbf{n}}_{\tilde{q}} \leq \alpha(6\lambda)$ provided $6\lambda < 1/4$, or $\lambda < 1/24$. Therefore,

$$\angle \mathbf{n}_t, \tilde{\mathbf{n}}_{\tilde{p}} \leq \beta(12\lambda) + \alpha(6\lambda).$$

Similar to triangles, small edges with vertices on G also lie almost parallel to Σ (proof in [Dey05]).

Lemma 8. *For $0 \leq \delta < \lambda < 1/48$ and $\ell > \sqrt{6\delta}$, let pq be an edge where*

(i) p and q lie in $\delta\Sigma$,
(ii) $\ell f(\tilde{q}) < \|p - q\| < \lambda f(\tilde{q})$.

Then, $\angle \mathbf{q}p, \tilde{\mathbf{n}}_{\tilde{q}} \geq \frac{\pi}{2} - \arcsin 6\lambda$.

5.2 Conditions

We use Lemma 7 and Lemma 8 to prove the correctness of the remeshing algorithms. These results depend on certain conditions, i.e., the values of λ and δ have to satisfy some constraints. They would in turn suggest some condition on the sparsity of the sampling by the Delaunay refinement.

Condition on sparsity

Lemma 7 and Lemma 8 will be applied to the Delaunay triangles and edges for a sample Q that the Delaunay refinement generates. It will be required that Q maintain a lower bound on the distances between its points.

Definition 3. *A point set Q is λ-sparse if each point $q \in Q$ is at least $\frac{\lambda}{(1+8\lambda)} f(\tilde{q})$ distance away from every other point in Q.*

The particular choice of the factor $\frac{\lambda}{1+8\lambda}$ will be clear when we argue about termination. One condition of lemmas 7 and 8 says that the length of an edge pq has to be more than $\sqrt{6\delta} f(\tilde{q})$. When Q is λ-sparse, this condition is satisfied if

$$\sqrt{6\delta} < \frac{\lambda}{1 + 8\lambda}. \tag{1}$$

This means that the Delaunay refinement has to maintain a λ-sparse sample Q where λ satisfies the inequality (1).

Bounding conditions

Inequality (1) says that λ needs to satisfy a lower bound in terms of δ. We will see later that it also needs to satisfy some upper bounds for guaranteeing termination of the algorithms. For reference to these conditions on λ, we state them with Condition 1 and 2 and refer them together as Bounding condition on λ.

$$\mathit{Condition}\ 1 : \sqrt{6\delta} < \frac{\lambda}{1 + 8\lambda} \text{ and } \lambda < \frac{1}{48}.$$
$$\mathit{Condition}\ 2 : \beta(12\lambda) + \alpha(6\lambda) + \alpha(4\lambda) + 3\mu < \frac{\pi}{2}.$$

Observation 5.1 *If Condition 1 holds, $\delta < \lambda$.*

Recall that the map ν takes a point to its closest point on Σ. It turns out that the map ν restricted to G induces a homeomorphism between G and Σ if δ and μ are sufficiently small(proof in [Dey05]).

Observation 5.2 *If G is (δ, μ)-flat with respect to Σ where δ and μ satisfy the Bounding conditions for some $\lambda > 0$, then G is homeomorphic to Σ and the map ν restricted to G is a homeomorphism.*

6 Termination proofs

The main theorems we prove are:

Theorem 2. *Let G be a polygonal mesh that is (δ, μ)-flat with respect a smooth surface. If there exists a $\lambda > 0$ so that the Bounding conditions hold, then*

(i) MfldRecov *terminates and outputs a manifold Delaunay mesh whose vertex set is λ-sparse,*
(ii) TopoRecov *terminates and outputs a Delaunay mesh homeomorphic to G whose vertex set is λ-sparse.*

Theorem 3. *Let G be a polygonal mesh that is (δ, μ)-flat with respect a smooth surface. If the chosen λ satisfies the Bounding conditions, then* GeomRecov *terminates and outputs a Delaunay mesh homeomorphic to G whose vertex set is λ-sparse.*

The key to the success of the topology recovery phase is that, for sufficiently small δ and μ, there exists a λ satisfying the Bounding conditions. For example, if $\delta = 4 \times 10^{-5}$ and $\mu = 0.1$, one can choose $\lambda = 0.02$. Notice that the requirement on δ is rather too small. First of all, this is an artifact of our proofs. Secondly, when G is a reconstructed mesh from a dense point sample, we will see later that δ will be $O(\varepsilon^2)$ where $\varepsilon < 1$ measures the sampling density and thus the requirement on ε will be less stringent. For geometry recovery phase we explicitly need that the user supplied λ satisfy the Bounding conditions.

