

Delaunay Mesh Construction

Ramsay Dyer, Hao Zhang, and Torsten Möller[†]

GrUVi Lab, School of Computing Science, Simon Fraser University, Canada

Abstract

We present algorithms to produce Delaunay meshes from arbitrary triangle meshes by edge flipping and geometry-preserving refinement and prove their correctness. In particular we show that edge flipping serves to reduce mesh surface area, and that a poorly sampled input mesh may yield unflippable edges necessitating refinement to ensure a Delaunay mesh output. Multiresolution Delaunay meshes can be obtained via constrained mesh decimation. We further examine the usefulness of trading off the geometry-preserving feature of our algorithm with the ability to create fewer triangles. We demonstrate the performance of our algorithms through several experiments.

Categories and Subject Descriptors (according to ACM CCS): I.3.5 [Computer Graphics]: Curve, surface, solid and object representations

1. Introduction

Delaunay triangulations have been studied extensively in computational geometry and have found applications in many fields, including numerical analysis, computer graphics, image processing, and geographical information systems. The classical theory of Delaunay triangulations has been developed in 2D; coverage on this can be found in any computational geometry text, e.g. [dBvKOS98, Ede01]. A variety of algorithms for producing planar Delaunay triangulations of bounded domains or point sets are known [Ede01].

2D Delaunay triangulations can be generalized to higher dimensions; e.g., in \mathbb{R}^3 we are concerned with Delaunay tetrahedralizations [She97]. Subject to some constraints, they can also be extended to non-Euclidean geometries. Most notably, the *intrinsic Delaunay triangulation* or *iDT* of a sufficiently dense set of points on a Riemannian manifold is well defined in terms of geodesic curves [LL00]. Bobenko and Springorn [BS05] defined iDTs of the vertex set of piecewise flat surfaces. In the mesh setting, the set of vertices of an iDT are the original vertices of the mesh while the edges may be formed by geodesic lines over the polyhedral mesh surface.

It has been noted and empirically verified that geometric processing over triangle mesh surfaces which involve nu-

merical evaluation of elliptic PDEs, e.g., for mesh parameterization and reaction diffusion textures, can benefit greatly from having an iDT [FSBS06]. Several desirable properties of the linear finite element discretization of the Laplace-Beltrami operator have also been identified for iDTs [BS05], e.g., guaranteed validity of discrete harmonic maps [Flo98].

Recently, Dyer et al. [DZM07] defined a *Delaunay mesh* to be a manifold triangle mesh whose edges form an iDT of its vertices. The importance of Delaunay meshes has been recognized [FSBS06]; having a Delaunay mesh for geometry processing would remove the need to store the connectivity and geometry of the iDT in addition to that of the mesh. However, to date there have been no algorithms that can guarantee such meshes. Dyer et al. [DZM07] allude to the use of edge flipping in the spirit of the traditional scheme [Law77, FSBS06], where any *mesh* edge that is *not locally Delaunay* or *NLD*, is flipped. But flipping mesh edges changes the geometry of the domain and the termination proofs for the traditional algorithms no longer apply. Although the geodesic edge flipping algorithm [FSBS06] does carry a termination guarantee, a mesh obtained by straightening the edges (geodesic lines over the mesh surface) of an iDT is not Delaunay in general [DZM07].

In this paper, we develop construction algorithms for Delaunay meshes and make the following contributions.

- We propose a *geometry-preserving* algorithm (i.e., one in which the polyhedral surface of the output mesh is iden-

[†] Email: rhdyer, haoz, torsten@cs.sfu.ca

tical to that of the input mesh) for producing a *Delaunay remeshing* of a given manifold triangle mesh and prove that the algorithm is guaranteed to terminate (Section 3).

- With geometric distortion tolerated, mesh edge flipping which can alter mesh geometry may be applied. We prove a new result that mesh edge flipping based on the NLD criterion is *area reducing* (Section 4).
- The only obstacle to constructing Delaunay meshes via edge flipping as described above is the existence of *unflippable edges*: an edge is unflippable if its opposing edge also exists in the mesh. We argue that the unflippable edges result from poor sampling. By combining edge flipping with refinement, we arrive at an algorithm that is guaranteed to produce Delaunay meshes (Section 4).
- Relying on constrained optimization [GW03] and a quadric-based error metric [GH97], we develop a mesh decimation algorithm to produce a series of Delaunay meshes, at multiple levels of detail, which approximate a given dense Delaunay mesh (Section 5).

While strict geometry preservation may sometimes be desirable, a tolerance of geometric distortion can offer more control over triangle count. Often the polyhedral geometry of the mesh is an approximation to a smooth surface, so there is little motivation to be strictly faithful to the mesh geometry. Nonetheless the geometry-preserving refinement steps serve to guarantee the correctness of the edge flipping algorithm. We can also apply appropriate thresholding to ensure a low distortion by preventing feature edges from being flipped.

2. Related work

Works on edge flipping to improve triangulation quality can be traced back to Lawson [Law77]. In the planar case, it is known that Delaunay edge flipping, i.e., swapping any NLD edge e with the other diagonal of the quad formed by the two triangles incident at e , monotonically reduces several functionals defined for planar triangulations [Mus97] and that the respective local minima are unique. Thus such edge flips do produce planar Delaunay triangulations, and the time complexity is known to be quadratic [Ede01]. Bobenko and Springborn [BS05] have shown that the same edge flipping scheme works on a piecewise flat surface, resulting in an iDT, where their termination proof is essentially the same as in the planar case via the harmonic index [Mus97]. However, the situation for Delaunay meshes is different. In particular, edge flipping only leads to one, of possibly many Delaunay meshes (Section 4.2), and it does not always result in a Delaunay mesh due to topological constraints (Section 4.3).

Edge flippings designed to minimize curvature measures have also been proposed, e.g., [DHKL01]. These algorithms produce visually pleasing meshes, but in general they are far from being Delaunay. A recent paper focuses more on triangle element quality by edge flipping so as to minimize

the discrete Willmore energy [ABR06]. All of these mesh edge flipping algorithms tend to converge to local rather than global minima. This includes the surface area minimization algorithm of O'Rourke [O'R81], which, in light of our new insight, turns out to be performing Delaunay edge flips.

The topological constraints which prevent guaranteed generation of Delaunay meshes via edge flipping are due to poor sampling. Thus appropriate refinement is necessary. Many Delaunay refinement techniques are known in the planar setting [Che89, Rup95, RI97] and for 3D tetrahedralizations [She97], where the primary concern is element quality, e.g., angles. Our goal is to guarantee algorithm termination and the scheme we adopt exploits a heuristic, the method of concentric shells, introduced by Ruppert [Rup95]. Ruppert used this heuristic to improve the performance of his Delaunay refinement algorithm in the presence of small input angles, but his algorithm held no termination guarantees in that context. The termination proof we give is original.

