

Weighted Delaunay Refinement for Polyhedra with Small Angles*

Siu-Wing Cheng[†] Tamal K. Dey[‡] Tathagata Ray[‡]

September 20, 2005

Abstract

Recently, a provable Delaunay meshing algorithm called QMESH has been proposed for polyhedra that may have acute input angles. The algorithm guarantees bounded circumradius to shortest edge length ratio for all tetrahedra except the ones near small input angles. This guarantee eliminates or limits the occurrences of all types of poorly shaped tetrahedra except slivers. A separate technique called weight pumping is known for sliver elimination. But, allowable input for the technique so far have been periodic point sets and piecewise linear complex with non-acute input angles. In this paper, we incorporate the weight pumping method into QMESH thereby ensuring that all tetrahedra except the ones near small input angles have bounded aspect ratio. Theoretically, the algorithm has an abysmally small angle guarantee inherited from the weight pumping method. Nevertheless, our experiments show that it produces better angles in practice.

1 Introduction

Meshing a polyhedral domain with well shaped tetrahedra occurs as an important problem in finite element methods. The *aspect ratio* of a tetrahedron is the ratio of its circumradius to its inradius. A tetrahedron has bounded aspect ratio if all the face angles and dihedral angles are greater than a constant threshold. A weaker measure is the *circumradius-edge ratio*: ratio of the circumradius to the shortest edge length. If the tetrahedra have bounded circumradius-edge ratio, only one class of poorly shaped tetrahedra may remain and they are known as *slivers* [7]. Eliminating slivers is a challenge in mesh generation.

Three main paradigms are known for polyhedra meshing: *octree based* [18, 23], *advancing front* [16, 17], and *Delaunay based* [1, 2, 4, 27] methods. The Delaunay based methods are popular in practice [20], perhaps due to their directional independence and good quality meshing in general. In this paper, we focus on the Delaunay based methods that can provide theoretical guarantees on the quality of tetrahedra.

Based on Ruppert's 2D Delaunay refinement paradigm [22], Shewchuk presented an algorithm that constructs a Delaunay mesh with bounded circumradius-edge ratio [24]. Unfortunately,

*Research is supported by RGC, Hong Kong, China, under grants HKUST6190/02E, Army Research Office, USA under grant DAAD19-02-1-0347 and NSF, USA under grants DMS-0310642 and CCR-0430735.

[†]Department of Computer Science, HKUST, Hong Kong. Email: scheng@cs.ust.hk

[‡]Department of Computer Science and Engineering, Ohio State University, OH, USA. Email: tamaldey,rayt@cse.ohio-state.edu

the algorithm may not recover the input boundary when some input angle is acute. Although Shewchuk [25], Murphy, Mount and Gable [19], and Cohen-Steiner, de Verdière and Yvinec [11] subsequently proposed algorithms that can handle acute input angles, no guarantee on the tetrahedral shape was provided. A significant theoretical progress was obtained by Cheng and Poon [9], who proposed an algorithm that guarantees bounded circumradius-edge ratio everywhere in the mesh. However, given the complexity of the algorithm, its practicality is doubtful. Recently, Cheng, Dey, Ramos, and Ray [8] presented a simple algorithm QMESH and an implementation for polyhedra. All tetrahedra have bounded circumradius-edge ratio, except those near small input angles. There are very few such tetrahedra as observed in the experiments. There are several innovations in this algorithm: local feature sizes are only needed at vertices with small input angles, explicit protecting regions for the input edges are no longer needed, and the splitting of non-Delaunay triangles in recovering the input boundary (instead of splitting non-Gabriel triangles alone). These new features ease the implementation tremendously and help to keep the mesh size small. Subsequently Pav and Walkington [21] proposed an algorithm for handling non-manifold boundaries with very similar guarantees.

All of the algorithms mentioned so far focused on ensuring bounded circumradius-edge ratio, which eliminates all types of poorly shaped tetrahedra but slivers. The work of Chew [10], Cheng et al. [7], and Edelsbrunner et al. [13] are the first theoretical results on sliver elimination, albeit for point sets only. Edelsbrunner and Guoy [12] experimented with the sliver exudation technique in [7]. They demonstrated that the technique is more effective in practice than what the theory predicts. Later, Li and Teng [15], and Cheng and Dey [6] presented algorithms for meshing piecewise linear complex with well-shaped tetrahedra. But only non-acute input angles were allowed.

In this paper, we incorporate the weight pumping technique [6, 7] into QMESH [8] to purge slivers. In essence, we make the following contributions in this paper.

Theoretical: We present a Delaunay meshing algorithm for polyhedra possibly with acute input angles. It guarantees bounded aspect ratio for all tetrahedra except the ones near small input angles. No Delaunay meshing algorithm with such guarantees is known to date. The new algorithm changes the mesh connectivity using the weighted Delaunay triangulation. Since weight pumping may challenge the input boundary, we first extend the algorithm in [8] to prepare for it. The extended algorithm offers the same guarantee that remaining skinny tetrahedra are near small input angles. Then the algorithm assigns proper weights to sliver vertices to purge them. Most slivers are eliminated except those near small input angles.

Practical: The weight pumping procedure of Cheng and Dey [6] needs the local feature sizes at the sliver vertices which are expensive to compute. We improve upon this by showing that it suffices to work with the nearest neighbor distances of the sliver vertices. This is readily available in a Delaunay mesh and eases the weight pumping implementation substantially.

Experimental: We experimented with an implementation of the proposed algorithm. In our experiments, we set the circumradius-edge ratio threshold at 2.2 and dihedral angle bound for slivers at 3° . This means that the algorithm tries to eliminate any poorly shaped tetrahedron with circumradius-edge ratio more than 2.2 and any sliver with dihedral angle less than 3° . Our experiments show that there are extremely few tetrahedra in the output that violate these thresholds. The plot of the distribution of the face and dihedral angles shows that over 90% of the angles are more than 10° and many of them are between 15° and 30° .

2 Input domain

The input domain is a polyhedron bounded by a 2-manifold. The 2-manifold is the underlying space of a piecewise-linear-complex called PLC defined as follows. A vertex v is a point in \mathbb{R}^3 ; its boundary ∂v is v itself. An edge e is a closed segment between two vertices v_1 and v_2 where its boundary is $\partial e = \{v_1, v_2\}$. A facet is a subset of the plane bounded by a collection simple polygonal cycles made out of vertices and edges. A PLC is a collection of vertices, edges, and facets that intersect properly, i.e., the intersection of any two elements is either empty or a collection of lower-dimensional elements. Although this definition disallows special cases such as isolated vertex or a dangling edge in the middle of a facet, we believe that, after adding extra steps to deal with special cases, our algorithm works for any input PLC as long as its underlying space remains a 2-manifold.

Also, we make a modification to the input domain by encompassing the original input polyhedron with a large enough bounding box. Our algorithm meshes the interior of the box conforming to the input polyhedron and keeps only the tetrahedra covering the interior of the input polyhedron. Let \mathcal{B} denote the bounding box. We use \mathcal{P} to denote the PLC of the input polyhedron together with \mathcal{B} . Two elements in \mathcal{P} are *incident* if one is in the boundary of the other. We call two elements of \mathcal{P} *adjacent* if they intersect.