We use several lemmas about the Voronoi diagram of Q to prove the above two theorems. A common theme in these lemmas is that if a Voronoi face does not intersect G appropriately, there is a point in G far away from all existing points in Q. Recall that algorithmically we used this result by inserting such a far-away point to drive the Delaunay refinement. A similar line of arguments was used by Cheng, Dey, Ramos and Ray [CDRR04] for meshing smooth surfaces. However, as we indicated before, some of the proofs need different reasoning since G is not smooth. We skip those proofs that can be adapted from Cheng et al. [CDRR04] with only minor changes and include the ones that need fresh arguments. In particular, Lemma 9, which needs new arguments, is an essential ingredient for other lemmas. In what follows we assume G to be (δ, μ)-flat with respect to a smooth surface Σ for some appropriate $\delta < 1$ and $\mu < 1$. All lemmas in this section involve faces of $\mathrm{Vor}\, Q$ where Q is λ-sparse for a λ satisfying the Bounding conditions.

Lemma 9. *Let $e \in V_q$ be a Voronoi edge that intersects G either (i) tangentially at a point, or (ii) transversally at two or more points. Let x be the point among these intersection points which is furthest from q. Then, x is at least $\lambda f(\tilde{q})$ away from q.*

Proof. Suppose that contrary to the lemma $\|q - x\| < \lambda f(\tilde{q})$. Observe that

$$\begin{aligned}\|\tilde{x} - \tilde{q}\| &\le 2(\lambda + \delta) f(\tilde{q}) \text{ (Lemma 2)} \\ &\le 4\lambda f(\tilde{q}) \text{ by Observation 5.1.}\end{aligned}$$

By Lemma 5, $\angle \tilde{\mathbf{n}}_{\tilde{x}}, \tilde{\mathbf{n}}_{\tilde{q}} \le \alpha(4\lambda)$.

Orient e along $\mathbf{n}_{pqr}$ where pqr is the dual Delaunay triangle of e. The conditions (i) and (ii) of Lemma 7 hold for pqr since $G \subset \delta\Sigma$, Q is λ-sparse and the Bounding condition 1 holds. The circumradius of pqr is no more than $\|q - x\| \le \lambda f(\tilde{q})$ satisfying the condition (iii) of Lemma 7. So, we have

$$\begin{aligned}\angle e, \tilde{\mathbf{n}}_{\tilde{x}} &\le \angle \mathbf{n}_{pqr}, \tilde{\mathbf{n}}_{\tilde{q}} + \angle \mathbf{n}_{\tilde{q}}, \tilde{\mathbf{n}}_{\tilde{x}} \\ &\le \beta(12\lambda) + \alpha(6\lambda) + \alpha(4\lambda) \text{ (Lemma 7, 5).}\end{aligned}$$

Let g be a polygon in G containing x. Since $\angle \mathbf{n}_g, \tilde{\mathbf{n}}_{\tilde{x}} \le \mu$, oriented e makes an angle of at most $\beta(12\lambda) + \alpha(6\lambda) + \alpha(4\lambda) + \mu$ with $\mathbf{n}_g$.

Suppose, e intersects G tangentially at x. Then, e makes at least $(\pi/2) - 2\mu$ angle with the normal of one of the polygons containing x since the normals of adjacent polygons in G make at most 2μ angle (Lemma 1). We reach a contradiction if

$$\beta(12\lambda) + \alpha(6\lambda) + \alpha(4\lambda) + \mu < \frac{\pi}{2} - 2\mu,$$

which is satisfied by the Bounding condition 2 proving (i).

