

Deforming Surface Meshes

Siu-Wing Cheng and Jiongxin Jin

Abstract We study the problem of maintaining a deforming surface mesh, specified only by a dense sample of n points that move with the surface. We propose a motion model under which the class of (ε, α) -meshes can be efficiently maintained by a combination of edge flips and insertion and deletion of vertices. We can enforce bounded aspect ratios and a small approximation error throughout the deformation.

1 Introduction

1.1 Background

The simulation of deforming surfaces appears in various settings such as the interface between fluids [14, 18, 21, 28], boundary element methods [5, 15], moving cloth [4, 29], and surgery simulation [10, 19]. In this paper, we consider the simulation of a surface that deforms without changing its topology. The surface is specified only by a set of sample points dense with respect to the local feature size (LFS) and the goal is to approximate the surface by a mesh with vertices chosen from the sample points and triangles of bounded aspect ratio, the latter being a desirable feature for numerical simulation. The deformation may make the angles in the mesh smaller between two successive time steps. Our problem is to restore the mesh quality at the next time step before resuming the deformation. The left two images in Fig. 1 shows snapshots of a twisting cylinder output by our algorithm.

Some notation is needed to state our results. The Euclidean distance between two points x and y is denoted by $d(x, y)$. For all $Y \subseteq \mathbb{R}^3$, $d(x, Y) = \inf_{y \in Y} d(x, y)$. For every pair of vectors \mathbf{u} and \mathbf{v} , $\angle(\mathbf{u}, \mathbf{v})$ denotes the angle between them which lies

S.-W. Cheng (✉)

Department of Computer Science & Engineering, HKUST, Clear Water Bay,
Kowloon, Hong Kong
e-mail: scheng@cse.ust.hk

J. Jin

Google Inc., Mountain View, CA, USA
e-mail: jamesjx@google.com

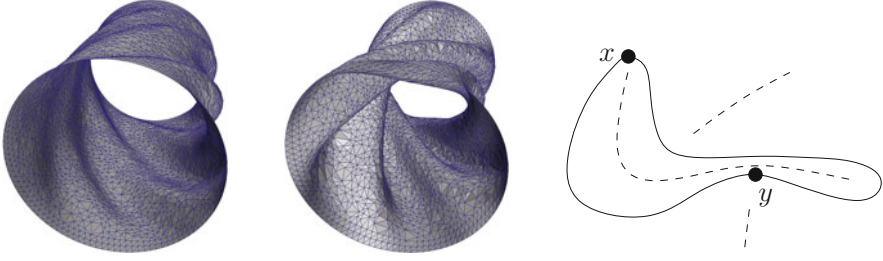


Fig. 1 The left two images show a twisting cylinder. On the right, the dashed skeleton is the medial axis. The local feature sizes are small at x and y because the curvature is high at x and y is near a subcurve that is far way from y along the curve

in the range $[0, \pi]$. Given three points a , b and c , we use $\angle abc$ to denote $\angle(a - b, c - b)$. Let h and h' be two linear objects such as vectors, segments, lines, polygons, and planes. We use $\angle_a(h, h')$ to denote the nonobtuse angle between the affine subspaces spanned by h and h' . $B(x, r)$ denotes the ball with center x and radius r . Given a ball B , we use ∂B to denote its boundary. Given a triangle τ , c_τ denotes its circumcenter, γ_τ denotes its circumradius, B_τ denotes the *diametric ball* $B(c_\tau, \gamma_\tau)$ of τ , and \mathbf{n}_τ denotes a unit vector orthogonal to $\text{aff}(\tau)$.

A *triangulated polygonal surface* T is a set of vertices, edges and triangles such that the intersection of every pair of elements in T is either empty or an element in T , and for every vertex of T , its incident triangles form a topological disk. The union of the vertices, edge and triangles form the *underlying space* $|T|$ of T . The *star* of a vertex $p \in T$, denoted $\text{star}(p)$, is the set of edges and triangles in T that are incident to p .

Let $\Sigma \subset \mathbb{R}^3$ be a closed connected C^2 -smooth surface throughout this paper. For every point $x \in \Sigma$, a *medial ball* B at x is a maximal ball tangent to Σ at x such that the interior of B does not intersect Σ . The *medial axis* \mathcal{M} of Σ is the set of centers of medial balls at points in Σ . The *local feature size* of a point $x \in \Sigma$ is $f(x) = d(x, \mathcal{M})$. The local feature size function f is 1-Lipschitz, i.e., $f(x) \leq f(y) + d(x, y)$ [11].

A finite point set $P \subset \Sigma$ is an ε -*sample* of Σ for some $\varepsilon \in (0, 1)$ if $d(x, P) \leq \varepsilon f(x)$ for every point $x \in \Sigma$. The local feature size is small at a point x if the curvature is high at x or if x is near a point in Σ whose geodesic distance from x is much larger. The image on the right in Fig. 1 illustrates these two cases in 2D. In such cases, a higher sampling density is needed around x for the reconstruction to be faithful. The local feature size must be nonzero for an ε -sample to be well defined. Thus, Σ is required to be C^2 -smooth (no sharp feature or junction). Boundary is also not allowed, although it seems not difficult to handle boundaries in practice as shown in our experiments. Suppose that one has the primitive to compute intersections between Σ and lines, and the primitive to check for every axes-aligned cube C , whether every pair of surface normals in $\Sigma \cap C$ deviate by a small angle. Then, one

can recursively refine an octree partition until the intersections between the leaf cell edges and Σ define a dense sample of Σ [25].

The *nearest point map* φ maps a point $x \in \mathbb{R}^3 \setminus \mathcal{M}$ to the point $\varphi(x) \in \Sigma$ closest to x . We use \mathbf{n}_x to denote the outward unit surface normal at a point $x \in \Sigma$.

A *mesh of Σ* is a triangulated polygonal surface T such that the vertices of T are points in Σ and $|T|$ is homeomorphic to Σ .

1.2 Motion Model

Consider the simulation of a deforming surface that progresses in unit time steps. The surface is specified by n moving sample points on it. At time t , Σ_t denotes the surface, f_t denotes the LFS function of Σ_t , P_t denotes the set of moving sample points, and $\Delta_t(x)$ denotes the speed of a point $x \in \Sigma_t$. For any vertex v of a mesh T of Σ_t , $n_T(v)$ denotes the distance from v to the nearest vertex in T and $R_T(v)$ denotes the largest circumradius of the triangles incident to v .

Suppose that P_t is an ε -sample of Σ_t at all times and, at any time step t , the velocities of the n sample points are returned by the numerical procedure that drives the simulation. Not all points in P_t can be used as mesh vertices in order that no triangle angle is too small. Theoretically, since an ε -sample can be arbitrarily dense locally, it is impossible to prove that some constant fraction of P_t must appear as mesh vertices. However, it is unlikely that P_t is arbitrarily dense anywhere in practice, and we have observed in our experiments that more than 50% of the sample points appear as mesh vertices. We assume that the deformation is *smooth* as defined below.

Definition 1 We say that deformation is *smooth* if the following conditions are satisfied at every time step t :

- (i) For every point $x \in \Sigma_t$, $\Delta_t(x)$ is at most $0.005\varepsilon_0 \sin \alpha_0$ times the LFS of x at t , and $\Delta_{t+1}(x) = O(\Delta_t(x))$.
- (ii) For every pair of sample points p and q , if $d(p, q) \leq \Delta_t(p)$, then $\Delta_t(p) = O(\Delta_t(q))$.
- (iii) At time t , the displaced mesh vertices from the previous time step form an ε_1 -sample, where $\varepsilon_1 \leq \kappa\varepsilon$ for some constant κ .

We are interested in the deformation of a particular class of surfaces meshes defined as follows [7, 8].

Definition 2 For every $\varepsilon \in (0, 1)$ and every constant $\alpha \in (0, \pi/3]$, an (ε, α) -mesh of Σ is a triangulation T that satisfies the following conditions.

- The vertices of T form an ε -sample of Σ .
- The angles of every triangle in T are at least α .

- There exists a triangle τ in T and a vertex p of τ such that $\angle_a(\mathbf{n}_p, \mathbf{n}_\tau) \leq \arcsin\left(\frac{0.8}{1+2\csc(\alpha/2)}\right)$.
- φ restricted to $|T|$ is a homeomorphism between $|T|$ and Σ .

1.3 Main Result

We prove that there exist constants $\varepsilon_0 \in (0, 1)$ and $\alpha_0 \in (0, \pi/6)$ such that an $(\varepsilon_0, \alpha_0)$ -mesh can be constructed before the simulation begins and, at each subsequent time step, an $(\varepsilon_0, \alpha_0)$ -mesh can be restored via edge flips and insertions and deletions of vertices. Theoretically, α_0 can be made close to $\pi/6$, but the sampling density would need to be extremely high. Our experiments suggest that α_0 can be made greater than 10° in practice. The asymptotic running time can be made $O(n)$ [7, 17]. In our experiments (Sect. 6), 90 % of angles are in the range $[30^\circ, 120^\circ]$, only less than 0.02 % of angles are less than 15° , and no angle is smaller than 11° . Our theoretical framework does not allow for topological changes, boundaries, or sharp features. We cater for boundaries in our experiments by keeping all input sample points on the boundaries as mesh vertices and the edges connecting such adjacent sample points as mesh edges.

Level set methods [12, 23, 27] and point-based methods [1, 22, 24] are popular methods to model deforming objects with topological changes, but an explicit mesh is not maintained. Our focus is on fast maintenance of a mesh with theoretical guarantees on its quality instead of producing the deformation. We avoid reconstruction from scratch in order to improve efficiency. Thus, our result is most similar to the prior work on tracking a deforming mesh without any topological change [14, 16, 28], but these prior work do not offer any guarantee. Several techniques have been developed to track and modify a mesh in order to produce topological changes [6, 26, 30]. It may be possible to combine them with our algorithm to allow topological changes and preserve sharp features.

2 High Level Strategy

Our strategy is to maintain an $(\varepsilon_0, \alpha_0)$ -mesh M_t with vertices from P_t that satisfies the following conditions C1–C3. Let ℓ and λ be two constants such that ℓ is sufficiently large and λ is less than 1. The setting of ℓ and λ will be explained later. By C2, α_0 can be set to be $\arcsin(\lambda/2)$.