Our algorithm works with two types of input angles. For any two incident edges of a vertex u , we measure the angle between them. We call such angles *edge-edge angles*. For any edge uv and a facet F incident to u such that uv is neither incident on F nor coplanar with F , the angle between uv and F is $\min\{\angle puv : p \in F, p \neq u\}$. We call such angles *edge-facet angles*. At an edge of \mathcal{P} , we measure the internal and external dihedral angles at the edge. Throughout this paper, we use ϕ_m to denote the minimum input angle in \mathcal{P} . We call an edge *sharp* if the internal or external dihedral angle at the edge is acute. We call a vertex u *sharp* if an edge-edge angle or an edge-facet angle at u is acute, or if u is an endpoint of a sharp edge.

The *local feature size* $f(x)$ for \mathcal{P} at $x \in \mathbb{R}^3$ is the radius of the smallest ball centered at x intersecting two non-adjacent elements of \mathcal{P} .

3 Weighted Delaunay and shape measure

We denote a *weighted point* at location x by \hat{x} and we use X^2 to denote its weight. The *weighted distance* between \hat{x} and \hat{y} is $\pi(\hat{x}, \hat{y}) = \|x - y\|^2 - X^2 - Y^2$. We say that \hat{x} and \hat{y} are *orthogonal* if $\pi(\hat{x}, \hat{y}) = 0$. One can view \hat{x} as a sphere centered at x with radius X . Then \hat{x} and \hat{y} are orthogonal if the two spheres intersect at right angle. An *orthosphere* of k , $2 \leq k \leq 4$, weighted points $\hat{x}_1, \dots, \hat{x}_k$ is a sphere \hat{y} such that $\pi(\hat{x}_i, \hat{y}) = 0$ for $1 \leq i \leq k$. The orthosphere is unique for four weighted points in general position. The center and radius of the orthosphere are known as *orthocenter* and *orthoradius*.

Given a weighted point set, a tetrahedron spanning four points is *weighted Delaunay* if the weighted distance between its orthosphere and any other weighted point is non-negative. The *weighted Delaunay triangulation* is the collection of all weighted Delaunay tetrahedra along with their triangles, edges and vertices. It can be built using an incremental algorithm [14] and CGAL has a library function for its construction [5]. We will be interested in the special case where the

weights are zero or small. Specifically, the weight of \hat{v} can take on a value between 0 and $\omega_0 N(v)$, where $\omega_0 \in [0, 1/3)$ is a constant to be defined later and $N(v)$ is the nearest neighbor distance of v . In this case, we say that the point set has *weight property* $[\omega_0]$. If all weights are zero for a vertex set \mathcal{V} , we get the standard unweighted Delaunay triangulation $\text{Del } \mathcal{V}$.

The volume of a tetrahedron τ in a normalized sense captures its shape quality. We define the *orthoradius-edge ratio* $\rho(\tau) = R/L$ and the *normalized volume* $\sigma(\tau) = \text{vol}(\tau)/L^3$, where R and L are the orthoradius and the shortest edge length of τ , respectively. These two values determine the shape of τ as indicated in the following lemma. In other words, the quality measure in the weighted mesh is sufficient for actual quality, if the weight property $[\omega_0]$ holds for sufficiently small ω_0 .

Lemma 3.1 *Assume that vertices of τ have weight property $[\omega_0]$. If $\rho(\tau) \leq \rho_1$ and $\sigma(\tau) > \sigma_0$ for some constants ρ_1 and σ_0 , all angles of τ are greater than some constant threshold.*

Proof. Let pq be the longest edge of τ . Let \hat{z} be the orthosphere of τ . Since both \hat{p} and \hat{q} overlap with \hat{z} , $Z + P \geq \|p - z\|$ and $Z + Q \geq \|q - z\|$. The weight property implies that $\max\{P, Q\} \leq \omega_0 \cdot \|p - q\|$. Thus, $2Z \geq \|p - z\| + \|q - z\| - 2\omega_0 \cdot \|p - q\| \geq (1 - 2\omega_0) \cdot \|p - q\|$. Then $\rho(\tau) \leq \rho_1$ implies that $L \geq Z/\rho_1 \geq \frac{1-2\omega_0}{2\rho_1} \cdot \|p - q\|$. That is, the length of any two edges of τ are within a constant factor. It follows that the area of any face of τ is $O(L^2)$. So any height of τ is at least $\sigma_0 L^3 / O(L^2) = \Omega(L)$. The sine of any dihedral angle of τ is at least the ratio of a height to the longest edge length. This implies that all dihedral angles of τ are greater than some constant. So are the angles of the triangles of τ . \square

4 Algorithm

The algorithm has four distinct phases, INITIALIZE, CONFORM, REFINER, and PUMP. The phases INITIALIZE, CONFORM, and REFINER are from [8]. Some modifications are incorporated into REFINER to prepare for PUMP. The result of the first three phases is a Delaunay mesh with bounded circumradius-edge ratio, except near the small input angles. In PUMP, we assign weights to sliver vertices to remove the slivers.

The algorithm maintains a vertex set \mathcal{V} . The edges of \mathcal{P} are divided into *subsegments* by inserted vertices. The subsegments inside the vertex balls are protected by the vertex balls, so we are not concerned about them. We call a subsegment *sharp* if it lies outside the vertex balls and on a sharp edge. It is *non-sharp* otherwise. A *circumball* of a subsegment is a finite ball with the subsegment endpoints on its boundary. The *diametric ball* of a subsegment is its smallest circumball. A point p *encroaches* a subsegment e if p is not an endpoint of e and lies on or inside its diametric ball. This encroachment definition is stricter than usual. When p is weighted, we say \hat{p} encroaches e if $\pi(\hat{p}, \hat{z}) \leq 0$ where \hat{z} is the diametric ball of e .

When no subsegment is encroached by any vertex in \mathcal{V} , the facets are decomposed into *subfacets* defined as follows. For each facet F of \mathcal{P} , consider the 2D Delaunay triangulation of the vertices in $\mathcal{V} \cap F$. The Delaunay triangles in F are its subfacets. Thanks to the stricter definition of subsegment encroachment, the circumcenters of subfacets lie strictly in the interior of F [6]. A *circumball* of a subfacet h is a finite ball with the vertices of h on its boundary. The *diametric ball* of h is its

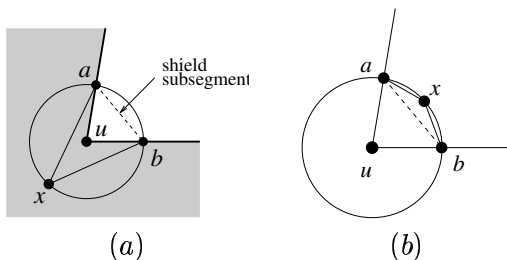


Figure 1: Shield subsegment and SOS splitting.

smallest circumball. A point p *encroaches* h if p lies inside its diametric ball. When p is weighted, we say \hat{p} encroaches h if $\pi(\hat{p}, \hat{z}) < 0$ where \hat{z} is the diametric ball of h .

There are several subroutines used by our algorithm. We describe them in Sections 4.1 and 4.2 and then give the algorithm in Section 4.3. Some of these subroutines are exactly same as those described in [8]. We include them here for completeness.

4.1 Sharp vertex protection

We protect each sharp vertex u with a *vertex ball* centered at u with radius $f(u)/4$. The distance of u from each non-adjacent input vertex, edge and facet is computed to determine $f(u)$. We denote the protecting ball of u by \hat{u} . The points where the boundary of \hat{u} intersects edges of \mathcal{P} are inserted into the vertex set \mathcal{V} . We protect a subset of $\hat{u} \cap \mathcal{P}$ using the SOS method of Cohen-Steiner, de Verdière and Yvinec [11] (see [3] for a generalized variant of this operation). At any generic step of the algorithm, \mathcal{V} contains vertices on the arc where a facet F incident to u intersects the boundary of \hat{u} . The segments connecting consecutive points on such an arc form *shield subsegments*. Let ab be a shield subsegment. If the angle of the sector aub on F is at least π , we insert the midpoint x on the arc between a and b on the boundary of $\hat{u} \cap F$. The subsegment ab is replaced with two shield subsegments ax and bx ; see Figure 1(a). If the angle of the sector aub is less than π and ab is encroached, we also insert x to split ab ; see Figure 1(b).