Many previous works on mesh generation and remeshing utilize the Delaunay concept and can produce meshes that are close to being Delaunay. Chen and Bishop [CB97] use Delaunay refinement in a 2D parametric space from which a triangulation is mapped back in 3D. Peyré and Cohen [PC03] describe a farthest point sampling method based on geodesic distances computed via fast marching. By taking the dual of the implied intrinsic Voronoi diagram [DZM07] of the sampled points, a triangulation is obtained which closely approximates the iDT of the sample points.

Another class of surface reconstruction and remeshing algorithms rely on the notion of a restricted Delaunay triangulation [ES94], which can be simply characterized as the dual of the restricted Voronoi diagram. There are a number of surface reconstruction algorithms which exploit this concept [AB98, ACDL00, DLR05]. Cheng et al. [CDRR04] use the topological properties of the restricted Voronoi diagram to drive a sampling scheme for triangulating smooth surfaces. For remeshing, Chew [Che93] adapts his Delaunay refinement technique [Che89] to curved surfaces, producing a restricted Delaunay triangulation of the new sample points with respect to the input mesh. Boissonnat and Oudot [BO03] improve on this algorithm and provide guarantees on the sampling distribution and angle bounds. Later, the algorithm was adapted to probe an unknown surface [BGO05]. Nonetheless, the restricted Delaunay triangulation does not define a Delaunay mesh in general [DZM07]. Finally, Li and Zhang [LZ06] have proposed a remeshing algorithm to produce guaranteed nonobtuse meshes, which are necessarily Delaunay. However, their approach is only approximating and cannot offer any form of interpolation of the original input mesh.

3. Geometry-preserving Delaunay remeshing

A mesh is Delaunay when it has no NLD edges [BS05], where an edge is NLD if the sum of the two face angles op-

```

while mesh  $M$  contains an NLD edge  $e = [p, q]$  do
  if  $e$  is not a physical edge then
    Edge flip at  $e$ .
  else
    if  $p$  is an original mesh vertex then
      Split  $e$  at  $SV(p, e)$ .
    else
      Split  $e$  at  $SV(q, e)$ .
    end if
  end if
end while

```

Algorithm 1: Geometry-preserving Delaunay remeshing.

posite to it is greater than π . For a mesh which tessellates a densely sampled smooth curved surface, edge flipping (Section 4) may be a fast and simple means of obtaining a Delaunay mesh. However, if a mesh is coarsely sampled, e.g., to exploit the principle curvatures over the surface, edge flipping may be inappropriate; the change in geometry may be too large. At the expense of adding new vertices, we can produce a geometry-preserving Delaunay mesh via refinement.

3.1. Remeshing algorithm

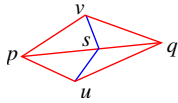


Figure 1: The original edge $[p, q]$ is split at s , creating two planar (blue) and two physical (red) edges incident to s .

The refinement proceeds by subjecting NLD edges to an *edge split*, which inserts a new vertex along an *original edge* of the input mesh and connects the newly inserted vertex with two vertices opposite to the current mesh edge being split, as shown in Figure 1. New vertices added during

refinement are called *split vertices* to distinguish them from the *original vertices* of the mesh. An edge that has a non-zero dihedral angle (non-coplanar adjacent faces) and is an original mesh edge, or part of an original mesh edge is called a *physical edge*. If e' is a portion of a pre-existing edge e , we say that e' is *embedded* in e . Edges between coplanar faces are called *planar edges*. Only physical edges are split in our algorithm; planar edges may be flipped without affecting the geometry.

A naive edge splitting algorithm, using edge bisection for example, will not terminate in general. The kind of refinement problem we are facing has been studied in the planar setting, where one seeks a *conforming Delaunay triangulation* [ET93]. Our scheme, outlined in Algorithm 1, employs the method of *concentric shells* introduced by Ruppert [Rup95].

The split vertex $SV(p, e)$, with p a vertex of edge e , is defined to be the point s along e that is the closest to the

midpoint m of edge e such that the length $||[p, s]|| = 2^k \delta$, for some (possibly negative) integer k . Formally,

$$SV(p, e) = \operatorname{argmin}_{s \in e, ||[p, s]|| = 2^k \delta, k \in \mathbb{Z}} ||s - m||. \quad (1)$$

The factor δ may be any positive number. In our implementation, we set it to unity.

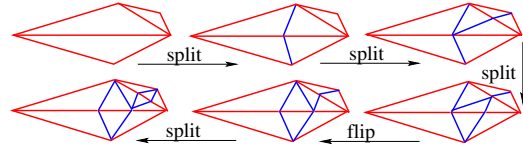


Figure 2: Geometry-preserving Delaunay remeshing on a simple example. Physical edges are shown in red and planar edges in blue.

Our Delaunay remeshing scheme combines edge flipping with edge splitting. Since only edges interior to the planar faces of the mesh can be flipped, the algorithm is geometry-preserving. Figure 2 shows our remeshing algorithm at work on a simple example.

3.2. Termination proof

It suffices to prove termination of the refinement steps, as all our edge flips are planar and they terminate as in the planar case [Law77]. Our proof is by contradiction while assuming that there are no degenerate triangles in the mesh. Since a mesh surface is compact, the Bolzano-Weierstrass theorem implies that any non-terminating refinement must produce an *accumulation point* a . That is, any arbitrarily small neighborhood of a contains infinitely many inserted split vertices. The algorithm never inserts vertices interior to the original mesh faces. Thus there are only two possible cases:

1. a lies on the interior of an original mesh edge e :

Then there is an $\epsilon > 0$ such that the geodesic disk O of radius ϵ centred at a contains none of the original vertices nor any portion of an original edge other than e . As edge splits happen along e , let us keep track of the sequence h_0, h_1, \dots of mesh edges which contain a , where $h_0 = e$. Since a is an accumulation point, this edge sequence must be infinite. Now let us show that $\frac{|h_i|}{|h_{i-1}|} \leq \frac{2}{3}$ for all $i > 0$.