- C1: For every vertex v of M_t , $n_{M_t}(v) \geq 20(\sin \alpha_0)^{-1} \Delta_t(v)$.
- C2: For every vertex v and triangle τ in M_t , if $v \in B(c_\tau, \ell\gamma_\tau)$, then $n_{M_t}(v) \geq \lambda\gamma_\tau$.
- C3: Some points in P_t may not appear as vertices in M_t . Such a point p is stored in some list $\text{points}(v)$, where v is a vertex of M_t and $d(p, v) \leq 2R_{M_t}(v)$.

Property C1 ensures that a mesh edge turns an angle less than $\alpha_0/10$ from time t to $t + 1$. Property C2 ensures that the triangle circumradii vary smoothly. By C3, every sample point that is not a vertex is stored at some vertex nearby. Let K_{t+1} denote the deformed M_t at time $t + 1$, which has the same connectivity as M_t . Based on C1–C3 and the smoothness of the deformation, we can show that the deformed mesh K_{t+1} is an $(\varepsilon_1, \alpha_1)$ -mesh for some $\alpha_1 \in [\frac{4}{5}\alpha_0, \alpha_0]$ that satisfies the following conditions $\hat{C}1$ – $\hat{C}3$, which are degraded versions of C1–C3. The proof is given in Sect. 3.

- $\hat{C}1$: For any vertex v of K_{t+1} , $n_{K_{t+1}}(v) \geq 18(\sin \alpha_0)^{-1} \Delta_t(v)$.
- $\hat{C}2$: For every vertex v and triangle τ in K_{t+1} , if $v \in B(c_\tau, \frac{1}{2}\ell\gamma_\tau)$, then $n_{K_{t+1}}(v) \geq \frac{1}{2}\lambda\gamma_\tau$.
- $\hat{C}3$: For every vertex v in K_{t+1} and every point $p \in \text{points}(v)$, $d(p, v) = O(R_{K_{t+1}}(v))$. The big-Oh constant depends on that the constant in assumption (ii) of our smooth deformation model.

Our problem is to compute an $(\varepsilon_0, \alpha_0)$ -mesh M_{t+1} from K_{t+1} that satisfies C1–C3 so that the simulation can continue to the next time step and so on.

The initial $(\varepsilon_0, \alpha_0)$ -mesh can be obtained by pruning the sample to an ε_0 -sample that is sparse with respect to their initial velocities and the local feature sizes, followed by running a surface reconstruction algorithm (e.g. [3, 9]) on the pruned sample. The output mesh is an $(\varepsilon_0, \alpha_0)$ -mesh satisfying C1–C3.

Lemma 1 *One can compute an $(\varepsilon_0, \alpha_0)$ -mesh from an ε -sample that satisfies C1, C2 and C3.*

Proof We first compute a surface mesh M' from the ε -sample using a surface reconstruction algorithm. Then for every vertex v , we define its decimation radius $\delta_v = \max\{4\lambda\gamma_\tau - \frac{1}{\ell}d(v, c_\tau) : \text{triangle } \tau \in M'\}$. Initially, all vertices are unmarked. Take an unprocessed vertex v that is unmarked. Mark all vertices in $B(v, \max\{\delta_v, 20(\sin \alpha_0)^{-1}\Delta_0(v)\}) \setminus \{v\}$. Repeat it until all vertices are processed. Run reconstruction again on the unmarked vertices to compute our initial mesh M . We enforce condition C3 by storing each non-vertex sample point in the point list of its nearest vertex in M . We can show that M is an $(\varepsilon_0, \alpha_0)$ -mesh that satisfies C1–C3 by using the same arguments as in the proofs of Lemmas 17 and 18. \square

At time $t + 1$, we call $\text{UPDATE}(K_{t+1})$ to compute M_{t+1} . UPDATE iterates two phases, REFINE and DECIMATE . In the first iteration, REFINE inserts some sample points as vertices to make the vertex set an $O(\lambda\varepsilon_1)$ -sample of Σ_{t+1} . Refinement alone cannot restore the angle lower bound α_0 , because the vertices may become very crowded in some region, where the inter-vertex distance is much less than ε times the local features sizes, and we run out of sample points to refine its neighborhood. In the second phase, DECIMATE deletes some vertices to increase the inter-vertex distances while keeping a good sampling density. At the end of the first iteration, the vertex set forms an $O(\lambda\varepsilon_1)$ -sample. Another iteration of the two phases makes the vertex set an $O(\lambda^2\varepsilon_1)$ -sample. So we obtain M_{t+1} in $O(\log(1/\varepsilon_0)) = O(1)$ iterations.

3 Mesh Deterioration

Inductively, the mesh at time t satisfies conditions C1–C3. We want to show that after the deformation, the mesh at time $t + 1$ satisfies $\hat{C}1$ – $\hat{C}3$, a set of similar conditions with some degraded constants. The next lemma can be proved by a straightforward trigonometric argument.

Lemma 2 *Let $\tau' = x'y'x'$ be a triangle at time t with minimum angle larger than α_0 , and it deforms to $\tau = xyz$ at time $t + 1$. Assume that the displacement of each vertex is at most $\kappa^{-1} \sin \alpha_0$ times the length of any of its incident edges, where κ is a constant greater than 14. Then $(1 - 6/\kappa)\gamma_{\tau'} \leq \gamma_{\tau} \leq (1 + 14/\kappa)\gamma_{\tau'}$.*

Lemma 3 *Let M_t be the mesh at time t satisfying C1–C3. Let K_{t+1} be the deformed mesh at time $t + 1$. Assume $\ell \geq 54$. Then K_{t+1} satisfies $\hat{C}1$ – $\hat{C}3$.*

Proof Consider $\hat{C}1$. Let u be the nearest vertex of v in K_{t+1} . Suppose u and v are at u' and v' , respectively, at time t . By C1, both $\Delta_t(v')$ and $\Delta_t(u')$ are less than $d(u', v')/20$. Therefore, $n_{K_{t+1}}(v) = d(u, v) \geq d(u', v') - \Delta_t(v') - \Delta_t(u') \geq 0.9d(u', v') \geq 0.9n_{M_t}(v')$, which is at least $18(\sin \alpha_0)^{-1}\Delta_t(v)$ by C1.

$\hat{C}2$ requires that for every triangle τ and every vertex v in $B(c_{\tau}, \frac{1}{2}\ell\gamma_{\tau})$, $n_{K_{t+1}}(v)$ is at least $\frac{1}{2}\lambda\gamma_{\tau}$. Let u be the vertex of τ . Let u' , v' and τ' be the counterparts of u , v and τ in M_t . $d(v', u') \leq d(v, c_{\tau}) + d(u, c_{\tau}) + \Delta_t(u') + \Delta_t(v') \leq (\ell/2 + 1)\gamma_{\tau} + d(v', u')/10$. Rearranging the terms, we obtain $d(v', u') \leq \frac{10}{9}(\frac{1}{2}\ell + 1)\gamma_{\tau} < (\ell - 1)\gamma_{\tau'}$ by Lemma 2 and the fact that $\ell \geq 54$. Since $d(v', c_{\tau'}) \leq d(v', u') + \gamma_{\tau'} \leq \ell\gamma_{\tau'}$, C2 implies that $n_{M_t}(v') \geq \lambda\gamma_{\tau'}$ and hence $n_{K_{t+1}}(v) \geq 0.9n_{M_t}(v') \geq 0.9\lambda\gamma_{\tau'}$. Lemma 2 further implies that $n_{K_{t+1}}(v) \geq 0.9\lambda\gamma_{\tau'} \geq \frac{0.9\lambda\gamma_{\tau}}{1+14/20} > \frac{1}{2}\lambda\gamma_{\tau}$, establishing $\hat{C}2$.

Consider $\hat{C}3$. Let v be a vertex in K_{t+1} , and p a sample point in $\text{points}(v)$. Suppose they are at v' and p' at time t , respectively. By C1 and C3, $d(p, v) \leq d(p', v') + \Delta_t(p') + \Delta_t(v') \leq 2R_{M_t}(v') + \Delta_t(p') + n_{M_t}(v')/20 \leq 2.1R_{M_t}(v') + \Delta_t(p')$. If $\Delta_t(p') \leq d(p', v')$, $\Delta_t(p') \leq d(p', v') \leq 2R_{M_t}(v')$. Otherwise, by our assumption on the smoothness of the deformation, $\Delta_t(p') \leq c\Delta_t(v')$ for some constant c . Therefore, $\Delta_t(p') \leq c\Delta_t(v') \leq cn_{M_t}(v')/20 \leq cR_{M_t}(v')/10$. In both cases, $d(p, v) < (5 + c/10)R_{M_t}(v') = O(R_{K_{t+1}}(v))$ by Lemma 2. \square

4 Refinement

In the first iteration, the input to REFINE is K_{t+1} . In remaining iterations, we feed it with the output of DECIMATE. Inductively, the input mesh X of REFINE is an $(\varepsilon_X, \alpha_0)$ -mesh satisfying $\hat{C}1$ – $\hat{C}3$ for some $\varepsilon_X \leq \varepsilon_1$. Our goal is to improve the sampling condition from ε_X to $O(\lambda\varepsilon_X)$.

First, we update the point lists of every vertex, so that a non-vertex sample point is stored in the point list of its nearest vertex. As a result, for every non-vertex sample

point p and the vertex u such that $p \in \text{points}(u)$, $d(p, u) \leq 2R_X(u)$. Then, initialize an intermediate mesh T to be X and incrementally inserts points. We go through the points in P_{t+1} that are not vertices and insert those that are far away from existing vertices. Let p be the current non-vertex point being processed. Let u be the vertex of X such that p is stored in $\text{points}(u)$. Let w be the vertex of the current mesh T nearest to p , which must lie in $B(u, 4R_X(u))$ because $d(u, w) \leq d(u, p) + d(p, w) \leq 2d(p, u) \leq 4R_X(u)$. We find w by searching T within $B(u, 4R_X(u))$. If the distance between p and w is less than $\lambda R_X(u)$ or $20(\sin \alpha_0)^{-1} \Delta_{t+1}(p)$, skip p ; otherwise, insert p as explained below.