When no shield subsegment corresponds to a sector at u with angle π or more, the shield subsegments around u create a set of *shield subfacets* incident to u . It turns out that the diametric ball of a shield subfacet lies in the union of the vertex ball \hat{u} and the diametric ball of the corresponding shield subsegment. Since \hat{u} is kept empty throughout the algorithm, it is sufficient to keep the shield subsegments non-encroached to ensure that shield subfacets appear in $\text{Del } \mathcal{V}$.

In INITIALIZE, we only insert the points where the incident edges of u intersect the boundary of \hat{u} , and split shield subsegments that correspond to sectors with angles π or more. The encroachment of shield subsegments is handled in the next phase CONFORM.

4.2 Edge, Facet, Ball and Tetra Splitting

Edges are split in both the CONFORM and REFINE phases to recover the edges of \mathcal{P} as union of Delaunay edges. Any subsegment (sharp, non-sharp or shield) that is encroached is split using

SPLITE until no such segment exists.

SPLITE(e)

If e is a shield subsegment, split it with SOS else insert the midpoint of e and update $\text{Del } \mathcal{V}$.

After we recover all the edges, we start splitting facets so that they appear in $\text{Del } \mathcal{V}$ as union of subfacets. Standard Delaunay refinement insists that no encroached subfacet exists. While such a condition can be enforced for subfacets on the boundary of the bounding box \mathcal{B} , it may never be satisfied for all subfacets. Instead, we check only that if any subfacet does not appear in $\text{Del } \mathcal{V}$. We argue that, for a polyhedron, all subfacets must appear in $\text{Del } \mathcal{V}$ after sufficient but finite amount of splitting.

A subfacet h that does not appear in $\text{Del } \mathcal{V}$ is split using the procedure SPLITF. Certainly, h cannot be a shield subfacet since there is no encroached shield subsegment when the algorithm reaches the facet splitting step.

SPLITF(h)

(i) Compute the circumcenter c of h ;

(ii) Let F be the facet containing h . If c does not encroach any subsegment, insert c and update $\text{Del } \mathcal{V}$. Otherwise, reject c and

1. pick a subsegment g encroached by c with preference for those in ∂F or on F (shield subsegment), and
2. call SPLITF(g).

In the CONFORM phase, after all the edges and facets are recovered, further splittings of subsegments and subfacets may be performed to reduce the diametric balls of the sharp subsegments roughly to the order of local feature sizes. In order to avoid the computation of local feature sizes, this is achieved in a roundabout way (see rule 3 in QMESH).

At the end of the CONFORM phase, for each sharp subsegment, we double the radius of its diametric ball with the center fixed and call this a *protecting ball*. These protecting balls and the vertex balls at the sharp vertices constitute the entire set of *protecting balls* for the REFINE phase.

Skinny tetrahedra are split in the REFINE phase by inserting their circumcenters. But we disallow the insertion of the circumcenters of skinny tetrahedra inside any protecting ball. The reason is that once these points are allowed to be inserted, they can cause perpetual splittings of the subsegments or subfacets. This means that some skinny tetrahedra may remain at the end. It has been proved that all such tetrahedra lie close to sharp vertices or edges [8]. We also prepare for pumping in the REFINE phase. For this we assign a weight to each vertex of a sliver tetrahedron and check if the weighted vertex threatens to destroy the input conformity. If so, we split subsegments and subfacets further. Potentially we can choose to pump all vertices in PUMP and thus can prepare them all in REFINE. But, for efficiency we choose only a subset of tetrahedra that have a small dihedral angle, say 3° .

4.3 WQMESH

The following algorithm WQMESH triangulates \mathcal{P} with the claimed theoretical guarantees. WQMESH uses the following subroutine ENCROACH to test whether a point c can be inserted and if not, return the appropriate point to be inserted.

ENCROACH(c)

1. If c does not encroach any subsegment or subfacet, return c .
2. If c encroaches some subsegment e , reject c and if e is a shield subsegment, return the point given by SOS; else return the midpoint of e .
3. If c encroaches some subfacet, one such subfacet h contains the orthogonal projection of c . Reject c . If the circumcenter p of h does not encroach any subsegment, return p ; otherwise, reject p and return the point as in case 2.

The following are the descriptions of the four phases of WQMESH(\mathcal{P}). Recall that \mathcal{P} includes a bounding box \mathcal{B} which encompasses the original polyhedron.

INITIALIZE. Initialize \mathcal{V} to be the set of vertices of \mathcal{P} . Compute the vertex balls. Insert the intersections between their boundaries and the edges of \mathcal{P} into \mathcal{V} . If any shield subsegment forms a sector with angle π or more, split it with SOS. Compute $\text{Del } \mathcal{V}$.

CONFORM. Repeatedly apply a rule from the following list until no rule is applicable. Rule i is applied if it is applicable and no rule j with $j < i$ is applicable.

- Rule 1. If there is an encroached subsegment e , call SPLITTE(e).
- Rule 2. If there is a subfacet $h \subset \mathcal{B}$ that is encroached, or if $h \not\subset \mathcal{B}$ and h does not appear in $\text{Del } \mathcal{V}$, call SPLITF(h).
- Rule 3. Let s be a sharp subsegment on an edge e . If the midpoint of s encroaches a subsegment or subfacet h , where h and e are contained in disjoint elements of \mathcal{P} , split h accordingly using SPLITTE(h) or SPLITF(h).

At the end of CONFORM, we double the sizes of the diametric balls of sharp subsegments. These expanded balls and the vertex balls are the *protecting balls*. The sharp subsegments may be split further in the next phase, but the locations and sizes of these protecting balls do not change.

We call a subfacet *guarded* if its diametric ball lies inside some protecting ball.

REFINE. Repeatedly apply a rule from the following list until no rule is applicable. Rule i is applied if it is applicable and no rule j with $j < i$ is applicable. The parameter $\rho_0 > 2/(1 - \tan(\pi/8))$ is a constant chosen a priori.

- Rule 4. If there is an encroached subsegment e , call SPLITTE(e).
- Rule 5. If there is a non-guarded subfacet h that is encroached or a guarded subfacet h that does not appear in $\text{Del } \mathcal{V}$, then call SPLITF(h).

Rule 6. Assume that there is a tetrahedron with circumradius-edge ratio exceeding ρ_0 . Let z be its circumcenter. If z does not lie on or inside any protecting ball, then compute $p := \text{ENCROACH}(z)$ and insert p into \mathcal{V} .

Rule 7. Take a vertex v of a tetrahedron τ with a dihedral angle of 3° or less. Let \hat{v} be the weighted vertex v with weight $L_v^2/9$ where L_v is the length of the shortest incident edge of v . We ignore v if \hat{v} encroaches the two times expansion of some protecting ball. Otherwise, we take action only in the following cases: (i) if \hat{v} encroaches some subsegment, we split it by `SPLITE`; (ii) if \hat{v} encroaches some non-guarded subfacet, we split by `SPLITF` the one that contains the projection of v ; (iii) if \hat{v} makes some guarded subfacet disappear from the 3D weighted Delaunay triangulation, we split the subfacet by `SPLITF`.