Suppose otherwise, i.e., $|h_i| > 2|h_{i-1}|/3$. Let $h_{i-1} = [p, q]$ and let h_i be obtained by splitting edge $[p, q]$ at the “power of 2 split” s , as shown in Figure 3(a). Let m be the midpoint of $[p, q]$ and r the midpoint of $[p, s]$. Then

$$|[r, m]| = \frac{|h_{i-1}|}{2} - \frac{|h_i|}{2} < \frac{|h_{i-1}|}{6}.$$

But since

$$|[s, m]| = |h_i| - \frac{|h_{i-1}|}{2} > \frac{2|h_{i-1}|}{3} - \frac{|h_{i-1}|}{2} = \frac{|h_{i-1}|}{6},$$

we have $|[r, m]| < |[s, m]|$. Thus, according to equation (1),

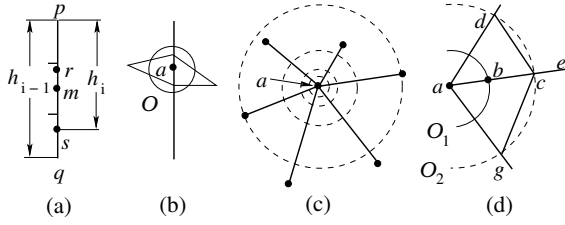


Figure 3: Figures for termination proof. (a) If $\frac{|h_i|}{|h_{i-1}|} > \frac{2}{3}$, then r , the midpoint of $[p, s]$, should have been the split vertex, instead of s . (b) A geodesic disk O centered at a contains an edge covering a . Since the vertices opposite to the edge lie outside O , the edge cannot be NLD. (c) Eventually, all edge splits towards a must occur on concentric geodesic circles, centered at a , with radii $2^k \delta$, where $k \in \mathbb{Z}$. (d) If $|[a, d]| \geq |[a, c]|$ and $|[a, g]| \geq |[a, c]|$, $[a, c]$ cannot be NLD.

vertex r , also a “power of 2 split” since $|[p, r]| = |[p, s]|/2$, should have been chosen as the split vertex instead of s . This is a contradiction.

Since $\frac{|h_i|}{|h_{i-1}|} \leq \frac{2}{3}$ for all $i > 0$, we conclude that there must be some h_i that falls entirely inside the geodesic disk O . Clearly, such an edge h_i cannot possibly be NLD since barring degeneracy, its two opposite vertices must lie outside the disk O , as shown in Figure 3(b). Therefore, h_i , which contains a , will not be split further. This contradicts the assumption that a is an accumulation point.

2. a is an original mesh vertex:

Since a is an accumulation point, there is an infinite accumulation of edge splits towards a . By Algorithm 1, eventually, such splits must all occur on geodesic circles centered at a with radii $2^k \delta$, $k \in \mathbb{Z}$, as shown in Figure 3(c). Let us consider one such geodesic circle O_1 that is sufficiently small that all original mesh edges meeting at a extend beyond the next two concentric geodesic circles, O_2 and O_3 . Suppose that b , along some edge e , is the very first split vertex that is created on O_1 , as shown in Figure 3(d). It follows that b must bisect some NLD edge $[a, c]$ embedded in e , where c lies on O_2 .

However, we can argue that edge $[a, c]$ cannot possibly be NLD, since the lengths of edges $[a, d]$ and $[a, g]$ in the quad containing vertices a and c must be greater than or equal to $|[a, c]|$; this is due to our assumption that b is the first split vertex created inside O_2 . To see that $|[a, d]|, |[a, g]| \geq |[a, c]|$ implies that $[a, c]$ is not NLD, we note that $\angle adc + \angle agc \leq \pi/2 + \pi/2 = \pi$. We have arrived at a contradiction. Proof is complete.

3.3. Meshes with boundaries

In [BS05, FSBS06] meshes with boundaries are dealt with by using a constrained Delaunay triangulation; there was not really another choice available in that context. However, with

refinement, we are by necessity allowed to add Steiner vertices. Therefore, we can also ensure that all the boundary edges are Delaunay, where we define a boundary edge e to be Delaunay if and only if it subtends a *nonobtuse angle*. This ensures that the edge would connect Voronoi neighbours and that it is contained in an empty geodesic disk.

Edge splitting at mesh boundaries works in exactly the same way as before. To show that an accumulation point cannot occur on the interior of an edge, note that if an edge embedded in the mesh boundary falls entirely inside the geodesic disk O , then it cannot be NLD because its opposite vertex lies outside O and thus must form a nonobtuse angle (see Figure 3(b)). In case 2 of the proof, if edge $[a, c]$ happens to fall on the mesh boundary, then $|[a, d]| \geq |[a, c]|$ implies that $\angle adc \leq \angle acd$ and thus $\angle adc \leq \pi/2$. It follows that edge $[a, c]$ is locally Delaunay.

4. Delaunay remeshing via mesh edge flipping

The geometry-preserving refinement algorithm preserves geometry at the expense of adding vertices to the original mesh. A Delaunay mesh can be obtained by allowing physical edges to be flipped. Such operations do not introduce vertices, although they do compromise the geometry. However, due to the possibility of (topologically) unflippable edges, a refinement step may be necessary to ensure that the final mesh is Delaunay.

In our geometry-preserving Delaunay remeshing, there are no physical (mesh) edge flips. We now consider flipping the physical edges of a mesh. This changes the nature of the problem considerably, for now the geometry of the underlying domain is changing each time an edge is flipped. Furthermore, the arguments for termination of edge flipping in the planar case [Mus97] no longer apply and we also need to deal with possible topological constraints.

4.1. Edge flipping and refinement algorithm

Our algorithm takes any manifold triangle mesh M as input and it is superficially similar to the one described by Bobenko and Springborn [BS05]. However, we require an additional refinement step. The order in which the edges are flipped is realized using a priority queue. The priority is related to the extent an edge is locally Delaunay and is set to be the sum of the opposite angles at the edge minus π . Some questions related to our algorithm include: does the edge flipping step terminate and if it does, is the resulting mesh unique? We also need to ensure that throughout the algorithm, the current mesh, possibly refined, is a manifold.

4.2. The geometry of edge flipping and non-uniqueness of Delaunay meshes

In the planar case, an edge e is seen to be a diagonal of a quadrilateral formed by the two triangles adjacent to it. An

```

while mesh  $M$  contains a flippable NLD edge  $e$  do
    Edge flip at  $e$ .
end while
Run geometry-preserving Delaunay remeshing.
    
```

Algorithm 2: Geometry-altering Delaunay remeshing

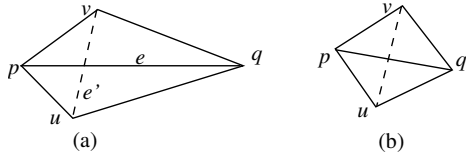


Figure 4: (a) A flip tet $puqv$. e' is the opposing edge to e . (b) A regular tet shows that both diagonals $[p,q]$ and $[u,v]$ can be locally Delaunay.

edge flip entails replacing e with the other diagonal of the quadrilateral. In a geodesic triangulation of a piecewise flat surface, the edge e and its two adjacent faces can be isometrically unfolded onto the plane, so an edge flip can be interpreted the same way [BS05]. However, when flipping a physical edge of a mesh, we cannot unfold the quadrilateral.