Suppose that p is a candidate point to be inserted. We apply a result to be introduced shortly, Theorem 1(iii), to the vertices whose incident triangles intersect $B(u, 4R_X(u))$. In the end, all the triangles intersecting $B(u, 4R_X(u))$ have almost empty diametric balls, and so do their neighboring triangles. The common edge ab between two triangles abc and abd is *flippable* if and only if the diametric ball of abc contains d in its interior and the diametric ball of abd contains c in its interior.

In particular, the triangle $v_1v_2v_3$ nearest to p has an almost empty diametric ball, because the distance between u and the nearest triangle $v_1v_2v_3$ is at most $d(u, p) + d(p, v_1v_2v_3) \leq 2d(u, p) \leq 4R_X(u)$. We insert p by calling $\text{ADD}(T, p, v_1v_2v_3)$. It works as follows. Compute the point \tilde{p} in $v_1v_2v_3$ nearest to p . Split $v_1v_2v_3$ using \tilde{p} . That is, we replace $v_1v_2v_3$ by three triangles $\tilde{p}v_1v_2$, $\tilde{p}v_2v_3$ and $\tilde{p}v_3v_1$. Flip the edges v_1v_2 , v_2v_3 and v_3v_1 if they are flippable. Finally, for each triangle incident to \tilde{p} , replace its vertex \tilde{p} by p . This adds p as a vertex to T .

After all the insertions, we flip the flippable edges in T , and migrate all non-vertex sample points to the point lists of their nearest vertices in the current mesh.

4.1 Mesh Properties

To analyze REFINE, we need some surface sampling results in the literature and some properties of (ε, α) -meshes that we established in another paper [8]. Some preliminary analogous results can be found in [7].

Lemma 4 ([2, 11, 13]) *Let p, q and r be any three points on Σ .*

- (i) *For every $\varepsilon \leq 1/3$, if $d(p, q) \leq \varepsilon f(p)$, then $\angle(\mathbf{n}_p, pq) \geq \pi/2 - \varepsilon$ and $\angle(\mathbf{n}_p, \mathbf{n}_q) \leq 2\varepsilon$.*
- (ii) *For every $\varepsilon \leq 1/10$, if $\gamma_{pqr} \leq \varepsilon f(p)$, then $\angle(\mathbf{n}_p, \mathbf{n}_{pqr}) \leq 10\varepsilon$.*
- (iii) *For every $\varepsilon \leq 1/4$ and every point z on tangent plane at p , if $d(p, z) \leq \varepsilon f(p)$, then $d(z, \Sigma) \leq \varepsilon d(p, z)$.*

Lemma 5 ([8]) *Let p, q and r be any three points on Σ . Let c be any positive constant. Suppose that $\gamma_{pqr} \leq c\varepsilon f(p)$ for some $\varepsilon < \min\{1, 1/(72c)\}$. Then, for every point x in the circumdisk of pqr , $d(x, \varphi(x)) \leq 10c\varepsilon d(p, x) \leq 20c^2\varepsilon^2 f(p)$ and $d(p, \varphi(x)) \leq (2c\varepsilon + 20c^2\varepsilon^2) f(p)$.*

Lemma 6 ([8]) *Let $\mu = 2(\csc \alpha)^{4\pi/\alpha+1}$. There exists an $\varepsilon_0 \in (0, 1)$ depending on α such that for every $\varepsilon \in (0, \varepsilon_0]$, if T is an (ε, α) -mesh of Σ , then*

- (i) *For each vertex $p \in T$ and every triangle $\tau \in \text{star}(p)$, $\gamma_\tau \leq \mu\varepsilon f(p)$ and $\angle_a(\mathbf{n}_p, \mathbf{n}_\tau) < 6\mu\varepsilon$.*
- (ii) *For every pair of triangles $\sigma, \tau \in T$ that share an edge, the dihedral angle at $\sigma \cap \tau$ is greater than $\pi - 12\mu\varepsilon$.*

Lemma 7 ([8, 17]) *For all $c > 1$, if T is an (ε, α) -mesh of Σ for a small enough ε , then for every vertex $p \in T$, $|T| \cap B(p, c\mu\varepsilon f(p))$ is connected and it projects injectively onto any plane that makes an angle at least $\pi/3$ with \mathbf{n}_p .*

Theorem 1 ([7, 8]) *For every constant $c \in (0, 0.5)$ and every constant $\alpha \in [0, \pi/3]$, there exists $\varepsilon_0 \in (0, 1)$ depending on c and α such that for every $\varepsilon \in (0, \varepsilon_0]$, if T is an (ε, α) -mesh of a connected closed smooth surface, then the following properties are satisfied. The common edge ab between two triangles abc and abd is flippable if and only if the diametric ball of abc contains d in its interior and the diametric ball of abd contains c in its interior.*

- (i) *We can flip flippable edges in T until no edge is flippable in time linear in the number of vertices in T . An (ε, α) -mesh T' is produced in the end and for every triangle $\tau \in T'$, $B(c_\tau, (1 - \varepsilon^c)\gamma_\tau)$ does not contain any vertex.*
- (ii) *For every vertex $p \in T$ and every triangle $\tau \in \text{star}(p)$, if $B(c_\tau, (1 - \varepsilon^c)\gamma_\tau)$ does not contain any vertex, then $\gamma_\tau \leq (\varepsilon + O(\varepsilon^{1+c}))f(p)$.*
- (iii) *Given any subset V of vertices of T , we can flip flippable edges in $O(|V|)$ time to produce an (ε, α) -mesh T' so that for every triangle $\tau \in T'$ that is incident to a vertex in V or a neighbor of a vertex in V , $B(c_\tau, (1 - \varepsilon^c)\gamma_\tau)$ does not contain any vertex.*

4.2 Analysis of Refinement

We apply the mesh properties to analyze the effects of refinement. The proofs of the next two technical lemmas are straightforward and omitted.

Lemma 8 *Let T be an (ε, α) -mesh of Σ for a sufficiently small ε .*

- (i) *For every point x in the circumdisk of a triangle $\tau \in T$, $d(x, \varphi(x)) \leq 10\mu\varepsilon\gamma_\tau \leq 10\mu^2\varepsilon^2 f(\varphi(x))$.*
- (ii) *For every point $y \in \Sigma$, the distance between y and its nearest point $z \in |T|$ is at most $10\mu^2\varepsilon^2 f(y)$. For any triangle $\sigma \in T$ that contains z , $d(y, z) < 24\mu\varepsilon\gamma_\sigma$.*

Lemma 9 *Let T_1 and T_2 be a (ξ_1, θ_1) -mesh and a (ξ_2, θ_2) -mesh of Σ , respectively, possibly with different vertex sets. Let τ be a triangle in T_1 . Let σ be the triangle in T_2 nearest to $\varphi(c_\tau)$.*

- (i) $d(c_\tau, \sigma) \leq 10\mu\xi_1\gamma_\tau + 24\mu\xi_2\gamma_\sigma$.

- (ii) Let β be any constant in the range $(0, 1]$. Assume that ξ_1 and ξ_2 are sufficiently small. If $B(c_\tau, \beta\gamma_\tau)$ does not contain any vertex of T_2 , then $\gamma_\sigma \geq 0.9\beta\gamma_\tau$.

Throughout the procedure, we need to maintain certain invariants so that the local edge-flip algorithm works (which requires a constant lower bound on angles).

Lemma 10 *Let X be an $(\varepsilon_X, \alpha_0)$ -mesh satisfying $\hat{C}1$ – $\hat{C}3$. During the execution of $\text{REFINE}(X)$, the following invariants on the intermediate mesh T are maintained.*

- (i) For every vertex a in X and every vertex b in T , if $d(a, b) \leq (\ell/4 - 3)R_X(a)$, then $n_T(b) \geq \frac{1}{4}\lambda^2 R_X(a)$.
- (ii) T is an (ε_X, θ) -mesh for some constant $\theta > 0$.

Proof Consider invariant (i). Let T be the mesh just after we add a vertex p . Let u be the vertex of X nearest to p . It is also the vertex whose point list contains p at the beginning of the main loop.

We first show that invariant (i) holds for $b = p$. Let a be any vertex in X such that $d(a, p) \leq (\ell/4 - 3)R_X(a)$. If $R_X(a) \leq R_X(u)$, then since $n_T(p) \geq \lambda R_X(u)$ for us to decide to insert p , we have $n_T(p) \geq \lambda R_X(a)$ as desired. Assume that $R_X(a) > R_X(u)$. We have $d(a, u) \leq d(a, p) + d(p, u) \leq (\ell/4 - 3)R_X(a) + 2R_X(u) < (\ell/4 - 1)R_X(a)$. Let τ be the triangle incident to a in X with the largest circumradius, i.e., $\gamma_\tau = R_X(a)$. Then, $d(u, c_\tau) \leq d(a, u) + d(a, c_\tau) \leq (\ell/4)\gamma_\tau$. So $\hat{C}2$ applies and implies that $R_X(u) \geq \frac{1}{2}n_X(u) \geq \frac{1}{4}\lambda\gamma_\tau = \frac{1}{4}\lambda R_X(a)$. Since $n_X(p) \geq \lambda R_X(u)$ for us to insert p , we conclude that $n_X(p) \geq \frac{1}{4}\lambda^2 R_X(a)$. This proves invariant (i) for the new vertex p .

Now consider a vertex b of T other than p . If the nearest vertex to b is not changed by the insertion of p , then invariant (i) holds for b inductively. Assume that the nearest vertex of b becomes p . Let a be any vertex of X such that $d(a, b) \leq (\ell/4 - 3)R_X(a)$. If $R_X(a) \leq R_X(u)$, then $n_T(b) = d(p, b) \geq n_T(p) \geq \lambda R_X(u) \geq \lambda R_X(a)$. So invariant (i) holds for b in this case. Assume that $R_X(a) > R_X(u)$. Since p is the nearest vertex of b , we have $d(p, b) \leq d(a, b) \leq (\ell/4 - 3)R_X(a)$. This implies that $d(a, u) \leq d(a, b) + d(p, b) + d(p, u) \leq (\ell/2 - 4)R_X(a)$. Then, we can invoke the same analysis as in the previous paragraph to show that $R_X(u) \geq \frac{1}{4}\lambda R_X(a)$. It follows that $n_X(b) = d(p, b) \geq n_X(p) \geq \frac{1}{4}\lambda^2 R_X(a)$. This proves invariant (i).