We always maintain an unweighted Delaunay triangulation in `REFINE` and \hat{v} is used only for checking encroachments. Notice that, as claimed in the introduction, the weight pumping depends on nearest neighbor distances as opposed to local feature sizes in [6].

PUMP. We examine all vertices v incident on a tetrahedron with a dihedral angle of 3° or less such that \hat{v} does not encroach the two times expansion of any protecting ball. For each such vertex v , we assign to v the weight in the interval $[0, L_v^2/9]$ that maximizes the minimum dihedral angle of the tetrahedra incident to v . We maintain the weighted Delaunay triangulation during the weight pumping. We claim that no pumped vertex encroaches upon any weighted-subsegment and weighted-subfacet.

Note: We used 3° as a threshold on the dihedral angles of tetrahedra to select vertices to pump. The choice of 3° is dictated by our experiments which shows that in most cases pumping eliminates tetrahedra with dihedral angles smaller than 3° . In theory, we show that pumping eliminates all tetrahedra with dihedral angle below a positive threshold. Although this threshold is much lower than 3° , the algorithm as stated works. Only that, sometimes tetrahedra with dihedral angles between the theoretical threshold and 3° may not get eliminated.

Let \mathcal{M} denote the unweighted Delaunay mesh at the end of `REFINE`. Let U denote the set of vertices of tetrahedra in \mathcal{M} with circumradius-edge ratio greater than ρ_0 . It has been proved that all vertices in U are close to a small input angle [8]. Our main result is that all sliver vertices left after `PUMP` also satisfies similar property. We say that a vertex v is in the E -neighborhood of U if v is within E edges in \mathcal{M} from some vertex of U . We say that τ is in the E -neighborhood of U if some vertex of τ is in the E -neighborhood of U . Recall that $\rho(\tau)$ and $\sigma(\tau)$ denote the orthoradius-edge ratio and the normalized volume of τ , respectively. The main theorem we prove is:

Theorem 4.1 *There are constants $\rho_1, \sigma_0, E > 0$ such that for every tetrahedron τ in the output mesh of `WQMESH`, $\rho(\tau) \leq \rho_1$ and $\sigma(\tau) > \sigma_0$ unless τ is in the E -neighborhood of U or τ is within distance $O(f(x))$ from some point x on some sharp edge.*

5 Analysis

In this section, we first prove that the first three phases terminate with a graded Delaunay mesh and all tetrahedra have bounded circumradius-edge ratio, except those near small input angles. Then we show that the phase PUMP eliminates slivers, except those that are near skinny tetrahedra or the protecting balls.

The main purpose of the bounding box \mathcal{B} is to disallow any point to be inserted outside \mathcal{B} so that one can claim termination by applying a packing argument within a bounded domain. The set of points, say P , that are inserted or rejected while conforming to \mathcal{B} maintain a lower bound on their distances to all other existing points by the arguments in [6] because all angles of \mathcal{B} are $\pi/2$. In the analysis we skip explicit arguments about P and focus on the set of points, say Q , that are inserted or rejected for conforming to the edges and facets of the original polyhedron. Of course, we do not lose any generality by doing so as the lower bounds on distances for P are dominated by those for Q .

5.1 Termination and conformity

In this section, we inductively prove that the inter-vertex distances remain above certain thresholds in the first three phases. The inductive argument makes use of a predecessor relation defined as follows. Let x be a vertex inserted or rejected by WQMESH. The *predecessor* of x is an input vertex or a vertex inserted or rejected by WQMESH. If x is a vertex of \mathcal{P} or a vertex inserted during INITIALIZE, its predecessor is undefined. Otherwise, the predecessor p is defined as follows.

- Suppose that x splits a subsegment (shield or non-shield) or a subfacet h . Let B be the diametric ball of h .
If B contains some vertex in \mathcal{V} , p is the encroaching vertex nearest to x . If B is empty and WQMESH is going to reject a vertex inside B for encroaching h , p is that vertex. Otherwise, B is empty and the encroachment must be due to a weighted vertex outside the ball in rule 7. Then p is that weighted vertex.
- If x is the circumcenter of a skinny tetrahedron τ , then p is one of the endpoints of the shortest edge of τ . Between the two endpoints of the shortest edge, p is the one that appears in \mathcal{V} later.

It is still possible that x has no predecessor. This happens when x is inserted to split a subsegment in rule 3. When a subfacet is split by x in rule 3, x may or may not have a predecessor depending on whether the diametric ball is empty or not.

The *neighbor radius* r_x of x is the distance from x to its nearest neighbor in the current \mathcal{V} when x is inserted or rejected. So if x is an input vertex, then $r_x \geq f(x)$. The following result shows that $r_x = \Omega(f(x))$ and that the protecting balls are not too small at the end of CONFORM.

Lemma 5.1 [8] *At the end of CONFORM, there are constants $\mu_1, \mu_2 > 0$ such that*

- (i) *For each vertex $x \in \mathcal{V}$, $r_x \geq \mu_1 f(x)$.*

- (ii) *For any point z on a sharp subsegment, a ball centered at z with radius $\mu_2 f(z)$ lies inside some protecting ball.*

So we focus on analyzing REFINE. We will show by induction that the rules 4–7 in REFINE preserve the following property. Let $\omega_0 \in [0, 1/3)$ be a constant to be defined later.

VERTEX GAP PROPERTY: For each vertex $p \in \mathcal{V}$, the nearest neighbor distance of p is at least $\omega_0 f(p)$ throughout REFINE.

We need the following result, Lemma 5.2, about REFINE. It is the same as Lemma 4.16 in [8](extended version). The original proof almost works here but some change is needed due to our extension of REFINE to prepare for PUMP. The only case that is not handled in the original proof is that the weighted encroachment in rule 7 may split a guarded subfacet h , although h belongs to the 3D unweighted Delaunay triangulation. If we can argue that the circumradius of h is $\Omega(f(x))$ where x is its circumcenter, the original argument goes through. We can invoke Lemma 4.10 in [8] to show that the circumradius of h is $\Omega(f(x))$, provided that we can find a circumball B of h such that B contains a vertex u on some element of \mathcal{P} , u and the center z of B lie on different sides of the plane of h , and $\text{radius}(B) \geq cf(x) - \|x - z\|$ for some constant $c > 0$. Since some weighted vertex \hat{v} makes h disappear from the 3D weighted Delaunay triangulation in rule 7 and \hat{v} is relatively far from any protecting ball, it can be checked that such a ball B exists. Thus Lemma 5.2 holds.

Lemma 5.2 *Assume that no subsegment is encroached by a vertex. Let x be the circumcenter of a subfacet inserted or rejected during REFINE and p be its predecessor. If p is a vertex in \mathcal{V} lying on some element of \mathcal{P} , then $r_x = \Omega(f(x))$.*

Lemma 5.3 *Let x be a vertex inserted or rejected during REFINE. Assume that x splits a subfacet h and its predecessor p is defined. If p does not encroach h , then $\|p - x\| \geq \min\{\mu_3 f(x), \mu_4 r_p\}$ for some constants $\mu_3, \mu_4 > 0$.*

Proof. Since p does not encroach h , the diametric ball of h centered at x is empty. Moreover, the encroachment happens in rule 7 when p is pumped. So $p \in \mathcal{V}$. By rule 7, p lies outside all protecting balls and h is a non-guarded subfacet. Go back to the time t when p was first inserted into \mathcal{V} . Let F be the input facet containing h .