Let edge e be joining two vertices p and q in a mesh and adjacent to triangles $f_1 = [p,q,u]$ and $f_2 = [q,p,v]$. Consider the Euclidean line segment $e' = [u,v]$. The edges of f_1 and f_2 together with e' form a tetrahedron σ , as shown in Figure 4(a). We call σ the *flip tet* associated with e . Performing an edge flip on e involves replacing e with the new edge e' and faces f_1 and f_2 with faces $f'_1 = [p,u,v]$ and $f'_2[q,v,u]$. We say that e' is the *opposing edge* to e .

Lemma 1 *If edge e in a (closed) mesh is not locally Delaunay, then its opposing edge e' is.*

This is easy to see since the sum of the angles subtending e and e' is at most 2π . Referring to Figure 4(a), we have

$$\angle puq + \angle uqv + \angle qvp + \angle vpu \leq 2\pi, \quad (2)$$

with equality holding only when p, q, u , and v are coplanar.

While Lemma 1 is true in this setting, its converse, which holds for any planar quad in general position [dBvKOS98], is not true in a mesh. The regular tet, as shown in Figure 4(b), gives an example where both edges e and e' are locally Delaunay. Consequently, there can be multiple Delaunay meshes on the same vertex set and defining the same topological surface. Thus in this sense, without demanding further qualifications, we do not have a general uniqueness theorem for Delaunay meshes, contrary to the case of fixed geometry, be it planar 2D or a fixed piecewise flat surface.

Although Lemma 1 ensures that any flippable edge in our algorithm would improve matters locally, it does not lead to a termination proof. An edge that has been flipped may

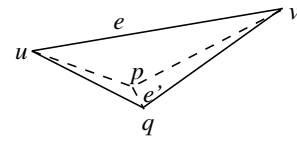


Figure 5: A Delaunay flip may decrease minimum angle and increase harmonic index: Let e be NLD, then flipping e into e' is a Delaunay flip. However, one can bend $\triangle[u,q,v]$ towards $\triangle[u,p,v]$ so that $\angle puq$ becomes arbitrarily small and the Harmonic index of $\triangle[p,u,q]$ becomes arbitrarily large.

become NLD again as a result of other flips. Also, an NLD edge may be unflippable if the resulting mesh would become non-manifold. These issues are addressed in Sections 4.2.1 and 4.3, respectively.

4.2.1. Delaunay edge flipping and area minimization

Termination of edge flipping is traditionally shown by defining a functional on a triangulation and proving that it is increased (or decreased) with each edge flip. However, most of the traditional measures applicable to 2D or intrinsic Delaunay triangulations do not extend to the case of Delaunay edge flips on a mesh. For example, the minimal angle in the triangles adjacent to an edge can sometimes be decreased after a Delaunay flip. Likewise, the harmonic index (sum of the cotangents of the angles), exploited by [BS05], may increase. See Figure 5 for an example. It turns out that a measure that is consistently non-increasing with each Delaunay flip is the mesh surface area:

Theorem 1 *If the sum of the two angles opposite to an edge e is greater than the corresponding angle sum for the opposing edge e' , then the combined area of the two triangles adjacent to e is greater than or equal to the combined area of the two triangles that would be adjacent to e' . Equality arises only when e and e' lie in the same plane.*

Proof We are concerned with the area of the quadrilaterals defined by the two triangles adjacent to each edge. Note that the quadrilaterals can be made planar without distorting the area by unfolding them on the associated edge. The edges e and e' define two different quadrilaterals, but they share the same set of sides. Let a, b, c and d be the lengths of each of the sides. We exploit Bretschneider's formula [Bre42] for the area of a quadrilateral $ABCD$:

$$A = \sqrt{(s-a)(s-b)(s-c)(s-d) - abcd \cos^2 \left(\frac{A+C}{2} \right)}$$

where $s = (a+b+c+d)/2$ is the semi-perimeter and A and C are angles opposite edge e . Let B' and D' be the angles opposite edge e' in the other quadrilateral.

Noting that $\cos^2 \theta$ is monotonically decreasing in the interval

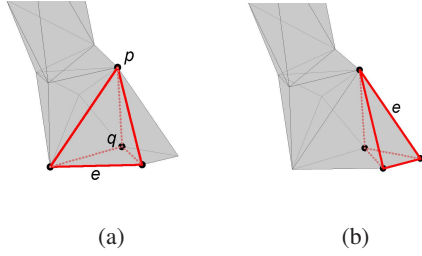


Figure 6: An unflippable edge e . The flip tet is outlined in red. (a) 2-exposed. (b) 3-exposed.

$[0, \pi/2]$ and that $\frac{A+C}{2} > \frac{B'+D'}{2}$ by hypothesis, we have

$$\cos^2\left(\frac{A+C}{2}\right) < \cos^2\left(\frac{B'+D'}{2}\right), \text{ if } \frac{A+C}{2} < \frac{\pi}{2}.$$

Thus, by the Bretschneider's formula, the area of the quadrilateral associated with e' is less than the area of that associated with e . On the other hand, if $\frac{A+C}{2} \geq \frac{\pi}{2}$, then by equation (2), we have

$$\pi/2 \geq \pi - \frac{A+C}{2} \geq \frac{B'+D'}{2}$$

with equality in the planar case. Thus

$$\cos^2\left(\frac{A+C}{2}\right) = \cos^2\left(\pi - \frac{A+C}{2}\right) \leq \cos^2\left(\frac{B'+D'}{2}\right).$$

Again, the Bretschneider's formula gives us a decrease in area except when e and e' lie in the same plane, in which case the area is unchanged. \square

Thus, the surface area of the mesh is monotonically non-increasing as we perform edge flips. Since the number of possible triangulations is finite, these edge flips either terminate, or become stuck in an endless sequence of planar edge flips. This latter possibility is eliminated by traditional termination proofs of Delaunay edge flipping [Law77, BS05]. Thus the edge flipping step of Algorithm 2 terminates.

Next, we discuss the possibility of having NLD edges which still remain after termination of edge flipping. These edges could not be flipped due to topological constraints.

4.3. Unflippable edges

An edge e in mesh M is *unflippable* if its opposing edge also exists in M . Recall that the opposing edge to e is the edge e' that would replace e after a flip. If e' is already in the mesh, flipping e would result in a non-manifold edge.