Consider invariant (ii). The mesh density cannot decrease because vertices are being inserted. We show that any new angle in X is at least some constant θ after inserting a point p . We omit the argument for establishing the bound on the triangle circumradii and the nearest point map being a homeomorphism as required by the definition of an (ε_X, θ) -mesh, which is similar to the proof of Theorem 1(i) [8].

Suppose that a point p is inserted into the triangle $v_1v_2v_3$. That is, $v_1v_2v_3$ is the closest triangle to p in the current mesh and we split $v_1v_2v_3$ into three smaller triangles by connecting the three vertices to the projection \tilde{p} of p on $v_1v_2v_3$. Let qv_1v_2 be the triangle that shares v_1v_2 with $\tilde{p}v_1v_2$.

We first bound the angles in the triangle qv_1v_2 from below. Applying Theorem 1(iii) makes the diametric ball of qv_1v_2 almost empty. Thus,

$B(c_{qv_1v_2}, 0.8\gamma_{qv_1v_2})$ does not contain any vertex of X . Then by Lemma 9, we can find a triangle σ in X such that $\gamma_\sigma > 0.7\gamma_{qv_1v_2}$, and $d(c_{qv_1v_2}, \sigma) < 0.1\gamma_{qv_1v_2} + 0.1\gamma_\sigma$. Take the vertex a of σ nearest to $c_{qv_1v_2}$. The distance between a and any vertex of qv_1v_2 is at most

$$d(a, c_{qv_1v_2}) + \gamma_{qv_1v_2} < \gamma_\sigma + (0.1\gamma_{qv_1v_2} + 0.1\gamma_\sigma) + \gamma_{qv_1v_2} < 3\gamma_\sigma. \quad (1)$$

Thus, invariant (i) applies to a and any vertex of qv_1v_2 . We conclude that qv_1v_2 has edge lengths at least $\frac{1}{4}\lambda^2 R_X(a) \geq \frac{1}{4}\lambda^2\gamma_\sigma > \frac{1}{6}\lambda^2\gamma_{qv_1v_2}$. The angles in qv_1v_2 are then at least $\psi = \arcsin(\lambda^2/12)$. Similarly, all angles in $v_1v_2v_3$ are at least ψ .

Since we decide to insert p , $n_T(p) \geq \lambda R_X(u)$. As argued in proving (i), this leads to $n_T(p) \geq \frac{1}{4}\lambda^2 R_X(a)$, so $n_T(p) > \frac{1}{6}\lambda^2\gamma_{qv_1v_2}$. This implies that $d(p, v_1)$ and $d(p, v_2)$ are at least $\frac{1}{6}\lambda^2\gamma_{qv_1v_2}$. Since \tilde{p} is the point on the current mesh nearest to p , Lemma 8(ii) implies that $d(p, \tilde{p}) \leq (40\kappa_{\text{rad}}\varepsilon_X)\gamma_{v_1v_2v_3} \leq \frac{40\kappa_{\text{rad}}\varepsilon_X}{\sin\psi}\gamma_{qv_1v_2} \leq \frac{40\kappa_{\text{rad}}\varepsilon_1}{\sin\psi}\gamma_{qv_1v_2}$. So for a small enough ε_1 , $d(\tilde{p}, v_1)$ and $d(\tilde{p}, v_2)$ are at least $\frac{1}{7}\lambda^2\gamma_{qv_1v_2}$.

Suppose that $\gamma_{qv_1v_2} \geq \gamma_{\tilde{p}v_1v_2} \sin(\psi/2)$. So $\sin\angle\tilde{p}v_1v_2 = \frac{d(\tilde{p}, v_2)}{2\gamma_{\tilde{p}v_1v_2}} \geq \frac{\frac{1}{7}\lambda^2\gamma_{qv_1v_2}}{2\gamma_{qv_1v_2}/\sin(\psi/2)} \geq \frac{\lambda^2}{14} \sin(\psi/2)$. It follows that $\angle\tilde{p}v_2v_1 \geq \arcsin(\frac{\lambda^2}{14} \sin(\psi/2))$. Similarly, $\angle\tilde{p}v_2v_1 \geq \arcsin(\frac{\lambda^2}{14} \sin(\psi/2))$. Also, we have $\angle v_1\tilde{p}v_2 \geq \angle v_1v_3v_2 \geq \psi$.

Suppose that $\gamma_{qv_1v_2} \leq \gamma_{\tilde{p}v_1v_2} \sin(\psi/2)$. We claim that v_1v_2 is flippable. Since $\angle v_1\tilde{p}v_2 \geq \angle v_1v_3v_2 \geq \psi$ and $\sin\angle v_1\tilde{p}v_2 = \frac{d(v_1, v_2)}{2\gamma_{\tilde{p}v_1v_2}} \leq \frac{2\gamma_{qv_1v_2}}{2\gamma_{\tilde{p}v_1v_2}} \leq \sin(\psi/2)$, the angle $\angle v_1\tilde{p}v_2$ must be obtuse and greater than $\pi - \psi/2$. Imagine that we rotate $v_1\tilde{p}v_2$ while fixing v_1v_2 to make the dihedral angle between $v_1\tilde{p}v_2$ and v_1qv_2 larger. Since $\angle v_1\tilde{p}v_2 > \pi/2$, \tilde{p} moves closer to the boundary of the diametric ball of v_1qv_2 as we rotate $v_1\tilde{p}v_2$. Let p' be the point in the plane of v_1v_2q such that $\angle v_1p'v_2 = \angle v_1\tilde{p}v_2$ and it is on the different side of v_1v_2 from q . So \tilde{p} is in the diametric ball of v_1qv_2 , if p' is in the diametric ball of v_1qv_2 , which is true because $\angle v_1p'v_2 + \angle v_1qv_2 < \pi$. We show below that q also lies inside the diametric ball of $v_1v_2\tilde{p}$, and hence v_1v_2 is flippable. The plane of v_1v_2q intersects $B_{v_1v_2\tilde{p}}$ in a disk D with radius $\gamma \geq (1 - O(\varepsilon))\gamma_{v_1v_2\tilde{p}}$, as the dihedral angle between v_1v_2q and $v_1v_2\tilde{p}$ is $\pi - O(\varepsilon)$. If q is outside $B_{v_1v_2\tilde{p}}$, then q is outside D , and $\psi \leq \angle v_1qv_2 \leq \arcsin\left(\frac{d(v_1, v_2)}{2\gamma}\right)$. Since $d(v_1, v_2) = 2\gamma_{v_1v_2\tilde{p}} \sin\angle v_1\tilde{p}v_2 \leq 2\gamma_{v_1v_2\tilde{p}} \sin(\psi/2)$, we get $\sin\psi \leq (1 + O(\varepsilon)) \sin(\psi/2) \Leftrightarrow 2\cos(\psi/2) \leq 1 + O(\varepsilon)$, which is impossible for $\psi < \pi/3$. So q lies inside the diametric ball of $v_1v_2\tilde{p}$.

Flipping v_1v_2 produces $v_1\tilde{p}q$ and $v_2\tilde{p}q$. First, $\angle\tilde{p}v_1q \geq \angle v_2v_1q \geq \psi$, as the dihedral angle between qv_1v_2 and $\tilde{p}v_1v_2$ is obtuse. Next, if $\angle v_1\tilde{p}q$ is non-obtuse, since v_1v_2 is flippable, one can show that $\angle v_1\tilde{p}q \geq \angle v_1v_2q \geq \psi$ [8]. If $\angle v_1\tilde{p}q$ is obtuse, it is less than $\pi - \angle\tilde{p}v_1q \leq \pi - \psi$. This implies that $\sin\angle v_1\tilde{p}q \geq \sin\psi$. Finally, we can bound $\angle v_1q\tilde{p}$: $\sin\angle v_1q\tilde{p} = \frac{d(\tilde{p}, v_1)}{d(q, v_1)} \cdot \sin\angle v_1\tilde{p}q \geq \frac{\frac{1}{7}\lambda^2\gamma_{qv_1v_2}}{2\gamma_{qv_1v_2}} \cdot \sin\psi = (\lambda^2/14) \sin\psi$. A similar argument bounds the angles in $v_2q\tilde{p}$ from below.

So all new angles are at least $\min\{\psi, \psi', \psi''\}$ before lifting the triangles incident to \tilde{p} to p , where $\psi' = \arcsin(\frac{\lambda^2}{14} \sin(\psi/2))$, and $\psi'' = \arcsin(\frac{\lambda^2}{14} \sin\psi)$. Since p

is very close to \tilde{p} , the lifting decreases the angles by only a constant factor. So an angle lower bound θ can be preserved by an appropriate setting of θ . \square

The lemma below summarizes the properties of the final mesh obtained.

Lemma 11 *Let X be an $(\varepsilon_X, \alpha_0)$ -mesh satisfying $\hat{C}1$ – $\hat{C}3$. $\text{REFINE}(X)$ runs in time linear in the number of sample points, and produces a mesh Y such that:*

- (i) *For every triangle τ in Y , there is no vertex in $B(c_\tau, 0.8\gamma_\tau)$, and if a vertex v lies in $B(c_\tau, (\ell/6 - 4)\gamma_\tau)$, then $n_Y(v) \geq \frac{1}{6}\lambda^2\gamma_\tau$.*
- (ii) *Y is an (ε_Y, θ) -mesh, where $\theta \in (0, \pi/3]$ and $\varepsilon_Y = \max\{0.3\varepsilon_0, 4\mu\lambda\varepsilon_X\}$.*

Proof Consider the running time of $\text{REFINE}(X)$. We first show that updating the nearest vertex for each sample point takes $O(1)$ time. For every non-vertex sample point p , it is initially stored in the list of some vertex u such that $d(p, u) = O(R_X(u))$, as X satisfies $\hat{C}3$. Set ℓ large enough so that $2d(p, u) \leq (\ell/2)R_X(u)$. The distance between p and its nearest vertex w is at most $d(p, u)$, so $d(w, u) \leq 2d(p, u)$. All triangles intersecting $B(u, 2d(p, u))$ are connected, and they project injectively in the tangent plane at u by Lemma 7. $\hat{C}2$ ensures that all edges of those triangles have lengths at least $\Omega(R_X(u))$. Since all angles in X are at least some constant, a standard packing argument shows that there are only $O(1)$ triangles intersecting $B(u, 2d(p, u))$. Therefore, we can search for the nearest vertex of p by traversing the triangles intersecting $B(u, 2d(p, u))$ in $O(1)$ time.