Let abc be the subfacet on F that contains the projection of p at that time. If p lies outside the diametric ball of abc , it can be shown that the distance from p to F is at least $r_p/\sqrt{2}$ [6]. So is $\|p - x\|$. If p lies inside the diametric ball of abc , then p lies on some input element F' . Otherwise, p would be the circumcenter of a tetrahedron and be rejected by the algorithm. If F' is disjoint from F , then $\|p - x\| \geq f(x)$. Otherwise, F and F' meet at a sharp vertex or a sharp edge. Let z be the point in $F \cap F'$ closest to p . Since p lies outside all protecting balls, Lemma 5.1(ii) implies that $\|p - z\| \geq \mu_2 f(z)$. The Lipschitz condition implies that $\|p - z\| \geq \mu_2 f(p)/(1 + \mu_2)$. Thus, $\|p - x\| \geq \|p - z\| \sin \phi_m \geq \mu_2 \sin \phi_m f(p)/(1 + \mu_2)$. Invoking the Lipschitz condition again yields $\|p - x\| \geq \mu_2 \sin \phi_m f(x)/(1 + \mu_2 + \mu_2 \sin \phi_m)$. \square

Now we can show that the splitting vertex x triggered by the weighted encroachment in rule 7 has neighbor radius $\Omega(f(x))$, if the predecessor of x is outside the diametric ball centered at x .

Lemma 5.4 *Suppose that the vertex gap property is satisfied. Let x be a vertex that splits a subfacet h . Assume that its predecessor p is defined. If p lies outside the diametric ball of h , then $r_x \geq \mu_5 f(x)$ for some constant $\mu_5 > 0$.*

Proof. Since p is the predecessor of x but p does not encroach h , the encroachment occurs in rule 7 when p is pumped. Let L_p be the nearest neighbor distance of p at that time.

We claim that $L_p \leq 2\|p - x\|$. If the claim is false, the ball B centered at p with radius L_p contains x . Moreover, the distance from x to the boundary of B is greater than $L_p - \|p - x\| \geq L_p/2$. By the definition of L_p , B is empty. So the vertices of h lies outside B , which implies that the diametric ball of h (centered at x) has radius at least $L_p/2 > \|p - x\|$. But then p must encroach h , contradicting our assumption. This proves the claim.

By the weighted encroachment, $r_x^2 + L_p^2/9 \geq \|p - x\|^2$. So our claim implies that $r_x \geq \frac{\sqrt{5}}{3}\|p - x\|$. By Lemma 5.3, $\|p - x\| \geq \min\{\mu_3 f(x), \mu_4 r_p\}$. If $\|p - x\| \geq \mu_3 f(x)$, then $r_x \geq \frac{\sqrt{5}\mu_3}{3} f(x)$. If $\|p - x\| \geq \mu_4 r_p$, then $\|p - x\| \geq \mu_4 L_p \geq \mu_4 \omega_0 f(p)$ by the vertex gap property. Thus, $f(x) \leq f(p) + \|p - x\| = O(\|p - x\|) = O(r_x)$. \square

We are ready to apply induction to prove the vertex gap property holds throughout REFINES and to bound the inter-vertex distances from below by the local feature sizes.

Lemma 5.5 *Let x be a vertex such that x exists in \mathcal{V} before the invocation of REFINES or x is inserted or rejected by WQMESH during REFINES. The following invariants hold.*

- (i) $r_x \geq f(x)/C$ for some constant $C > 0$.
- (ii) For any other vertex y currently in \mathcal{V} , $\|x - y\| \geq \max\{f(x)/C, f(y)/(1 + C)\}$.
- (iii) Assume that $\omega_0 = 1/(1 + C)$. If x is inserted, the vertex gap property holds afterwards.

Proof. By Lemma 5.1(i), if x exists in \mathcal{V} before the invocation of REFINES, then $r_x \geq \mu_1 f(x)$. Assume that x is inserted or rejected during REFINES. If x splits a subsegment, we say x has type 4; if x splits a subfacet, x has type 5; and if x splits a tetrahedron, x has type 6. We prove a stronger statement by induction: if x has type i , then $r_x \geq f(x)/C_i$ for some constants $C_4 > C_5 > C_6 > 1$. Let p be the predecessor of x .

Case 1: x has type 4. Let ab be the subsegment split by x . Since only subfacets are split in rule 7, p must be unweighted and p encroaches ab . If p has type 4, then p must be inserted and it is a subsegment endpoint. In this case, it has been proved that $r_x \geq \mu_6 f(x)$ for some constant $\mu_6 > 0$ [8]. Suppose that p has type 5 or 6. It has been shown [8] that $r_x \geq r_p/\beta_1$ and $r_x \geq \|p - x\|/\beta_2$, where $\beta_1 = \frac{\sqrt{2}}{1 - \tan(\pi/8)}$ and $\beta_2 = \frac{1 + \tan(\pi/8)}{1 - \tan(\pi/8)}$. Since $C_5 > C_6$, the induction assumption implies that $r_p \geq f(p)/C_5$ regardless of the type of p . Thus $\frac{f(x)}{r_x} \leq \beta_1 C_5 + \beta_2$.

So the inequalities $\beta_1 C_5 + \beta_2 \leq C_4$ and $C_4 \geq 1/\mu_6$ should hold.

Case 2: x has type 5. If p is a vertex lying on some element of \mathcal{P} , then by Lemma 5.2, $r_x \geq \mu_7 f(x)$ for some constant $\mu_7 > 0$. Otherwise, p must have type 6. Let h be the subfacet split by x . There are two cases.

Case 2.1: p encroaches h . In this case, it has been shown p is not a vertex in \mathcal{V} [8]. Therefore, we can infer that WQMESH tries to insert p to split a tetrahedron but then it rejects p for encroaching h . And the diametric ball of h is empty. Therefore, $r_x \geq r_p/\sqrt{2}$ and $r_x \geq \|p - x\|$. Thus, $\frac{f(x)}{r_x} \leq \sqrt{2}C_6 + 1$.

Case 2.2: If p does not encroach h , \hat{p} encroaches h in rule 7. And the diametric ball of h is empty. Lemma 5.4 implies that $r_x \geq \mu_5 f(x)$.

So the inequalities $\sqrt{2}C_6 + 1 \leq C_5$ and $C_5 \geq \max\{1/\mu_5, 1/\mu_7\}$ need to hold.

Case 3: x has type 6. In this case, it has been proved that either $r_x \geq \frac{\rho_0\mu_1}{\rho_0\mu_1+1}f(x)$ or $\frac{f(x)}{r_x} \leq \frac{f(p)}{\rho_0 \cdot r_p} + 1 \leq \frac{C_4}{\rho_0} + 1$ [8]. So the inequalities $C_4/\rho_0 + 1 \leq C_6$ and $C_6 \geq \frac{\rho_0\mu_1+1}{\rho_0\mu_1}$ need to hold.

This finishes the case analysis. To satisfy all the above inequalities, we set C_6 to be the maximum of $\frac{\rho_0+\beta_1+\beta_2}{\rho_0-\sqrt{2}\beta_1}$, $\frac{1}{\mu_5}$, $\frac{1}{\mu_6}$, $\frac{1}{\mu_7}$, and $\frac{\rho_0\mu_1+1}{\rho_0\mu_1}$. Then we set $C_5 = \sqrt{2}C_6 + 1$ and $C_4 = \sqrt{2}C_5 + 1$. Notice that $C_i > 0$ for $i = 4, 5, 6$ since $\rho_0 > \sqrt{2}\beta_1 = 2/(1 - \tan(\pi/8))$ as chosen by WQMESH. Finally, we set $C = \max\{C_4, 1/\mu_1\}$. This proves invariant (i).