Examining the flip tet σ associated with e , we identify only three possible cases of unflippable edges. At least two of the faces of σ , those adjacent to e , belong to M . We say σ is *2-exposed*, *3-exposed*, or *4-exposed*, corresponding to it having zero, one, or both of the remaining two faces belonging to M . The 4-exposed case is trivial as it can only occur

when M is a tet (assuming M is connected). Figure 6 depicts the other two cases: 2-exposed and 3-exposed.

We argue that unflippable edges are a manifestation of poor sampling, thus motivating the use of refinement. We appeal to the *well-formedness* criteria of the *intrinsic Voronoi diagram* of the vertices of a mesh M . For a vertex p in M , the (intrinsic) *Voronoi cell*, $\mathcal{V}(p)$, associated with p is the set of points on M closer to p than any other vertex: $\mathcal{V}(p) = \{x \in M \mid d(x, p) \leq d(x, q), \forall q \in V(M)\}$, where d is the geodesic distance. Dyer et al. [DZM07] show that the lack of a well-formed Voronoi diagram is an indication of poor sampling (a good sampling yields a proper Delaunay triangulation [LL00]). We will show that the 2-exposed case violates one of the criteria of well-formedness, namely, neighbouring Voronoi cells meet at a single contiguous edge.

Let e be an NLD edge that is unflippable in a 2-exposed flip tet, as shown in Figure 6(a). The opposing edge, $[p, q]$, must be flippable (although $[p, q]$ is the opposing edge to e , e is not the opposing edge to $[p, q]$). Indeed it can be shown that the contrary assumption would imply that M is non-manifold. Assuming all flippable edges of M are locally Delaunay, as is the case after the flipping step of Algorithm 2, it follows that the Voronoi cells $\mathcal{V}(p)$ and $\mathcal{V}(q)$ of p and q must meet somewhere on the faces adjacent to $[p, q]$. Likewise, since e is NLD, $\mathcal{V}(p)$ and $\mathcal{V}(q)$ must also meet on the faces adjacent to e , resulting in at least two separate intersections between $\mathcal{V}(p)$ and $\mathcal{V}(q)$. This completes our argument. Similar arguments show that the 3-exposed case arises when a vertex has less than three Voronoi neighbours, again indicating a violation of well-formedness.

The problems presented by unflippable edges, as well as boundary edges, can be treated by the final geometry-preserving remeshing step in Algorithm 2, which ensures that all the edges in the final mesh are locally Delaunay.

5. Delaunay mesh decimation

A limitation of our current Delaunay mesh algorithms is that we cannot place a bound on triangle count. Our geometry-preserving refinement may produce an excess of small triangles. In general, it is desirable to have a progression of Delaunay meshes, at multiple levels of details (LODs), which approximate an initial dense Delaunay mesh. In practice, any resource-constrained application can benefit from LOD representations to avoid redundancy and allow for more effective use of the triangle budget.

In this section, we describe a mesh decimation algorithm for LOD modeling with Delaunay meshes. It has been adapted from a similar scheme for nonobtuse meshes [LZ06] and is based on edge collapse prioritized by a quadric error [GH97]. For each edge collapse, the resulting vertex needs to lie in a feasible region to ensure that all the affected edges remain locally Delaunay. The optimal position of the vertex is chosen to minimize the standard quadric error [GH97],

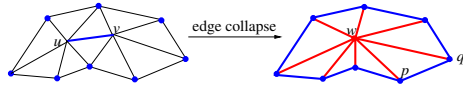


Figure 7: An edge collapse and edges affected (blue and red) need to remain locally Delaunay.

subject to constraints, and the resulting error sets the priority. By choosing the *allowable region* to be a linearized and convexified subset of the feasible region, i.e., being conservative, the decimation algorithm is formulated as a constrained least-square problem and is solved using the OOQP solver of Gertz and Wright [GW03]. Thus the error quadric associated with each edge collapse is minimized subject to linear constraints in the form of planes bounding half spaces.

Since quadric-based edge collapsing schemes using priority queues are well-known [GH97, LZ06], we will concentrate on describing the feasible region and its linearization. In Figure 7, we show an edge collapse $[u, v] \rightarrow w$ after which we need to ensure that all the *incident edges* at w (shown in red), e.g., $[w, p]$ and $[w, q]$, and all the *subtending edges* to w (shown in blue), e.g., $[p, q]$, remain locally Delaunay.

1. Allowable region associated with a subtending edge:

Let $[p, q]$ subtend w . If $[p, q]$ is on the boundary, then any w with $\angle pwq \leq \pi/2$ is feasible; it follows that w lies outside the sphere with $[p, q]$ as diameter; see Figure 8(a).

If $[p, q]$ is an interior edge, let w' be the other subtended vertex. Then any w with $\angle pwq \leq \pi - \angle pw'q$ is feasible. If we draw the circumcircle of $[p, w', q]$ and rotate the arc subtended by edge $[p, q]$ about $[p, q]$, as shown in Figure 8(b), we obtain a surface of revolution which we call a *chordal spheroid*. Any w outside of the spheroid is feasible. To linearize the feasible region, we replace it by a half space defined by a plane L tangent to the sphere or the chordal spheroid defined above (Figure 8(c)). The point of tangency, r , is chosen so that L is parallel to $[p, q]$ and perpendicular to the plane containing $[p, q]$ and the centroid of the collapsing edge.

2. Allowable region associated with an incident edge:

To ensure that an incident edge $[w, p]$ remains locally Delaunay, we need $\angle pow + \angle pqw \leq \pi$, where o and q are the one-ring vertices of w adjacent to p ; see Figure 9(a). If w were to lie in the plane of $[o, p, q]$, then it is constrained by an arc of the circumcircle of $[o, p, q]$. An example of the actual surface bounding the feasible region in 3D is shown in Figure 9(b).

To construct the linearized allowable region, we place r midway between o and q on the arc of the circumcircle that does not contain p . A plane L_1 is constructed such that it contains $[q, r]$ and is perpendicular to the plane of the circumcircle of $[o, p, q]$. This is the boundary of a half-space H_1 that contains the centre of the circumcircle. Half-space H_2 is constructed similarly, with $[o, r]$

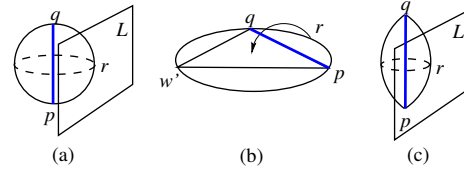


Figure 8: Feasible and allowable regions for subtending edges. (a) If $[p, q]$ is a boundary edge, the feasible region is the complement of the sphere; it is linearized by a plane L tangent to the sphere, as defined in the text. (b) If $e = [p, q]$ is an interior edge, the feasible region is the complement of the chordal spheroid obtained by rotating the subtended arc of the circumcircle of $[p, w', q]$, where w' is the other subtending vertex of e . (c) Similarly to (a), an allowable region is obtained by linearizing the feasible region described in (b).