Because of the previous argument we can assume that e intersects G only transversally. Let y be an intersection point next to x on e and $g' \in G$ be a polygon containing y. The distance $\|q - y\|$ is at most $\lambda f(\tilde{q})$ as the furthest intersection point x from q is within $\lambda f(\tilde{q})$ distance from it. Then, applying the same argument as for x, we get that e makes an angle of at most $\beta(12\lambda) + \alpha(6\lambda) + \alpha(4\lambda) + \mu$ with $\mathbf{n}_{g'}$. The oriented e leaves the bounded component of $\mathbb{R}^3 \setminus G$ at one of x and y. At this exit point, e makes more than $\frac{\pi}{2}$ angle with the oriented normal of the corresponding polygon. This means we reach a contradiction if

$$\beta(12\lambda) + \alpha(6\lambda) + \alpha(4\lambda) + \mu < \frac{\pi}{2}.$$

This inequality is satisfied if the Bounding condition 2 holds.

Next lemma says that if a Voronoi facet does not intersect G properly, one can find a far away point to insert. Its proof depends on Lemma 9 and is very similar to Lemma 8 of [CDRR04].

Lemma 10. *Let F be a Voronoi facet in V_q where $F \cap G$ contains at least two closed topological intervals. Furthermore, assume that each Voronoi edge intersects G in at most one point. The furthest point in $F \cap G$ from q which lies on a Voronoi edge of V_q is at least $\lambda f(\tilde{q})$ away from q.*

Next three lemmas deal with different cases of the boundaries of the manifold in which a Voronoi cell intersects G. We skip the proof of Lemma 11 since it is same as that of Lemma 11 of [CDRR04] and refer to [Dey05] for the proofs of others.

Lemma 11. *For a vertex q in $\mathrm{Vor}\, Q$ let $W = V_q \cap G$ is a manifold with at least two boundaries both of which intersect Voronoi edges of V_q. Then the point $x \in W$ furthest from q is within $\lambda f(\tilde{q})$ distance.*

Lemma 12. *Let $F \subset V_q$ intersect G in a cycle. The point in $F \cap G$ furthest from q is at least $\lambda f(\tilde{q})$ distance away from q.*

Lemma 13. *For a point $q \in Q$ let $W = V_q \cap G$ intersect no Voronoi edge. Then, the point $x \in W$ furthest from q is at least $\lambda f(\tilde{q})$ away from q.*

The next lemma will ensure that the point inserted by VCELL cannot be very close to all other points in Q (proof in [Dey05]).

Lemma 14. *Let x be a point in G and $W \subset G$ be a subset so that $x \in W$ and $\|x - y\| \leq \lambda f(\tilde{x})$ for each point $y \in W$. Furthermore, W has a single boundary. Then, W is a 2-disk when $\delta < 1/5$ and $\lambda < 1/4$.*

Observation 6.1 *Let p and q be any two points in $\delta\Sigma$ with $\|p - q\| \geq \lambda f(\tilde{q})$. Then, for $\delta < \lambda$, $\|p - q\| \geq \frac{\lambda}{(1+4\lambda)} f(\tilde{p})$.*

Proof. If $\|p - q\| \geq \lambda f(\tilde{p})$, there is nothing to prove. So, we assume $\|p - q\| < \lambda f(\tilde{p})$. By Observation 2, $\|\tilde{p} - \tilde{q}\| \leq 2(\lambda + \delta) f(\tilde{p}) \leq 4\lambda f(\tilde{p})$. By Lipschitz property of $f()$, $f(\tilde{q}) \geq \frac{1}{1+4\lambda} f(\tilde{p})$ which applied to the given inequality $\|p - q\| \geq \lambda f(\tilde{q})$ yields $\|p - q\| \geq \frac{\lambda}{1+4\lambda} f(\tilde{p})$.

Now we have all ingredients to prove Theorem 2.

Proof. (Theorem 2) We show that the vertex set Q remains λ-sparse for a $\lambda > 0$ throughout MFLDRECOV and TOPORECOV. Termination of these algorithms is immediate since only finitely many points can be accommodated in the bounded domain $\delta\Sigma$ with non-zero nearest neighbor distances.

Initially Q is λ-sparse trivially since it contains a single point from each component of G. Let p be any point inserted by any of the subroutines called by MFLDRECOV and TOPORECOV.