Inside the main loop, for each sample point p , we need to find its nearest vertex again. The point p is stored in the point list of the vertex u in X nearest to p . So $d(p, u) \leq 2R_X(u)$. Lemma 10(ii) ensures that all angles in an intermediate mesh are at least some constant. Lemma 10(i) implies that all edges intersecting $B(u, 4R_X(u))$ have lengths $\Omega(R_X(u))$. Then the nearest vertex of p can be found in $O(1)$ time by the same reasoning. Consider edge flips before the search for the nearest triangle of p . There are constant number of vertices whose incident triangles intersect $B(u, 4R_X(u))$. So by Theorem 1(iii), it takes $O(1)$ time to flip edges so that their incident triangles have almost empty diametric balls. Finding the nearest triangle of p can also be done in $O(1)$ time by traversing the triangles intersecting $B(u, 4R_X(u))$. The remaining operations in the main loop clearly take $O(1)$ time.

The post-processing, including the edge flips and point migrations, can be done in linear time. Therefore, REFINE runs in linear time.

Consider (i). Since edges are flipped in post-processing until none is flippable, Theorem 1(i) any triangle τ in Y , $B(c_\tau, 0.8\gamma_\tau)$ does not contain any vertex. Let v be a vertex such that $d(v, c_\tau) \leq (\ell/6 - 4)\gamma_\tau$. By Lemma 9, we can find a triangle σ in X such that $d(c_\tau, \sigma) < 0.1(\gamma_\tau + \gamma_\sigma)$ and $\gamma_\sigma \geq 0.7\gamma_\tau$. Let a be a vertex of σ . So we have $d(a, v) \leq d(a, c_\tau) + d(v, c_\tau) \leq 2\gamma_\sigma + 0.1(\gamma_\tau + \gamma_\sigma) + (\ell/6 - 4)\gamma_\tau \leq (\ell/4 - 3)\gamma_\sigma$. Lemma 10(i) applies and implies that $n_Y(v) \geq \frac{1}{4}\lambda^2 R_X(a) \geq \frac{1}{4}\lambda^2\gamma_\sigma > \frac{1}{6}\lambda^2\gamma_\tau$.

Consider (ii). The angle lower bound θ follows from Lemma 10(ii). It remains to show that the vertices of Y form an ε_Y -sample of Σ_{t+1} . Take any point x in Σ_{t+1} . Since P_{t+1} remains an ε -sample of Σ_{t+1} in our motion model, there exists $p \in P_{t+1}$ such that $d(p, x) \leq \varepsilon f_{t+1}(x)$. If p is a vertex of Y , we are done. Suppose that p is not a vertex of Y . Let u be the vertex of X where $p \in \text{points}(u)$. Recall that, before

the main loop, we move every sample point to the point list of its nearest vertex in X , so $d(p, u) \leq 2R_X(u)$. Let w' be the nearest vertex to p in the intermediate mesh when we decided not to insert p . Let w be the vertex of Y nearest to p . The point p was not inserted because $d(p, w') \leq 20(\sin \alpha_0)^{-1} \Delta_{t+1}(p)$ or $d(p, w') \leq \lambda R_X(u)$. Since nearest vertex distances can only decrease in the insertion phase, we have $d(p, w) \leq d(p, w') \leq 20(\sin \alpha_0)^{-1} \Delta_{t+1}(p)$ or $d(p, w) \leq d(p, w') \leq \lambda R_X(u)$.

In the case that $d(p, w) \leq 20(\sin \alpha_0)^{-1} \Delta_{t+1}(p)$, we have $d(p, w) \leq 0.1\varepsilon_0 f_{t+1}(p)$ as $\Delta_{t+1}(p) \leq (0.005\varepsilon_0 \sin \alpha_0) f_{t+1}(p)$ by the smoothness of the deformation. Then, $d(w, x) \leq d(p, w) + d(p, x) \leq 0.1\varepsilon_0 f_{t+1}(p) + \varepsilon f_{t+1}(x)$. The Lipschitzness of LFS implies that $d(w, x) \leq 0.1\varepsilon_0(1 + \varepsilon) f_{t+1}(x) + \varepsilon f_{t+1}(x) < 0.3\varepsilon_0 f_{t+1}(x)$ because we assume that $\varepsilon_0 \geq 10\varepsilon$. Therefore, $d(w, x) \leq \varepsilon_Y f_{t+1}(x)$.

Thus, the requirement of an ε_Y -sample is fulfilled by the vertex w of Y .

Consider the case that $d(p, w) \leq \lambda R_X(u)$. Since X is an $(\varepsilon_X, \alpha_0)$ -mesh, $d(p, u) \leq 2R_X(u) \leq 2\mu\varepsilon_X f_{t+1}(u)$. So $d(u, x) \leq d(p, u) + d(p, x) \leq 2\mu\varepsilon_X f_{t+1}(u) + \varepsilon f_{t+1}(x)$. The Lipschitzness of LFS implies that $f_{t+1}(u) \leq (1 + 2\varepsilon + 4\mu\varepsilon_X) f_{t+1}(x) < 2f_{t+1}(x)$. So $d(p, w) \leq \lambda R_X(u) \leq \mu\lambda\varepsilon_X f_{t+1}(u) = 2\mu\lambda\varepsilon_X f_{t+1}(x)$, and $d(w, x) \leq d(p, w) + d(p, x) \leq (2\mu\lambda\varepsilon_X + \varepsilon) f_{t+1}(x) \leq \max\{4\mu\lambda\varepsilon_X, 2\varepsilon\} \cdot f_{t+1}(x) < \varepsilon_Y f_{t+1}(x)$. \square

5 Decimation

Some vertices must be deleted in order to increase the inter-vertex distances. This will restore conditions C1 and C2 and the angle lower bound $\alpha_0 = \Omega(\lambda)$. Let Y be the output of the refinement phase. Our idea is to define a *decimation radius* δ_v for each vertex v of Y sensitive to the triangle circumradii nearby, so that if we keep v , then the vertices at distance less than radius δ_v from v should be deleted. We want the decimation radius to satisfy the following properties.

- P1: For any two vertices v and u , $\delta_v - \delta_u \leq (\lambda/\ell)d(v, u)$.
P2: There exists a constant κ_{dec} such that for each vertex v , $\delta_v \geq 4\lambda R_Y(v)$ and there exists a triangle τ in Y such that $d(v, c_\tau) \leq \kappa_{\text{dec}}\gamma_\tau$ and $\delta_v \leq \frac{1}{15}\gamma_\tau$.

P1 ensures that the function is smooth, so after the decimation, the nearest vertex distances of close vertices are similar. P2 provides lower and upper bounds on the decimation radius.

Define the decimation radius $\delta_v = \max\{4\lambda\gamma_\tau - (\lambda/\ell)d(v, c_\tau) : \text{triangle } \tau \text{ in } Y\}$. It is easy to verify that it satisfies P1 and P2 with $\kappa_{\text{dec}} = 4\ell$ for $4\lambda \leq \frac{1}{15}$. To evaluate the decimation radii for all vertices, we perform a breadth-first search from each triangle τ , update the decimation radii at the vertices visited, and stop at edges e where $4\lambda\gamma_\tau \leq (\lambda/\ell)d(e, c_\tau)$.

Vertex deletions cannot be performed in an arbitrary order because this may produce tiny angles in an intermediate mesh that makes it impossible to apply Theorem 1(iii) in subsequent vertex deletions. To avoid this problem, we decimate vertex neighborhoods gradually in rounds as explained below.

Algorithm 1 DELETE(mesh T , vertex w)

-
- 1: Flip edges so that all triangles incident to w have almost empty diametric balls.
 - 2: $H \leftarrow$ the plane through a triangles incident to u .
 - 3: Remove all the triangles incident to w , and let P be the polygonal hole.
 - 4: $Q \leftarrow$ the projection of P in H .
 - 5: Compute the constrained Delaunay triangulation of Q .
 - 6: Fill the hole P by lifting these triangles.
-

Define n_{\min} to be the smallest distance between two vertices in Y . Define $\delta_{\max} = \max\{\max\{\delta_v, 20(\sin \alpha_0)^{-1} \Delta_{l+1}(v)\} : \text{vertex } v \text{ in } Y.\}$. Place the vertices of Y into lists S_0, \dots, S_m such that S_i contains v iff $2^i n_{\min} \leq n_Y(v) < 2^{i+1} n_{\min}$, where $m = \max\{0, \lfloor \log_2(\delta_{\max}/n_{\min}) \rfloor\}$. This can be done in $O(m+n)$ time as follows without assuming constant-time integral logarithm operations. We find m by computing δ_{\max} in $O(n)$ time and computing the integral logarithm in $O(m)$ time in a brute force manner. Then initialize $m+1$ empty lists, and insert the vertex v_{\min} with the smallest nearest vertex distance in S_m . Next, start a breadth-first search to traverse the mesh from v_{\max} . Notice that for any two adjacent vertices, their nearest vertex distances differ by a constant factor. So for each vertex encountered during the traversal, knowing at which list any of its adjacent vertices is placed allows us to locate its list in $O(1)$ time.

Then, we initialize a mesh T to be Y and decimate its vertices in $m+1$ rounds. In round i , we decimate the neighborhood of each vertex $v \in S_i$ as follows.

1. Remove v from S_i . If $2^i n_{\min} \leq \max\{\delta_v, 20(\sin \alpha_0)^{-1} \Delta_{l+1}(v)\}$, delete the vertices in $B(v, 2^{i+1} n_{\min})$ other than v from T . Algorithm 1 explains how to delete a vertex from T . Deleted vertices are also removed from the lists S_0, \dots, S_m .
2. Compute the nearest vertex distance $n_T(v)$ of v by finding its nearest vertex in the updated mesh T . The nearest vertex of v can be found by traversing the triangles intersecting $B(v, d(u, v))$ for any edge uv incident to v in T .
3. Compute the index j such that $2^j n_{\min} \leq n_T(v) < 2^{j+1} n_{\min}$. We will show in Lemma 16 that $j = i + O(1)$, so j can be computed in $O(1)$ time. If $j > m$, then $n_T(v) \geq \delta_{\max}$, so we can drop v . If $j \leq m$, we put v in S_j for future processing.