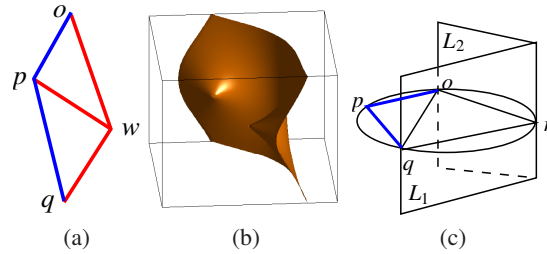


Figure 9: Feasible and allowable regions for incident edges. (a) For $[w, p]$ to be locally Delaunay, we need $\angle pow + \angle pqw \leq \pi$. (b) Plot of actual surface defining the feasible region with respect to o, p , and q . The two tips on the surface correspond to o and q and the supporting plane of $[o, p, q]$ is parallel to the top face of the box. (c) Planes L_1 and L_2 enclose a conservative, linearized allowable region for our optimization; definition of L_1 and L_2 is given in the text.

on its boundary. See Figure 9(c). It can be shown by elementary, but nontrivial calculation that $H_1 \cap H_2$ is indeed contained in the feasible region.

Referring to Figure 7, for an edge collapse $[u, v] \rightarrow w$, the allowable region for w is the intersection of all the allowable regions defined for the incident and subtending edges for w . If this set is empty, then $[u, v]$ will not be collapsed.

6. Experimental results

We have tested our mesh edge flipping and refinement algorithm (Algorithm 2) on a few dozen mesh models and statistics collected on some representative data are reported in Table 1. The first group of models are well-known and serve as examples of typical datasets. Note that the Stanford bunny and Max Plank models have boundary edges. In the second group, we choose two low-resolution meshes that have thin structures; they were obtained via QSlim [GH97]. These

Mesh	#E	Flips(# %)	Splits	Min angle		Max angle		% Small		% Large		Error (ϵ)
Horse	59 547	2 957 (5.0%)	28	1.7	1.6	171.3	164.3	8.7	7.9	1.3	1.2	0.4741%
Hand	74 997	10 935 (14.6%)	0	2.1	3.2	175.3	156.9	15.5	8.5	3.3	0.6	0.1061%
Bunny	104 288	2 202 (2.1%)	82	0.5	7.8	177.6	146.1	2.3	2.1	0.2	0.0	0.2367%
Igea	165 000	19 349 (11.7%)	0	0.1	6.1	179.8	148.3	10.0	5.2	2.2	0.3	0.1037%
Isis	562 926	13 497 (2.4%)	2	0.4	3.1	176.8	163.6	2.5	2.4	0.1	0.0	0.1591%
Max Planck	597 211	12 157 (2.0%)	12	0.1	5.1	178.9	145.2	1.7	1.1	0.3	0.0	0.0992%
Coarse hand	294	99 (33.7%)	34	4.7	12.4	167.4	142.1	27.2	11.4	7.8	1.5	4.3245%
Coarse horse	1 050	231 (22.0%)	36	3.5	12.9	171.3	133.5	17.9	6.9	5.5	0.3	1.6475%
Peyre hand	894	22 (2.5%)	0	24.7	31.0	129.9	112.5	0.6	0.0	0.2	0.0	1.4095%
Peyre horse	1 944	16 (0.8%)	4	12.0	19.4	136.4	113.8	0.2	0.2	0.1	0.0	0.8684%

Table 1: Output statistics for our edge flipping and refinement algorithm (Algorithm 2) on several mesh models. Input mesh sizes are measured by #E, the edge count. We report the number of edge flips performed until termination both in absolute number and as a percentage of #E, as well as the number of edge splits due to unflippable and boundary edges. The next four double columns have before (left sub-column) and after (right sub-column) figures for the minimum (Min angle) and maximum (Max angle) face angles (in degrees) in the meshes, as well as the percentage of angles that are less than 30° (% Small) and the percentage of angles that exceed 120° (% Large). The last column reports the approximation error given by Metro, as a percentage of the Hausdorff distance between the meshes against the length of the bounding box diagonal. The longest running time is recorded on the Max Plank, which took 12.2 seconds to process on a 2.4 GHz Opteron processor.

models have a greater percentage of 3- or 2-exposed tets compared with those from the first group and provide a more rigorous test for the refinement component of our algorithm. The final group contains two meshes that were produced by the remeshing algorithm of Peyré and Cohen [PC03]; these meshes are generally quite close to being Delaunay.

As we can observe from Table 1, our algorithm consistently terminates after flipping and splitting a small fraction of the mesh edges. Denser models tend to incur smaller percentages of such operations. Note that although the Peyré hand and horse models, both coarsely sampled, were obtained from approximate Delaunay geodesic triangulations of the original fine mesh surfaces, the meshes produced do have a small fraction of edges that are NLD. In terms of angle quality, although the smallest angle is not required by theory to increase via Delaunay remeshing, Table 1 does show that in practice it generally does with few exceptions. In no case did the size of the maximum angle, or the percentage of either small ($< 30^\circ$) or large ($> 120^\circ$) angles increase in the Delaunay meshes.

In the last column of Table 1, we report the approximation error ϵ of the Delaunay meshes produced, measured using the well-known Metro tool [CRS98]. As can be seen, the geometric approximation error tends to be large for coarse models. In contrast, errors associated with the Delaunay versions of the corresponding full-resolution models are much smaller; see results for models from the first group.

When used on its own, the geometry-preserving refinement scheme may produce excessive and poorly distributed vertices. However, when strict geometry preservation is relaxed and we allow flipping of edges that have small dihedral angles, we obtain an algorithm that produces more pleasing results. Essentially this is edge flipping with *feature preser-*

vation, where physical mesh edges with dihedral angles exceeding a user-set threshold are flagged and they are split according to the scheme of Section 3 as they appear in the priority queue. A similar argument to that of Section 3.2 carries the termination guarantee to this modified algorithm.

Figure 10(a) shows a close-up of a coarsely sampled dolphin model (6,002 vertices). The mesh edge flipping algorithm does a poor job of preserving the detailed features in the initial shape, as shown in (b), while geometry-preserving refinement creates a large vertex count (15,560 vertices) and uneven vertex distributions, as shown in (c). However, edge flipping with feature preservation presents a nice compromise, shown in (d), where the vertex count is reduced to 7,509 and the Metro error ϵ is reduced from 0.385% to 0.071%, when a dihedral angle threshold of 10° is used.