We claim that p is at least $\lambda f(\tilde{q})$ distance away from all other points in Q where q is a nearest point to p.

If VEDGE inserts p, the claim is true by Lemma 9. If DISK inserts p, then there was an existing point $q \in Q$ so that $p \in V_q$ and Δt_q was not a disk. If Δt_q were empty, $G \cap V_q$ did not intersect any Voronoi edge. Then, by Lemma 13 the claim is true. If Δt_q were not empty, either (i) there was an edge e of Δt_q not having two triangles incident to it, or (ii) there were two topological disks pinched at q. For (i) let F be the dual Voronoi facet of e. If e had a single triangle incident to it, G intersected a Voronoi edge in F either tangentially or at least twice. Both of these cases would have been caught by VEDGE test. So, e had three or more incident triangles. Hence F intersected G in more than one topological interval and the claim follows from Lemma 10. For (ii) observe that $G \cap V_q$ had two or more boundaries. The claim follows from Lemma 11. If p is inserted by FCYCLE, apply Lemma 12 for the claim. For VCELL observe that if it inserts a point, the subset $W = V_q \cap G$ is not a 2-disk. Also, since it is called after all other tests, W has a single boundary. Then, Lemma 14 is violated which implies the claim.

Applying Observation 6.1, we get that p is at least $\frac{\lambda}{1+4\lambda} f(\tilde{p})$ away from all other points of Q. Applying Observation 6.1 once more to any other point $s \in Q$, we get that

$$\|s - p\| \geq \frac{\lambda}{1 + 8\lambda} f(\tilde{s})$$

proving Q remains λ-sparse after insertion of p.

Next, we prepare to prove Theorem 3. Recall that q^+ and q^- are the two poles defined for a vertex q. First, we show that these poles are far away from q.

Lemma 15. *If Q is λ-sparse and the Bounding conditions hold, then for each vertex $q \in Q$, $\min\{\|q - q^+\|, \|q - q^-\|\} \geq \frac{f(\tilde{q})}{12}$.*

Proof. Following the proof of Lemma 7, we get two empty balls that are tangent to each other at q whose radii are at least $\frac{f(\tilde{q})}{12}$. The centers of these empty balls reside inside V_q. Also, they are separated locally within V_q by G. The points q^+ and q^- are even further from q than these centers. The lemma follows.

Proof. (Theorem 3) Since GEOMRECOV calls TOPORECOV it is sufficient to argue that if Q is λ-sparse, then it remains so after inserting a point c in the steps 3(i) and 3(ii) of GEOMRECOV. First, consider step (i). Since Q is λ-sparse, $\ell(t) > \frac{\lambda}{1+8\lambda} f(\tilde{q})$ where q is a vertex of the shortest edge in t. Then c is at least $\lambda f(\tilde{q})$ distance away from q. Next, consider setp (ii). The radius $r(t)$ is more than $12\lambda h_q$ which by Lemma 15 is at least $\lambda f(\tilde{q})$. Therefore, the point c is at least $\lambda f(\tilde{q})$ distance away from q.

Observe that in both cases q is also a nearest point of c in Q. Therefore, following the proof of Observation 6.1, we get that Q remains λ-sparse after the insertion of c.

7 Input Meshes

We have already seen that when δ and μ are sufficiently small, there exists a $\lambda > 0$ satisfying the Bounding conditions. This means that, for a mesh G that is (δ, μ)-flat with respect to a smooth surface for sufficiently small values of δ and μ, the manifold recovery and topology recovery terminate. Figure 4 shows two examples of polygonalized surfaces on which our remeshing algorithm is applied.

An interesting and perhaps the most important input for our algorithms would be the polygonal meshes created from point cloud data. When G is such a mesh we show that it is necessarily (δ, μ)-flat with respect to the surface Σ from which the point cloud is drawn. Of course, the point cloud should be sufficiently dense. A point set $P \subset \Sigma$ is called an ε-sample if $d(x, P) \leq \varepsilon f(x)$ for each point x of Σ [ABE98]. When G is reconstructed from an ε-sample P of Σ, it becomes (δ, μ)-flat where δ and μ depend on the sampling density ε. In general we can assume that any of the provable reconstruction algorithms [Dey03] is applied to create G from P. What is important is that all these algorithms produce triangles in G with small circumradius. For precision we assume that G is created from P using the Cocone algorithm of Amenta, Choi, Dey and Leekha [ACDL00]. Then, the following fact holds.