In other words, vertices within distances $2n_{\min}, 4n_{\min}, 8n_{\min}, \dots$ from v are deleted in rounds. After processing all vertices in S_{m+1} , we need to migrate the points in $\text{points}(w)$ for each vertex w deleted.

In the end, the algorithm flips edges until none is flippable, and places every point that is not a vertex to the point list of its nearest vertex.

In the following, we show that the algorithm correctly decimates the neighborhood of each vertex (Lemma 12); and it does not over-decimate in the sense that each deleted vertex is at distance $O(\max\{\delta_v, 20(\sin \alpha_0)^{-1} \Delta_{l+1}(v)\})$ from some vertex v in Z (Lemma 13).

Lemma 12 *Let Z be the output mesh of DECIMATE(Y). The distance between any vertex v to its nearest vertex in Z is at least $\max\{\delta_v, 20(\sin \alpha_0)^{-1} \Delta_{t+1}(v)\}$.*

Proof Throughout the decimation procedure, we maintain the invariant that each vertex v in S_i has nearest vertex distance $n_Z(v) \geq 2^i n_{\min}$, and we drop it only when $2^i n_{\min} > \max\{\delta_v, 20(\sin \alpha_0)^{-1} \Delta_{t+1}(v)\}$. So the lemma follows. \square

Consider a chronological sequence of vertices (w_0, w_1, \dots, w_g) that satisfies the following properties: (1) for $k \in [0, g]$, w_k is a vertex in the initial mesh Y ; (2) w_g is a vertex of an intermediate mesh T in DECIMATE(Y); (3) for $k \in [1, g]$, $w_{k-1} \in B(w_k, 2^{k+1}n_{\min})$ and w_{k-1} was deleted when we processed w_k in some list S_{i_k} . We call the chronological sequence (w_0, w_1, \dots, w_g) a *deletion chain*. It has the following property.

Lemma 13 *Let (w_0, w_1, \dots, w_g) be a deletion chain. Suppose that $w_g \in S_j$ when w_{g-1} was deleted. If $2^j n_{\min} \leq \delta_{w_g}$, then $d(w_0, w_g) \leq 2^{j+2}n_{\min} \leq 4\delta_{w_g}$; otherwise, $d(w_0, w_g) \leq 80(\sin \alpha_0)^{-1} \Delta_{t+1}(w_g) \leq 0.4\varepsilon_0 f(w_g)$.*

Proof For $k \in [0, g-1]$, after deleting the vertices in $B(w_k, 2^{k+1}n_{\min}) \setminus \{w_k\}$ for some i_k , we cannot delete w_k when examining another vertex in the same list S_{i_k} . Thus, $w_{k+1} \in S_{i_{k+1}}$ for some $i_{k+1} > i_k$ when w_k was deleted. We have $d(w_0, w_g) \leq \sum_{k=1}^g d(w_{k-1}, w_k) \leq \sum_{i=0}^j 2^{i+1}n_{\min} \leq 2^{j+2}n_{\min}$. Since $w_g \in S_j$ when w_{g-1} was deleted, we have $2^j n_{\min} \leq \max\{\delta_{w_g}, 20(\sin \alpha_0)^{-1} \Delta_{t+1}(w_g)\}$. If $2^j n_{\min} \leq \delta_{w_g}$, we obtain $d(w_0, w_g) \leq 4\delta_{w_g}$. Suppose that $2^j n_{\min} \leq 20(\sin \alpha_0)^{-1} \Delta_{t+1}(w_g)$. Since we assume that $\Delta_{t+1}(w_g) \leq (0.005\varepsilon_0 \sin \alpha_0) f_{t+1}(w_g)$, we get $d(w_0, w_g) \leq 80(\sin \alpha_0)^{-1} \Delta_{t+1}(w_g) \leq 0.4\varepsilon_0 f_{t+1}(w_g)$. \square

In DELETE(T, w), the first step is to flip edges so that the triangles incident to w have almost empty diametric balls. The next technical result shows the property ensured by this step.

Lemma 14 *Consider the intermediate mesh T during DECIMATE(Y). Suppose that ℓ is sufficiently large and λ is sufficiently small. (Their settings depend on the hidden constant in assumption (i) in our motion model.) Suppose that all the angles are at least some constant before calling DELETE(T, w). Then after the execution of line 1 of DELETE(T, w), for any vertex v of T , $n_T(v) \geq \frac{1}{14} \lambda^2 R_T(v)$.*

Proof Let τ be the triangle incident to v such that $\gamma_\tau = R_T(v)$. Recall that Y stands for the output mesh of the insertion phase. Let σ be the triangle in Y closest to $\varphi_{\Sigma_{t+1}}(c_\tau)$. By Lemma 9, we have $d(c_\tau, \sigma) < 0.1(\gamma_\tau + \gamma_\sigma)$. Let a be any vertex of σ . The distance $d(a, c_\tau)$ is at most $2.1\gamma_\sigma + 0.1\gamma_\tau$.

Suppose that the vertices of σ lie outside $B(c_\tau, 0.5\gamma_\tau)$. Then, $\gamma_\sigma \geq 0.45\gamma_\tau$ by Lemma 9. We have $d(a, v) \leq d(a, c_\tau) + d(v, c_\tau) \leq 2.1\gamma_\sigma + 0.1\gamma_\tau + \gamma_\tau < 5\gamma_\sigma$. Then, for $\ell \geq 54$, Lemma 11(i) implies that $n_T(v) \geq n_Y(v) \geq \frac{1}{6} \lambda^2 \gamma_\sigma \geq \frac{1}{14} \lambda^2 \gamma_\tau$.

Suppose that some vertex a of σ lies in $B(c_\tau, 0.5\gamma_\tau)$. There is a deletion chain (a, \dots, u, w) , where w is a vertex in the current T . Since $B(c_\tau, 0.8\gamma_\tau)$ contains no vertex of T , $d(w, c_\tau) \geq 0.8\gamma_\tau$. So we have $d(a, w) \geq d(w, c_\tau) - d(a, c_\tau) \geq 0.3\gamma_\tau$

Hence, $d(v, w) \leq d(v, c_\tau) + d(a, c_\tau) + d(a, w) < 1.5\gamma_\tau + d(a, w) \leq 6d(a, w)$. Assume that u was deleted when we processed w in some list S_i .

Suppose that $2^i n_{\min} \leq \delta_w$. Lemma 13 implies that $\delta_w \geq \frac{1}{4}d(a, w)$. Combining it with the previous inequality and the smoothness of the decimation radii, we have $\delta_v \geq \delta_w - (\lambda/\ell)d(v, w) \geq \delta_w - (24\lambda/\ell)\delta_w > \delta_w/2$. If we dropped v before processing w (i.e., did not place v in any list for further processing), we must have $n_T(v) \geq \delta_v$, which is greater than $\frac{1}{2}\delta_w$. Then, Lemma 13 implies that $n_T(v) > \delta_w/2 \geq d(a, w)/8 \geq 0.3\gamma_\tau/8$, which is greater than $\lambda^2\gamma_\tau/14$. If we did not drop v before processing w , we must have $n_T(v) \geq 2^i n_{\min}$. Then, Lemma 13 implies that $n_T(v) \geq 2^i n_{\min} \geq d(a, w)/4 \geq 0.3\gamma_\tau/4$, which is also greater than $\lambda^2\gamma_\tau/14$.

If $2^i n_{\min} > \delta_w$, then $d(a, w) \leq 80(\sin \alpha_0)^{-1} \Delta_{t+1}(w)$ by Lemma 13. By assumption, $\Delta_{t+1}(w) \leq c \Delta_t(w)$ for some constant c . Condition C1 and the working of REFINE ensure that $R_Y(w) \geq n_Y(w) \geq \frac{18\Delta_t(w)}{\sin \alpha_0} \geq \frac{18\Delta_{t+1}(w)}{c \sin \alpha_0} \geq \frac{9d(a, w)}{40c}$. Recall that $d(a, w) \geq 0.3\gamma_\tau$. By property P2 of decimation radii, we have $\delta_w \geq 4\lambda R_Y(w) > \frac{9}{10c} \lambda d(a, w) \geq \frac{27}{100c} \lambda \gamma_\tau$. Set ℓ to be a constant greater than $20c/3$. Then $\delta_v \geq \delta_w - \frac{\lambda}{\ell} \cdot d(w, v) \geq \delta_w - \frac{\lambda}{\ell} \cdot 6d(a, w) \geq \delta_w - \frac{\lambda}{\ell} \cdot \frac{10c}{9\lambda} \cdot 6\delta_w \geq \delta_w/2 \geq \frac{27}{200c} \lambda \gamma_\tau$. Then, we can invoke the same analysis in the previous paragraph and show that $n_T(v) \geq \lambda^2\gamma_\tau/14$ by setting λ small enough. \square

By Lemma 14, all triangles have angles at least $\arcsin\left(\frac{1}{28}\lambda^2\right)$ after the execution of line 1 in DELETE. This allows us to show that the deletion of a single vertex preserves an $\Omega(1)$ lower bound on angles. Then, inductively, any intermediate mesh during the decimation has constant lower bound on its smallest angle.

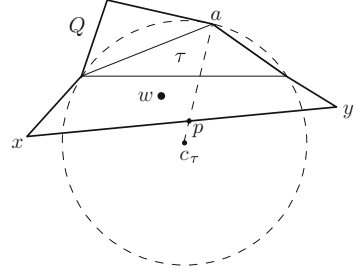
Lemma 15 *At any time during DECIMATE, DELETE(T, w) produces angles that are at least some constant.*

Proof Consider the deletion of a vertex w from T . We first show that the distance between any two vertices adjacent to w is at least $\Omega(n_T(w))$. Take any two vertices a and b incident to w . If $\angle awb \geq \pi/2$, then $d(a, b) \geq d(a, w) \geq n_T(w)$. Suppose $\angle awb < \pi/2$. Then $d(a, b) \geq d(a, w) \cdot \sin \angle awb \geq n_T(w) \cdot \sin \angle awb$. Since each angle at w is at least $\arcsin\left(\frac{1}{28}\lambda^2\right)$ and triangles incident to w are fairly flat, we have $\angle awb \geq \arcsin\left(\frac{1}{28}\lambda^2\right)$, and hence $d(a, b) \geq \frac{1}{28}\lambda^2 n_T(w)$.