A tradeoff between feature preservation and vertex count versus the threshold dihedral angle is illustrated in Figure 11. This graph is produced from the Isis model of 187,644 vertices. If the threshold is too high, many vertices are introduced in order to preserve geometry that is probably more associated with the initial discretization than with the intended model. For coarser initial models, the graphs have a similar appearance, but the scale on the vertical axis is much larger. For example, the same graph generated from an initial Isis model of 600 vertices (QSlimmed) has a scale an order of magnitude larger on the vertical axis. Thus the cost, in terms of vertex count, of geometric fidelity, is much greater (relatively) for a coarsely sampled model.

Finally, a multiresolution family of meshes produced by our Delaunay mesh decimation algorithm is given in Figure 12. The decimation generally performs well in preserving features in its attempts to minimize the quadric errors. If needed, our framework easily allows for feature-driven con-

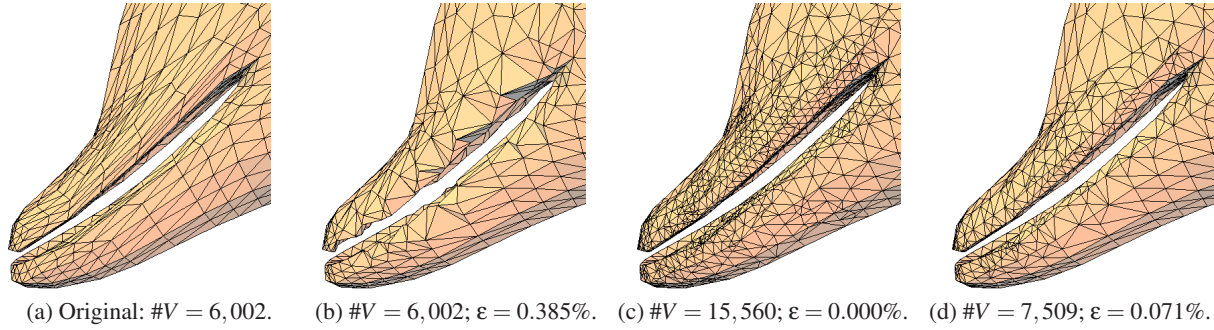


Figure 10: A close-up of results from Delaunay remeshing on the dolphin model. Mesh vertex count (#V) and Metro error (ϵ), measured against the original model, are given below the figures. (a) Original. (b) After unconstrained edge flipping. (c) After geometry-preserving refinement. (d) After feature-preserving edge flipping with a threshold of 10° on dihedral angles.

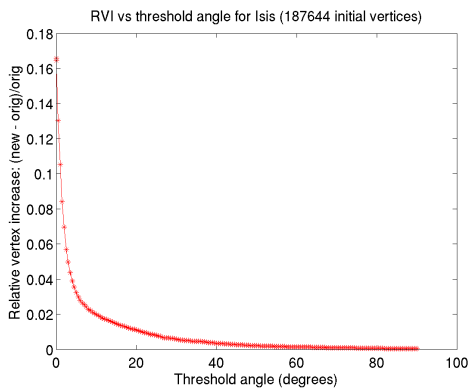


Figure 11: Graph of relative increase in vertex count versus the threshold dihedral angle for feature preserving edge flipping of the Isis model (187,644 vertices initially).

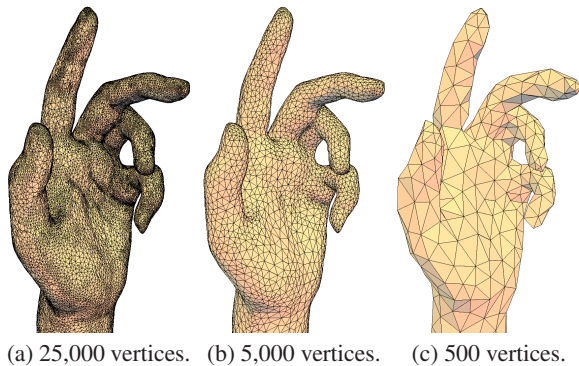


Figure 12: Hand model at three resolutions produced by our decimation algorithm. The Metro errors are: 0.015% for (a), 0.197% for (b) and 1.189% for (c), all measured against a Delaunay remeshing obtained from the original model.

straints to be incorporated. However, our current implementation does not employ lazy evaluation or other heuristics to speed up the optimization and is thus slow: it took over three minutes to generate the 500-vertex model in Figure 12.

7. Conclusions and future work

We have proposed the first Delaunay mesh construction algorithms and proven their correctness. When employed to preserve feature edges, the geometry-preserving refinement scheme nicely supplements the edge flipping algorithm so as to minimize its geometric distortion. It also provides a correctness guarantee for edge flipping in the presence of unflippable edges, which arise as a result of inadequate sampling. Finally, we developed LOD Delaunay mesh representations via quadric-based constrained optimization.

As a current limitation, the refinement algorithm alone may produce uneven triangle distributions with excessively densely sampled clusters and we do not have a bound on the final triangle count. If the input mesh is intended to represent a smooth, or piecewise smooth surface, then there is no reason to be strictly faithful to the piecewise flat geometry of the input mesh. Thus edge flipping with thresholding on the dihedral angles is a sensible option. On the other hand, if the polyhedral geometry of the input is exactly the geometry under investigation, then numerical simulations on the surface will presumably require a much denser set of nodes than those required to represent the geometry alone. In this case a Delaunay refinement algorithm which introduces nodes on the faces of the input mesh would be more suitable than our edge splitting algorithm. A search for such an algorithm together with a consideration of grading and element quality is a possible avenue for future research.

Numerical investigation of convergence properties of discrete operators defined on meshes need to ensure that the multiresolution family of meshes used converge not only point-wise, but also in their normals [HPW06]. We would like to show that a sequence of Delaunay meshes possess

such a property. This has been demonstrated for meshes derived from intrinsic [DLYG06] as well as restricted [AB98] Delaunay triangulations of a smooth surface. Many algorithms have correctness guarantees which rely on the properties of these latter structures. Delaunay meshes may yield a cheaper avenue for producing the same guarantees.

Acknowledgments: We are grateful to John Li for providing the implementation of his nonobtuse decimation algorithm, which we modified to produce Delaunay meshes. We also thank the anonymous reviewers for their helpful suggestions.