Fact 7.1 *Each triangle $t \in G$ has a circumradius of $\frac{1.15\varepsilon}{1-\varepsilon} f(p)$ where p is any vertex of t.*

We can derive bounds on δ and μ from the above fact. It turns out that $\mu = O(\varepsilon)$ while $\delta = O(\varepsilon^2)$. The proofs appear in [Dey05].

Lemma 16. *Let x be any point in a triangle $t \in G$. We have $\angle \mathbf{n}_t, \tilde{\mathbf{n}}_{\tilde{x}} \leq \beta(\varepsilon) + \alpha(4.6\varepsilon')$ where $\varepsilon' = \frac{\varepsilon}{1-\varepsilon}$.*

Lemma 17. *Let x be any point in a triangle $t \in G$. Then $\|x - \tilde{x}\| \leq \left(\frac{4.6\varepsilon'}{1-4.6\varepsilon'}\right)^2 f(\tilde{x})$.*

From the above two lemmas, we find that $\delta \leq \left(\frac{4.6\varepsilon'}{1-4.6\varepsilon'}\right)^2$ and $\mu \leq \beta(\varepsilon) + \alpha(4.6\varepsilon')$ for G. If $\varepsilon \leq 0.001$, we get $\delta = 2 \times 10^{-5}$ and $\mu = 0.009$. This allows to choose $\lambda = 0.02$ to satisfy the Bounding conditions. Figure 4 shows examples of reconstructed meshes that are remeshed.

Acknowledgements. We acknowledge the support of Army Research Office, USA under grant DAAD19-02-1-0347 and NSF, USA under grants DMS-0310642 and CCR-0430735.

[Alliez03] P. Alliez, E. C. de Verdière, O. Devillers and M. Isenburg. Isotropic surface remeshing. *Proc. Shape Modeling Internat.* (2003).

[AB98] N. Amenta and M. Bern. Surface reconstruction by Voronoi filtering. *Discr. Comput. Geom.* **22** (1999), 481–504.

[ABE98] N. Amenta, M. Bern and D. Eppstein. The crust and the β-skeleton: combinatorial curve reconstruction. *Graphical Models and Image Processing*, **60** (1998), 125-135.