Let Q be the projection of the triangles incident to w to the plane of one of these triangles. By Lemma 7, Q is a simple polygon. The distance between any two vertices adjacent to w is shortened by a constant factor by the projection. So the distance between any two vertices of Q is $\Omega(\lambda^2 n_T(w))$. In the following, we prove that any triangle τ in the constrained Delaunay triangulation of Q has circumradius at most $\frac{112}{\lambda^2} R_T(w)$. Then, the ratio of the shortest edge length of τ to γ_τ is $\Omega\left(\lambda^4 \cdot \frac{n_T(w)}{R_T(w)}\right) = \Omega(1)$, which implies that the smallest angle in τ is $\Omega(1)$.

Assume to the contrary that $\gamma_\tau > \frac{112}{\lambda^2} R_T(w)$. Refer to Fig. 2. Since the distance between w and any vertex incident to w is at most $2R_T(w)$, each edge of τ is at most $4R_T(w)$. The large circumradius of τ implies that it has an obtuse angle. Moreover, c_τ lies outside Q because γ_τ is larger than the largest distance between two vertices

Fig. 2 τ is a constrained Delaunay triangle of the projected polygon Q



of Q . Let a be the vertex of τ with an obtuse angle. Since Q contains τ , the line segment ac_τ must intersect at least one edge of Q . Let xy be the one closest to a . Let p denote the intersection point $xy \cap ac_\tau$. Both x and y avoid the interior of the circumcircle of τ by the constrained Delaunay condition. The line segment ap is inside Q by our choice of xy . The polygon Q is star-shaped with w in the kernel, so the line segment wp lies inside Q as well. Thus, w and a are on the same side of xy . If $\angle wxy > \angle axy$ and $\angle wyx > \angle ayx$, the triangle wxy would contain a in its interior, which is impossible. So $\angle wxy \leq \angle axy$ or $\angle wyx \leq \angle ayx$. Assume the former is true.

Recall that all angles in the triangles incident to w are at least $\arcsin(\frac{1}{28}\lambda^2)$. They can be affected by the projection only slightly. So $\angle axy \geq \angle wxy > \arcsin(\frac{1}{36}\lambda^2)$. Since x and y are outside the circumcircle of τ and xy separates a and c_τ , we have $\gamma_\tau \leq \gamma_{axy} \leq \frac{d(a,y)}{2 \sin \angle axy} \leq \frac{28d(a,y)}{\lambda^2}$. Both a and y are at distance at most $2R_T(w)$ from w , so $d(a, y) \leq 4R_T(w)$. It follows that $\gamma_\tau \leq \frac{112}{\lambda^2}R_T(w)$, a contradiction. \square

Lemma 16 *Let v be a vertex in list S_i . Lines 14–15 of DECIMATE place v in list S_j for some $j = i + O(1)$.*

Proof Let T be the current mesh. Consider the mesh T' right before the last vertex u in $B(v, 2^i n_{\min}) \setminus \{v\}$ is deleted. Let $w \neq v$ be a vertex adjacent to u in T' . By Lemma 15, $d(w, u) \leq cn_{T'}(u) \leq cd(u, v)$ for some constant c . So $d(v, w) \leq d(v, u) + d(u, w) \leq (c+1)d(v, u) \leq (c+1)2^i n_{\min}$, and the nearest vertex distance of v after deleting u is at most $(c+1)2^i n_{\min}$.

If u is deleted during the decimation of the neighborhood of v , then w remains a vertex in T . Therefore, $n_T(v) \leq d(v, w) \leq (c+1)2^i n_{\min}$, and $j = i + O(\log c)$.

Suppose that u is deleted during the decimation of the neighborhood of another vertex. Consider the deletion chain (w_0, w_1, \dots, w_g) that contains u , where w_g is a vertex in T . The neighborhood of w_g is decimated in round i or earlier, so we have $d(u, w_g) \leq \sum_{k=0}^{g-1} d(w_k, w_{k+1}) < 2d(w_{g-1}, w_g) \leq 2^{i+1}n_{\min}$. So $n_T(v) \leq d(v, w_g) \leq d(v, u) + d(u, w_g) \leq 3 \cdot 2^i n_{\min}$, meaning that $j \leq i + 2$. \square

Next, we show that the vertices of the output mesh Z of DECIMATE is still a dense sample despite the vertex deletions.

Lemma 17 *Let Y be the output of the refinement phase, and $Z = \text{REFINE}(Y)$. The vertices of Z form an ε_Z -sample, where $\varepsilon_Z = \max\{\varepsilon_0, 15\mu\lambda\varepsilon_X\}$.*

Proof Take any point $x \in \Sigma_{t+1}$. Let w be the nearest vertex in Y to x . So $d(w, x) \leq \varepsilon_Y f_{t+1}(x)$ by Lemma 11. If w is a vertex of Z , we are done. Suppose that w was deleted. Then by Lemma 13, there exists a vertex v in Z such that $d(v, w) \leq \max\{4\delta_v, 0.4\varepsilon_0 f_{t+1}(v)\}$.

Suppose $d(v, w) \leq 0.4\varepsilon_0 f_{t+1}(v)$. Then, $d(x, v) \leq d(x, w) + d(w, v) \leq \varepsilon_Y f_{t+1}(x) + 0.4\varepsilon_0 f_{t+1}(v)$. The Lipschitzness of LFS implies that $d(x, v) \leq \frac{0.4\varepsilon_0 + \varepsilon_Y}{1 - 0.4\varepsilon_0} f_{t+1}(x) \leq (0.7\varepsilon_0 + \varepsilon_Y) \cdot f_{t+1}(x)$. Since $\varepsilon_Y \leq \max\{0.3\varepsilon_0, 4\mu\lambda\varepsilon_X\}$ by Lemma 11, it can be verified that $0.7\varepsilon_0 + \varepsilon_Y \leq \max\{\varepsilon_0, 15\mu\lambda\varepsilon_X\}$. So $d(x, v) \leq \varepsilon_Z f_{t+1}(x)$.

Consider the case that $d(v, w) \leq 4\delta_v$. By property P2 of decimation radii, we can find a triangle τ in Y such that $d(v, c_\tau) \leq \kappa_{\text{dec}}\gamma_\tau$ and $\delta_v \leq \frac{1}{15}\gamma_\tau$. Let a be a vertex of τ . By Theorem 1(ii), $\gamma_\tau \leq 1.5\varepsilon_Y f_{t+1}(a)$. Therefore, we have $d(v, c_\tau) \leq 1.5\kappa_{\text{dec}}\varepsilon_Y f_{t+1}(a)$ and $\delta_v \leq \frac{1}{15}\gamma_\tau \leq 0.1\varepsilon_Y f_{t+1}(a)$. The Lipschitzness of LFS implies that $\delta_v \leq 0.1\varepsilon_Y(1 + 1.5(\kappa_{\text{dec}} + 1)\varepsilon_Y)f_{t+1}(v) \leq 0.2\varepsilon_Y f_{t+1}(v)$. Thus, $d(x, v) \leq 4\delta_v + \varepsilon_Y f_{t+1}(x) \leq 0.8\varepsilon_Y f_{t+1}(v) + \varepsilon_Y f_{t+1}(x) < 3\varepsilon_Y f_{t+1}(x) \leq \varepsilon_Z f_{t+1}(x)$. \square

We are ready to show that DECIMATE restores the angle lower bound to α_0 , while ensuring a good sampling density.

Lemma 18 *Let X be an $(\varepsilon_X, \alpha_1)$ -mesh satisfying $\hat{C}1-\hat{C}3$ for some constants ε_X and α_1 . Let Y be the output mesh of REFINE(X). Assume that the decimation radii for the vertices in Y are known. Then, DECIMATE(Y) runs in linear time and produces an $(\varepsilon_Z, \alpha_0)$ -mesh Z that satisfies conditions C1–C3, where $\varepsilon_Z = \max\{\varepsilon_0, 15\mu\lambda\varepsilon_X\}$ and $\alpha_0 = \Omega(\lambda)$.*

Proof The angles in Y are at least some constant by Lemma 11, so deleting the first vertex can be done in $O(1)$ time. Then by Lemma 15, all angles in the mesh after the deletion are still at least some constant. So Lemmas 14 and 15 are applicable inductively, and hence the deletion of each single vertex can be deleted in $O(1)$ time. Thus, we only need to bound the total number of migrations of vertices in the $m + 1$ rounds of vertex deletions. Let τ be the triangle in Y with the largest circumradius. Let u be the vertex in Y with the largest $\Delta_{t+1}(u)$. So, $m \leq \log\left(\frac{1}{n_{\min}} \max\left\{\frac{1}{15}\gamma_\tau, 20(\sin\alpha_0)^{-1}\Delta_{t+1}(u)\right\}\right) = \log\frac{O(\gamma_\tau + \Delta_t(u))}{n_{\min}} = \log\frac{O(\gamma_\tau + R_Y(u))}{n_{\min}} = O(\log(\gamma_\tau/n_{\min}))$. Since each angle in Y is at least some constant, the circumradii of two adjacent triangles differ by a constant factor, implying that the ratio of the largest to the smallest circumradii is at most $2^{O(n)}$. Therefore, $m = O(n)$. Consider the processing of $v \in S_i$ in a round. In the case that $i < j \leq m + 1$, we deleted some vertex p in a previous round where $d(p, v)$ equals the old $n_M(v)$. We charge to p the migration of v to S_j . We can show that p is charged only $O(1)$ time as in bounding the degree of a planar minimum spanning tree by six. Therefore, the total number of migrations is $O(n)$, implying that the total running time is $O(n)$.

Condition C1 follows from Lemma 12.