Consider invariant (ii). For any vertex y that appears in \mathcal{V} currently, $\|x - y\| \geq f(x)/C$. Since $f(x) \geq f(y) - \|x - y\|$, we have $\|x - y\| \geq f(x)/C \geq f(y)/C - \|x - y\|/C$. This implies that $\|x - y\| \geq f(y)/(1 + C)$.

Consider invariant (iii). Invariant (ii) implies that for any vertices $a, b \in \mathcal{V}$, $\|a - b\| \geq f(a)/(1 + C)$. The choice of the values of C and ω_0 enforce that $f(a)/(1 + C) = \omega_0 f(a)$. This proves that the vertex gap property holds. \square

By Lemma 5.5 and a standard packing argument, WQMESH must terminate. Since no protecting ball is encroached by any weighted vertex at the end, the non-shield subsegments appear as weighted Delaunay edges in the end, i.e., the input edges are recovered. The guarded subfacets and subfacets incident to sharp vertices are explicitly kept as weighted Delaunay by WQMESH. The proof of Theorem 7.2 in [6] can be used to show that the other subfacets are recovered as the union of weighted Delaunay triangles. Hence, WQMESH terminates with a conforming weighted Delaunay mesh. It has been proved [8] that at the end of REFINE, the vertices of any tetrahedra with circumradius-edge ratio exceeding ρ_0 are within distance $O(f(x))$ from some point x on some sharp edge.

5.2 Sliver exudation in PUMP

Let \mathcal{M} denote the unweighted Delaunay mesh at the end of REFINE. Let $\mathcal{M}[\omega_0]$ denote a weighted Delaunay mesh obtained after assigning arbitrary weights to the vertices of \mathcal{M} such that the weight property $[\omega_0]$ holds. So $\mathcal{M}[\omega_0]$ may be the mesh obtained at the end of PUMP, but there are many other possibilities for $\mathcal{M}[\omega_0]$ as well. Note that \mathcal{M} and $\mathcal{M}[\omega_0]$ share the same vertex set. Let U denote the set of vertices of tetrahedra in \mathcal{M} with circumradius-edge ratio greater than ρ_0 . In this section we complete the proof of Theorem 4.1.

We need the following result from Talmor's thesis [26]. We have rephrased it to fit our presentation. Given any vertex p in \mathcal{M} , we use V_p to denote its Voronoi cell in the unweighted Voronoi

diagram. We say that V_p is ρ_0 -round if the circumradius-edge ratio of all tetrahedra in \mathcal{M} incident to p are bounded by ρ_0 . In fact, if V_p is ρ_0 -round, it has a bounded aspect ratio. It is known that the lengths of adjacent edges in \mathcal{M} differ by a constant factor $L = 2^{2m-1}\rho_0^{m-1}$, where $m = 2/(1 - \cos \frac{\eta}{4})$ and $\eta = \frac{1}{2} \arctan(2\rho_0 - \sqrt{4\rho_0^2 - 1})$ [7]. Let B_p be a ball centered at p with radius $\rho_0 L \cdot N(p)$.

Lemma 5.6 *Let xz be a line segment lying inside $\bigcup_{p \in \mathcal{V}} V_p \cap B_p$. Let \hat{z} be the sphere centered at z with radius $\|x - z\|$. There exists a constant $K > 0$ such that if \hat{z} is empty and xz intersects ρ_0 -round Voronoi cells only, then xz intersects at most K Voronoi cells and $N(z) \leq C \cdot N(x)$ for some constant $C > 0$.*

Recall the definition of E -neighborhood from section 4.3.

Lemma 5.7 *There exists a constant $\rho_1 > 0$ such that any tetrahedron τ in $\mathcal{M}[\omega_0]$ with orthoradius-edge ratio exceeding ρ_1 is in the K -neighborhood of U where K is given by Lemma 5.6.*

Proof. Let \hat{z} be the orthosphere of some tetrahedron τ in $\mathcal{M}[\omega_0]$. Let qr be the shortest edge of τ . Let x be the intersection point $qz \cap \hat{z}$. Note that x lies inside \hat{q} and \hat{q} lies inside V_q . By assumption, z lies inside $\text{Conv } \mathcal{V}$, the convex hull of \mathcal{V} . So xz lies inside $\text{Conv } \mathcal{V}$ by convexity. It has been proved that $V_p \cap \text{Conv } \mathcal{V} \subseteq V_p \cap B_p$ [6]. So $xz \subseteq \bigcup_{p \in \mathcal{V}} V_p \cap B_p$. Note that \hat{z} is empty. Walk from x towards z . Stop at z or when we have encountered K Voronoi cells (including V_q). If we encounter a Voronoi cell that is not ρ_0 -round, the site v owning the cell is incident to some tetrahedron in \mathcal{M} with circumradius-edge ratio exceeding ρ_0 . That is, $v \in U$. So q is in the K -neighborhood of U . Otherwise, Lemma 5.6 says that we must reach z and $Z \leq N(z) \leq C \cdot N(x)$, where Z is the radius of \hat{z} . The Lipschitz condition implies that $N(x) \leq N(q) + \|q - x\| \leq N(q) + \omega_0 N(q)$. Since $N(q) \leq \|q - r\|$, we have $N(x) \leq (1 + \omega_0) \cdot \|q - r\|$. Thus $Z \leq C(1 + \omega_0) \cdot \|q - r\|$ and the lemma is true for $\rho_1 = C(1 + \omega_0)$. \square

Let \mathcal{G} be the graph consisting of the edges in all possible meshes $\mathcal{M}[\omega_0]$. Let p be a vertex in \mathcal{G} . It has been proved that [6] if both p and its neighbors are not incident to any tetrahedron τ with orthoradius-edge ratio $\rho(\tau) > \rho_1$ in any $\mathcal{M}[\omega_0]$, then $\deg(p) \leq \delta_0$ for some constant δ_0 . Thus, by Lemma 5.7, the following result holds.

Lemma 5.8 *For any vertex p of \mathcal{G} , if p is not in the $(K + 1)$ -neighborhood of U , then $\deg(p) \leq \delta_0$ for some constant δ_0 .*

We are now ready to analyze the effects of weight assignment in PUMP. Let σ_0 be a constant to be specified later. Let $pqrs$ be a tetrahedron with $\sigma(pqrs) \leq \sigma_0$ that is not in the $(K + 1)$ -neighborhood of U . Some vertices may have been assigned some weights already. Suppose that p is pumped with weight P^2 from the interval $[0, L_p^2/9]$. It has been proved [6] that for $pqrs$ to remain weighted Delaunay, P^2 must be at most $k\sigma_0 L_p^2$ for some constant k . That is, $pqrs$ define a forbidden weight interval for p and the length of this weight interval is at most $k\sigma_0 L_p^2$. By Lemma 5.8, there are no more than δ_0^3 tetrahedra incident to p throughout the pumping. Thus the total length of forbidden weight intervals for p is at most $k\delta_0^3 \sigma_0 L_p^2$. Since the weight of p is chosen from $[0, L_p^2/9]$, if $\sigma_0 < 1/(9k\delta_0^3)$, p can be assigned a weight such that p is not incident to any tetrahedron τ with $\sigma(\tau) \leq \sigma_0$.