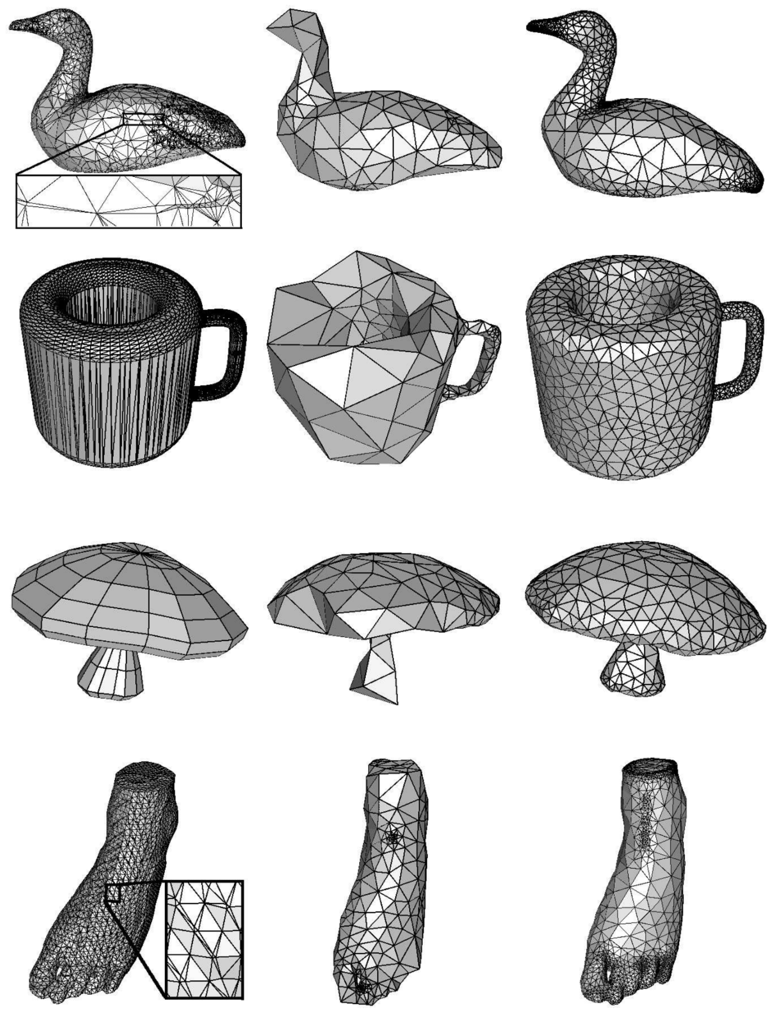

Fig. 4. The input meshes are shown in the first column. Second and third columns show the output meshes for two different levels of refinements. (a) First row shows remeshing of a reconstructed surface from a point cloud. Notice the skinny triangles present in the input. (b) Second and third rows show remeshing of a designed triangular and non-triangular surface respectively. (c) Fourth row shows remeshing of an iso-surface extracted by a marching cube algorithm from volume data. Notice that the artificial small feature created in the iso-surface is meshed densely.

[ACDL00] N. Amenta, S. Choi, T. K. Dey and N. Leekha. A simple algorithm for homeomorphic surface reconstruction. *Internat. J. Comput. Geom. Applications* **12** (2002), 125–141.

[BO03] J.-D. Boissonnat and S. Oudot. Provably good surface sampling and approximation. *Eurographics Sympos. Geom. Process.* (2003), 9–18.

[CDES01] H.-L. Cheng, T. K. Dey, H. Edelsbrunner and J. Sullivan. Dynamic skin triangulation. *Discrete Comput. Geom.* **25** (2001), 525–568.

[CDRR04] S.-W. Cheng, T. K. Dey, E. A. Ramos and T. Ray. Sampling and meshing a surface with guaranteed topology and geometry. *Proc. 20th Annu. Sympos. Comput. Geom.* (2004), 280–289.

[CDRR204] S.-W. Cheng, T. K. Dey, E. A. Ramos and T. Ray. Quality meshing for polyhedra with small angles. *Proc. 20th Annu. Sympos. Comput. Geom.* (2004), 290–299.

[Chew89] L. P. Chew. Guaranteed-quality triangular meshes. Report TR-98-983, Comput. Sci. Dept., Cornell Univ., Ithaca, New York, (1989).

[Chew93] L. P. Chew. Guaranteed-quality mesh generation for curved surfaces. *Proc. 9th Annu. ACM Sympos. Comput. Geom.*, (1993), 274–280.

[Dey03] T. K. Dey. Curve and surface reconstruction. Chapter in *Handbook on Discrete and Computational Geometry*, 2nd Edition (2004), eds. J. Goodman and J. O'Rourke, CRC press, Boca Raton, Florida.

[Dey05] T. K. Dey, G. Li and T. Ray. Polygonal surface remeshing with Delaunay refinement. Extended version, 2005. http://www.cse.ohio-state.edu/∼tamaldey/papers.html.

[ES97] H. Edelsbrunner and N. Shah. Triangulating topological spaces. *Internat. J. Comput. Geom. Appl.* **7** (1997), 365–378.

[PW04] S. Pav and N. Walkington. A robust 3D Delaunay refinement algorithm. *Proc. Intl. Meshing Roundtable* (2004).

[PB01] P. P. Pébay and T. J. Baker. Comparison of triangle quality measures. *Proc. 10th Internat. Meshing Roundtable*, Sandia National Laboratories, (2001), 327-340.

[Rup95] J. Ruppert. A Delaunay refinement algorithm for quality 2-dimensional mesh generation. *J. Algorithms*, **18**, (1995), 548–585.

[Shew98] J. R. Shewchuk. Tetrahedral mesh generation by Delaunay refinement. *Proc. 14th Annu. ACM Sympos. Comput. Geom.*, (1998), 86–95.

[Sifri] O. Sifri, A. Sheffer and C. Gotsman. Geodesic-based surface remeshing. *Proc. Internat. Meshing Roundtable* (2003).