References

- [AB98] AMENTA N., BERN M. W.: Surface Reconstruction by Voronoi Filtering. In *Symp. Comp. Geom.* (1998), pp. 39–48. [2](#), [10](#)
- [ABR06] ALBOUL L., BRINK W., RODRIGUES M.: Mesh optimisation based on Willmore energy. In *Euro. Workshop on Comp. Geom.* (2006), pp. 133–136. [2](#)
- [ACDL00] AMENTA N., CHOI S., DEY T. K., LEEKHA N.: A Simple Algorithm for Homeomorphic Surface Reconstruction. In *Symp. Comp. Geom.* (2000), pp. 213–222. [2](#)
- [BGO05] BOISSONNAT J.-D., GUIBAS L. J., OUDOT S.: Learning smooth objects by probing. In *Symp. Comp. Geom.* (2005), pp. 198–207. [2](#)
- [BO03] BOISSONNAT J.-D., OUDOT S.: Provably Good Surface Sampling and Approximation. In *SGP* (2003), pp. 9–18. [2](#)
- [Bre42] BRETSCHNEIDER C. A.: Untersuchung der trigonometrischen Relationen des geradlinigen Vierecks. *Archiv der Math.* [2](#) (1842), 225–261. [5](#)
- [BS05] BOBENKO A. I., SPRINGBORN B. A.: A discrete Laplace-Beltrami operator for simplicial surfaces. arXiv:math.DG/0503219 v1, 2005. [1](#), [2](#), [4](#), [5](#), [6](#)
- [CB97] CHEN H., BISHOP J.: Delaunay triangulation for curved surfaces. In *Meshing Roundtable* (1997), pp. 115–127. [2](#)
- [CDRR04] CHENG S.-W., DEY T. K., RAMOS E. A., RAY T.: Sampling and meshing a surface with guaranteed topology and geometry. In *Symp. Comp. Geom.* (2004), pp. 280–289. [2](#)
- [Che89] CHEW L. P.: Constrained Delaunay triangulations. *Algorithmica* [4](#), 1 (1989), 97–108. [2](#)
- [Che93] CHEW L. P.: Guaranteed-quality mesh generation for curved surfaces. In *Symp. Comp. Geom.* (1993), pp. 274–280. [2](#)
- [CRS98] CIGNONI P., ROCCHINI C., SCOPIGNO R.: Metro: Measuring error on simplified surfaces. *Computer Graphics Forum* [17](#), 2 (1998), 167–174. [8](#)
- [dBvKOS98] DE BERG M., VAN KREVELD M., OVERMARS M., SCHWARZKOPF O.: *Computational Geometry. Algorithms and Applications*. Springer-Verlag, 1998. [1](#), [5](#)
- [DHKL01] DYN N., HORMANN K., KIM S.-J., LEVIN D.: Optimizing 3D triangulations using discrete curvature analysis. In *Mathematical Methods for Curves and Surfaces*. Vanderbilt University, 2001, pp. 135–146. [2](#)
- [DLR05] DEY T. K., LI G., RAY T.: Polygonal surface remeshing with Delaunay refinement. In *Meshing Roundtable* (2005), pp. 343–361. [2](#)
- [DLYG06] DAI J., LUO W., YAU S.-T., GU X.: Geometric accuracy analysis for discrete surface approximation. In *Geometric Modelling and Processing* (2006), pp. 59–72. [10](#)
- [DZM07] DYER R., ZHANG H., MÖLLER T.: Voronoi-Delaunay duality and Delaunay meshes. In *Symp. Solid and Physical Modelling* (2007). to appear. [1](#), [2](#), [6](#)
- [Ede01] EDELSBRUNNER H.: *Geometry and Topology for Mesh Generation*. Cambridge, 2001. [1](#), [2](#)
- [ES94] EDELSBRUNNER H., SHAH N. R.: Triangulating topological spaces. In *Symp. Comp. Geom.* (1994), pp. 285–292. [2](#)
- [ET93] EDELSBRUNNER H., TAN T. S.: An upper bound for conforming Delaunay triangulations. *Discrete and Computational Geometry* [10](#), 2 (1993), 197–213. [3](#)
- [Flo98] FLOATER M. S.: Parametric tilings and scattered data approximation. *Int. J. of Shape Modeling* [4](#) (1998), 165–182. [1](#)
- [FSBS06] FISHER M., SPRINGBORN B., BOBENKO A. I., SCHRÖDER P.: An algorithm for the construction of intrinsic Delaunay triangulations with applications to digital geometry processing. In *SIGGRAPH Courses* (2006), pp. 69–74. [1](#), [4](#)
- [GH97] GARLAND M., HECKBERT P. S.: Surface simplification using quadric error metrics. In *ACM SIGGRAPH* (1997), pp. 209–216. [2](#), [6](#), [7](#)
- [GW03] GERTZ E. M., WRIGHT S. J.: Object-oriented software for quadratic programming. *ACM Trans. on Math. Software* [29](#) (2003), 58–81. www.cs.wisc.edu/~swright/ooqp/. [2](#), [7](#)
- [HPW06] HILDEBRANDT K., POLTHIER K., WARDETSKY M.: On the Convergence of Metric and Geometric Properties of Polyhedral Surfaces. *Geometriae Dedicata* [123](#), 1 (2006), 89–112. [9](#)
- [Law77] LAWSON C. L.: Software for C^1 surface interpolation. In *Math. Software III*, Rice J. R., (Ed.). Academic Press, New York, 1977, pp. 161–194. [1](#), [2](#), [3](#), [6](#)
- [LL00] LEIBON G., LETSCHER D.: Delaunay Triangulations and Voronoi Diagrams for Riemannian Manifolds. In *Symp. Comp. Geom.* (2000), pp. 341–349. [1](#), [6](#)
- [LZ06] LI J., ZHANG H.: Nonobtuse remeshing and decimation. In *SGP* (2006), pp. 235–238. [2](#), [6](#), [7](#)
- [Mus97] MUSIN O. R.: Properties of the Delaunay triangulation. In *Symp. Comp. Geom.* (1997), pp. 424–426. [2](#), [4](#)
- [O’R81] O’ROURKE J.: Polyhedra of minimal area as 3d object models. In *Intl. Joint Conf. on AI* (1981), pp. 664–666. [2](#)
- [PC03] PEYRÉ G., COHEN L.: Geodesic Remeshing Using Front Propagation. In *Proceedings VLISM* (2003), pp. 33–40. [2](#), [8](#)
- [RI97] RIVARA M.-C., INOSTROZA P.: Using longest-side bisection techniques for the automatic refinement of Delaunay triangulations. *Int. J. Num. Meth. in Eng.* [40](#) (1997), 581–597. [2](#)
- [Rup95] RUPPERT J.: A Delaunay refinement algorithm for quality 2-dimensional mesh generation. *J. of Algorithms* [18](#), 3 (1995), 548–585. [2](#), [3](#)
- [She97] SHEWCHUK J. R.: *Delaunay Refinement Mesh Generation*. PhD thesis, School of Computer Science, Carnegie Mellon University, 1997. Technical Report CMU-CS-97-137. [1](#), [2](#)