It may be the case that we abort the pumping of p because it encroaches the two times expansion of some protecting ball. A vertex ball \hat{x} has radius $f(x)/4$. By Lemma 5.1(ii), a protecting ball with center x has radius $\Omega(f(x))$. Thus there is some constant $\mu > 0$ such that $\mu f(x) + L_p/3 \geq \|p - x\|$. It has been shown that $L_p \leq 2\sqrt{2}f(p)$ (Lemma 7.5 [6]). Thus $\mu f(x) + 2\sqrt{2}f(p)/3 \geq \|p - x\|$. By the Lipschitz condition, we have $(\mu + 2\sqrt{2}f(x)/3) + (2\sqrt{2}/3) \cdot \|p - x\| \geq \|p - x\|$, which yields $\|p - x\| \leq \frac{3\mu + 2\sqrt{2}}{3 - 2\sqrt{2}} f(x)$.

This completes the proof of all claims in Theorem 4.1.



Figure 2: PAWN: input (left), slivers before PUMP (middle), few slivers survive PUMP (right). Most of the input face angles close to $\frac{\pi}{2}$ on the boundary are evaluated as acute for numerical tolerances. All surviving slivers are near some such acute angles.

6 Experimental results and conclusions

We have seen that WQMESH is a modification of QMESH [8] (with the step REFINE modified) and an added step for sliver removals. Therefore, we took an implementation of QMESH [28] and modified it to WQMESH. This required adapting QMESH to weighted Delaunay triangulations. We used CGAL [5] for this weighted Delaunay triangulation.

Figure 4 shows the results. Although we proved a bound of $2/(1 - \tan(\pi/8))$ for the circumradius-edge ratio, we do not know the optimal bound on ρ_0 which the algorithm can tolerate. We took $\rho_0 = 2.2$ for the experiments because earlier work proved the guarantees for $\rho_0 = 2$ [21, 24].

We experimented with the performance of WQMESH on a number of data sets. We made several observations summarized in Table 2 and in the bar graph of Figure 3. Table 2 shows the relevant data about input and output. Observe that most of the tetrahedra have good circumradius-edge ratio. For measuring slivers we put a threshold of 3° on dihedral angles, i.e., a tetrahedron is declared sliver if its circumradius-edge ratio is no more than 2.2, but has a dihedral angle less than 3° . One should note that, in theory, the sliver exudation step can only eliminate slivers with abysmally small dihedral angles. However, our experiments show that it is more effective than the theory confirms. The eighth column of Table 2 shows the number of slivers in different dihedral

angle ranges remaining in the output. The distribution of minimum angles (both dihedral and face) for all tetrahedra are shown in Figure 3. It is clear that over 95% of the tetrahedra have minimum angle more than 3° while 90% of them have minimum angles more than 10° .

Slivers with a dihedral angle less than 3° and tetrahedra with circumradius-edge ratio greater than 2.2 are shown in the second column of Figure 4. All of them lie near some small input angle. Third column of the figure shows the triangulation of the input polyhedron.

Table 1: Effect of sliver exudation.

model	# slivers before exudation	# slivers after exudation
Simplebox	6	0
Anchor	9	0
Tower	9	2
Hole	14	0
Hammer	30	6
Pawn	183	16

In spite of the abysmally small angle guaranteed by the sliver exudation step in theory, perhaps its most convincing advantage is its ability to remove slivers with dihedral angles much larger than the ones predicted by theory. Table 1 shows the number of slivers remaining at the end of QMESH which does not have any sliver removal step, and the number of slivers after WQMESH. The data clearly shows that slivers are drastically reduced by the sliver exudation step incorporated in WQMESH. Figure 2 also confirms this conclusion.

Of course, from practical viewpoint it remains open if there is any provable method that can produce guaranteed quality meshing of polyhedra with a sufficiently large lower bound on angles. Also, the case for non-polyhedral inputs remains open.

Acknowledgments

Research is supported by RGC, Hong Kong, China, under grants HKUST6190/02E, Army Research Office, USA under grant DAAD19-02-1-0347 and NSF, USA under grants DMS-0310642 and CCR-0430735.

Table 2: Relevant input and output data.

model	# input points	# sharp elements	#points inserted	# tetrahedra with R/l			# tets(min dihed. ang.)		
				0.6-1.4	1.4-2.2	> 2.2	0-3	3-5	5-10
Anchor	28	27	981	2324	1045	35	0	15	48
Box	32	30	1044	2468	832	41	0	15	42
Wiper	72	58	979	1895	573	19	0	4	43
Blade	36	32	367	777	212	3	0	4	16
Tower	33	59	512	969	399	15	2	1	41

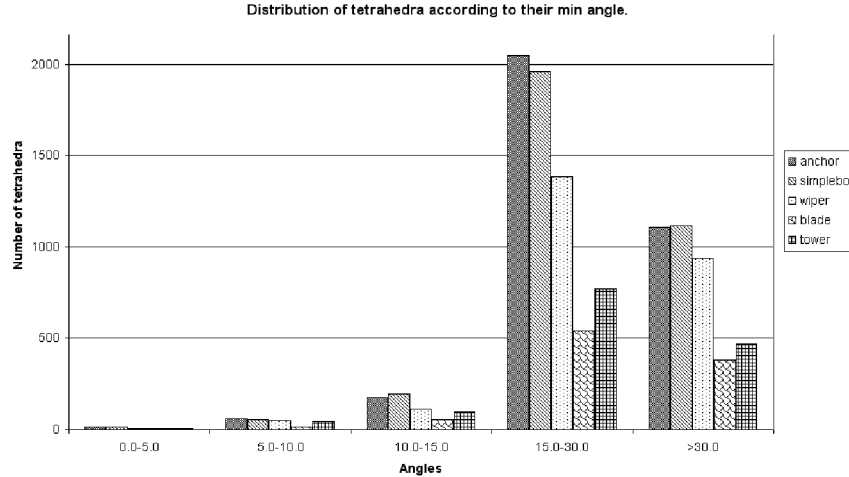


Figure 3:

References

- [1] P. Alliez, D. Cohen-Steiner, M. Yvinec and M. Desbrun. Variational tetrahedral meshing. *Proc. SIGGRAPH 2005*, to appear.
- [2] T.J. Baker. Automatic mesh generation for complex three-dimensional regions using a constrained Delaunay triangulation. *Engineering with Computers*, 5 (1989), 161–175.
- [3] C. Boivin and C. Ollivier-Gooch. Guaranteed-quality triangular mesh generation for domains with curved boundaries. *Intl. J. Numer. Methods in Engineer.*, 55 (2002), 1185–1213.
- [4] H. Borouchaki, P.L. George, and S.H. Lo. Boundary enforcement by facet splits in Delaunay based mesh generation. *Numerical Grid Generation in Computational Field Simulations*, 2000, 203–221.
- [5] Computational Geometry Algorithms Library (CGAL). URL: www.cgal.org
- [6] S.-W. Cheng and T. K. Dey. Quality meshing with weighted Delaunay refinement. *SIAM J. Comput.*, 33 (2003), 69–93.
- [7] S.-W. Cheng, T. K. Dey, H. Edelsbrunner, M. A. Facello and S.-H. Teng. Sliver exudation. *J. ACM*, 47 (2000), 883–904.
- [8] S.-W. Cheng, T. K. Dey, E. A. Ramos, and T. Ray. Quality Meshing for Polyhedra with Small Angles. *Proc. 20th Annu. ACM Sympos. Comput. Geom.*, 2004, 290–299. Extended versions available from authors' web-pages.
- [9] S.-W. Cheng and S.-H. Poon. Graded conforming tetrahedralization with bounded radius-edge ratio. *Proc. 14th Annu. ACM-SIAM Sympos. Discrete Algorithms* (2003), 295–304.

- [10] L. P. Chew. Guaranteed-quality Delaunay meshing in 3D. *Proc. 13th Annu. ACM Sympos. Comput. Geom.* (1997), 391–393.
- [11] D. Cohen-Steiner, E. C. de Verdière and M. Yvinec. Conforming Delaunay triangulations in 3D. *Proc. Annu. Sympos. Comput. Geom.* (2002), 199–208.
- [12] H. Edelsbrunner and D. Guoy. An Experimental Study of Sliver Exudation. *Engineering With Computers*, 18 (2002), 229–240.
- [13] H. Edelsbrunner, X.-Y. Li, G. Miller, A. Stathopoulos, D. Talmor, S.-H. Teng, A. Üngör, and N. Walkington. Smoothing cleans up slivers. *Proc. 32th Annu. ACM Sympos. Theory of Computing*, 2000, 273–277.
- [14] H. Edelsbrunner and N. R. Shah. Incremental topological flipping works for regular triangulations. *Algorithmica*, 15 (1996), 223–241.
- [15] X.-Y. Li and S.-H. Teng. Generating well-shaped Delaunay meshes in 3D. *Proc. 12th. Annu. ACM-SIAM Sympos. Discrete Algorithm* (2001), 28–37.
- [16] S.-H. Lo. Volume discretization into tetrahedra-II. 3D triangulation by advancing front approach. *Computers and Structures*, 39 (1991), 501–511.
- [17] R. Lohner. Progress in grid generation via the advancing front technique. *Engineering with Computers*, 12 (1996), 186–210.
- [18] S. A. Mitchell and S. A. Vavasis. Quality mesh generation in higher dimensions. *SIAM J. Comput.*, 29 (2000), 1334–1370.
- [19] M. Murphy, D. M. Mount and C. W. Gable. A point placement strategy for conforming Delaunay tetrahedralization. *Intl. J. Comput. Geom. Appl.*, 11 (2001), 669–682.
- [20] S.J. Owen. A survey of unstructured mesh generation technology. *Proc. 7th Intl. Meshing Roundtable 1998*.
- [21] S. Pav and N. Walkington. A robust 3D Delaunay refinement algorithm. *Proc. Intl. Meshing Roundtable 2004*.
- [22] J. Ruppert. A Delaunay refinement algorithm for quality 2-dimensional mesh generation. *J. Algorithms*, 18 (1995), 548–585.
- [23] M.S. Shephard and M.K. Georges. Three-dimensional mesh generation by finite octree technique. *Intl. J. Numer. Methods in Engineer.*, 32 (1991), 709–749.
- [24] J. R. Shewchuk. Tetrahedral mesh generation by Delaunay refinement. *Proc. 14th Annu. ACM Sympos. Comput. Geom.* (1998), 86–95.
- [25] J. R. Shewchuk. Mesh generation for domains with small angles. *Proc. 16th Annu. Sympos. Comput. Geom.* (2000), 1–10.
- [26] D. Talmor. Well-spaced points for numerical methods. Report CMU-CS-97-164, Dept. Comput. Sci., Carnegie-Mellon Univ., Pittsburgh, Penn., 1997.

- [27] N.P. Weatherill and O. Hassan. Efficient three-dimensional Delaunay triangulation with automatic point creation and imposed boundary constraints. *International Journal for Numerical Methods in Engineering*, 37 (1994), 2005–2039.
- [28] <http://www.cse.ohio-state.edu/~tamaldehy/qualmesh.html>

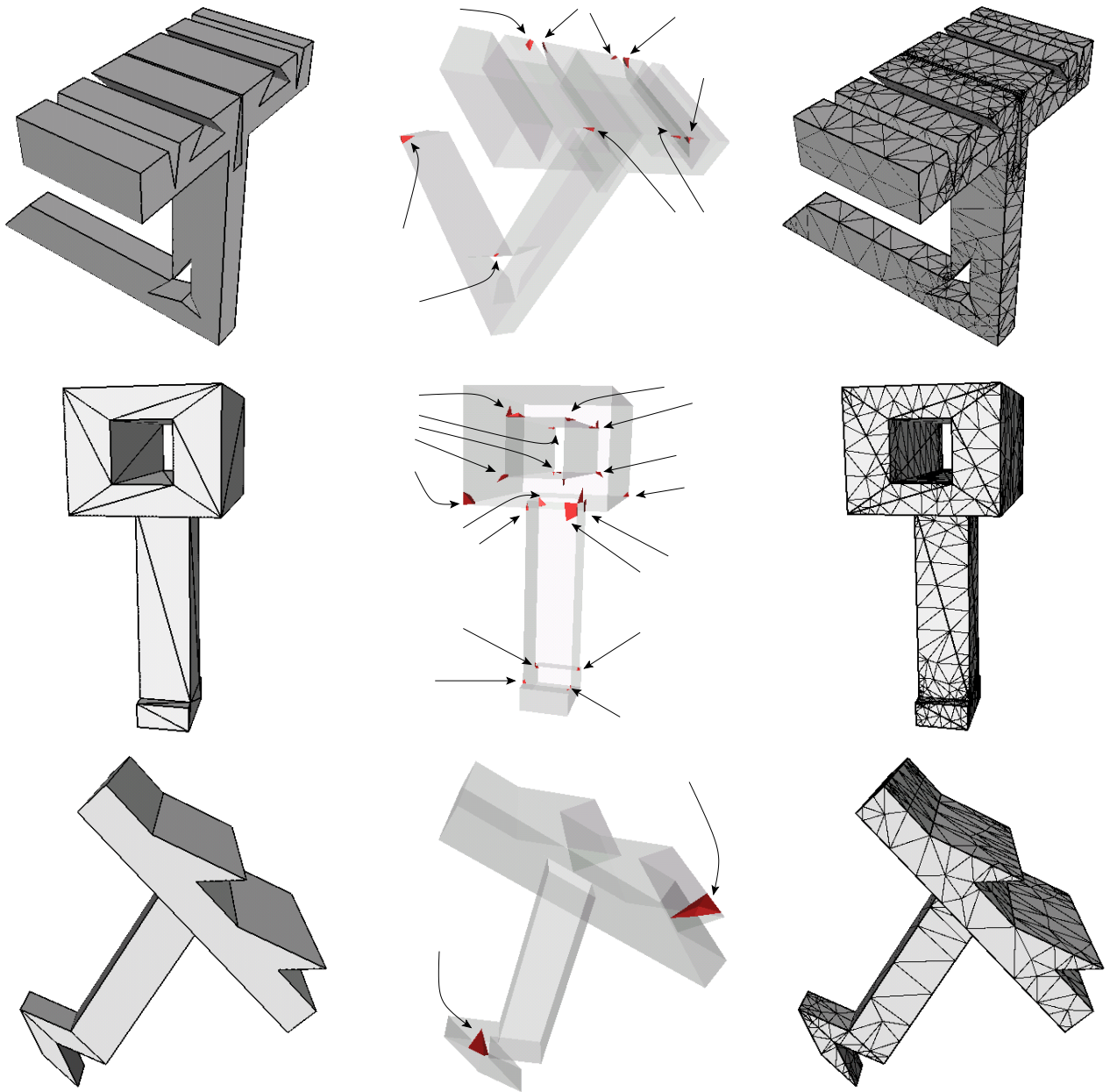


Figure 4: First column shows the input, second column shows the skinny tetrahedra left (none of the slivers are left after exudation) and third column shows the surface triangulation.