Condition C2 requires that for any vertex v and any triangle τ in Z , if $v \in B(c_\tau, \ell\gamma_\tau)$, then $n_M(v) > \lambda\gamma_\tau$. Let σ be the triangle in Y closest to $\varphi_{\Sigma_{t+1}}(c_\tau)$. Let a be any vertex of σ . By Lemma 9, $d(a, c_\tau) < 2.1\gamma_\sigma + 0.1\gamma_\tau$. Suppose that the vertices of σ lie outside $B(c_\tau, 0.7\gamma_\tau)$. Then, Lemma 9 implies that $\gamma_\sigma > 0.6\gamma_\tau$, and we have $d(a, v) \leq d(a, c_\tau) + d(v, c_\tau) \leq 2.1\gamma_\sigma + 0.1\gamma_\tau + \ell\gamma_\tau < 2\ell\gamma_\sigma$. By the properties of decimation radii, $\delta_a \geq 4\lambda\gamma_\sigma$ and $n_Z(v) \geq \delta_v \geq \delta_a - (\lambda/\ell)d(a, v) \geq 2\lambda\gamma_\sigma > \lambda\gamma_\tau$. If some vertex a of σ lies inside $B(c_\tau, 0.7\gamma_\tau)$, $d(a, v) \leq \ell\gamma_\tau + 0.7\gamma_\tau < 2\ell\gamma_\tau$. We can invoke the same proof as that in Lemma 14: identify the deletion chain (a, \dots, w) where w is a vertex of Z , and relate δ_v and δ_w . We have $d(w, c_\tau) \geq 0.8\gamma_\tau$ and $0.1\gamma_\tau \leq d(a, w) \leq 4\delta_w$. Set λ to be a small enough number. We have $\delta_v \geq \delta_w - (\lambda/\ell)d(v, w) \geq \delta_w - (\lambda/\ell)(d(v, a) + d(a, w)) \geq (1 - 4\lambda/\ell)\delta_w - (\lambda/\ell)(2\ell\gamma_\tau) > \frac{1}{80}\gamma_\tau > \lambda\gamma_\tau$.

Condition C3 is satisfied simply because we put every point $p \in P_{t+1}$ that is not a vertex of M in points(w) such that w is the nearest vertex of M to p .

Finally, by Lemma 17, the vertices of the output mesh Z form an ε_Z -sample of Σ_{t+1} , and by C2, α_0 can be set to $\arcsin(\lambda/2)$, so Z is an $(\varepsilon_Z, \alpha_0)$ -mesh. \square

Lemma 19 *Let K_{t+1} be the deformed mesh at time $t + 1$, which is an $(\varepsilon_1, \alpha_1)$ -mesh satisfying $\hat{C}1 - \hat{C}3$. We can iterate REFINE and DECIMATE $O(1)$ times to return an $(\varepsilon_0, \alpha_0)$ -mesh M_{t+1} that satisfies C1–C3.*

Proof Our mesh update procedure iterates REFINE and DECIMATE for $O(\log \frac{1}{\varepsilon_0}) = O(1)$ times. The angle bound α_0 and the conditions C1–C3 are enforced by Lemma 18. Consider the sampling condition. The vertex set of K_{t+1} is an ε_1 -sample, so after the first iteration, we obtain a mesh whose vertex set is an $\max\{\varepsilon_0, O(\lambda\varepsilon_1)\}$ -sample by Lemma 18. Another iteration gives a $\max\{\varepsilon_0, O(\lambda^2\varepsilon_1)\}$ -sample. Therefore, after $O(\log \frac{1}{\varepsilon_0})$ iterations, we can obtain an $(\varepsilon_0, \alpha_0)$ -mesh. \square

Finally, we obtain our deforming mesh maintenance result as stated in the theorem below. The $O(n)$ running time bound is obtained by using an octree to speed up the computation of the decimation radii.

Theorem 2 ([7, 17]) *Consider the simulation of a deforming surface, specified by n moving sample points, that progresses in unit time steps. Suppose that the sample points form an ε -sample for a sufficiently small ε throughout the simulation. There exist constants $\varepsilon_0 \in (10\varepsilon, 1)$ and $\alpha_0 > 0$ such that an $(\varepsilon_0, \alpha_0)$ -mesh can be built before the simulation begins and if the deformation is smooth, an $(\varepsilon_0, \alpha_0)$ -mesh can be restored in $O(n)$ time at each subsequent time step.*

6 Experiments

We built a prototype of the mesh maintenance algorithm and experimented with two test cases on a machine with Intel Xeon E5450 3GHz CPU and 16G memory. Two iterations of the insertion and deletion phases were run on the two test cases, and we found that the mesh size and quality are nearly identical at the end of the first

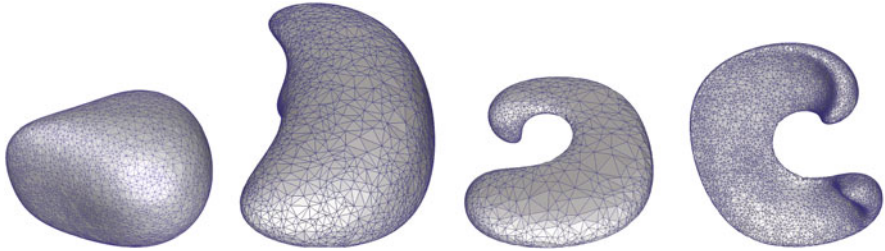


Fig. 3 The *leftmost screen shots* are taken at time 0.2, 0.4 and 0.7 respectively. The *rightmost screen shot* is the back of the object at time 0.7

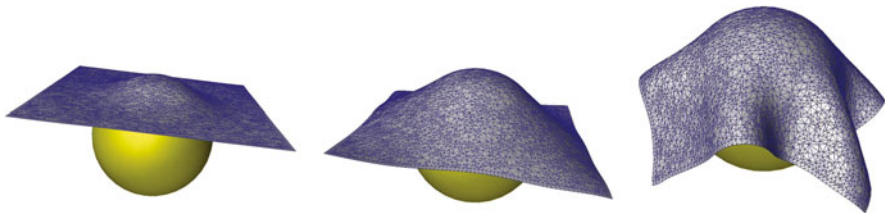


Fig. 4 A piece of cloth falling on a ball

and second iterations. Since the iterations aim to increase the mesh density, it is reasonable to stop if the mesh size stays nearly the same.

The first test case is a deformation of a sphere [12, 20]. The velocity field is defined as: $v_x = 2 \sin^2(\pi x) \sin(2\pi y) \sin(2\pi z)$, $v_y = -\sin(2\pi x) \sin^2(\pi y) \sin(2\pi z)$, and $v_z = -\sin(2\pi x) \sin(2\pi y) \sin^2(\pi z)$. Initially we have 10K random samples on a sphere centered at $(0.35, 0.35, 0.35)$ with radius 0.15. After the simulation starts, the points move with the velocity determined by its current position. Figure 3 shows how the object deforms. The rightmost two images in Fig. 3 show the front and back views of the surface mesh at time 0.7. As the object deforms, the front side is stretched and the sample points move to the back side of object. From the screen shots, one can see that our algorithm is indeed adaptive to the sampling: the triangles in the front are much larger than those at back. The average number of mesh vertices is around 6K, 90% of angles are in the range $[30^\circ, 120^\circ]$, less than 0.02% of them are less than 15° , and none is smaller than 11° . The average update time per time step is less than 0.36 s. Reduction in running time seems possible as we did not optimize the code. For comparison, we reconstructed the mesh from the sample points using Cocone [3]. Cocone took 3.7 s on average per time step.

We also ran our algorithm with different numbers of sample points and observed that the average update time increases linearly with the number of sample points.

Our second test case is the deformation of a piece cloth as it falls on a sphere [4]. 10K sample points are used. The cloth boundary is preserved by not allowing any boundary edge to be flipped. Figure 4 shows the results. The statistics is highly similar to the first test case.

Acknowledgements Research supported by the Research Grant Council, Hong Kong, China (project no. 612107).

References

1. Adams, B., Pauly, M., Keiser, R., Guibas, L.J.: Adaptively sampled particle fluids. *ACM Trans. Graph.* **26**, 3–48 (2007)
2. Amenta, N., Bern, M.: Surface reconstruction by Voronoi filtering. *Discrete Comput. Geom.* **22**, 481–504 (1999)
3. Amenta, N., Choi, S., Dey, T.K., Leekha, N.: A simple algorithm for homeomorphic surface reconstruction. *Int. J. Comput. Geom. Appl.* **12**, 125–141 (2002)
4. Baraff, D., Witkin, A.: Large steps in cloth simulation. In: *SIGGRAPH*, pp. 43–54 (1998)
5. Beer, G., Smith, I., Duenser, C.: *The Boundary Element Method with Programming*. Springer, New York (2008)
6. Bredno, J., Lehmann, T.M., Spitzer, K.: A general discrete contour model in two, three, and four dimensions for topology-adaptive multichannel segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **25**, 550–563 (2003)
7. Cheng, S.-W., Jin, J.: Edge flips and deforming surface meshes. In: *Proceedings of the 28th Annual Symposium on Computational Geometry*, pp. 331–340 (2011)
8. Cheng, S.-W., Jin, J.: Edge flips in surface meshes. Manuscript. <http://www.cse.ust.hk/faculty/scheng/pub/deform.pdf> (2013)
9. Cheng, S.-W., Jin, J., Lau, M.-K.: A fast and simple surface reconstruction algorithm. In: *Proceedings of the 28th Annual Symposium on Computational Geometry*, pp. 69–78 (2012)
10. Delingette, H.: Towards realistic soft tissue modeling in medical simulation. In: *Proceedings of the IEEE: Special Issue on Surgery Simulation*, pp. 512–523 (1998)
11. Dey, T.K.: *Curve and Surface Reconstruction: Algorithms with Mathematical Analysis*. Cambridge University Press, New York (2006)
12. Enright, D., Fedkiw, R., Ferziger, J., Mitchell, I.: A hybrid particle level set method for improved interface capturing. *J. Comput. Phys.* **183**, 83–116 (2002)
13. Giesen, J., Wagner, U.: Shape dimension and intrinsic metric from samples of manifolds. *Discrete Comput. Geom.* **32**, 245–267 (2004)
14. Glimm, J., Grove, J.W., Li, X.L., Tan, D.C.: Robust computational algorithms for dynamic interface tracking in three dimensions. *SIAM J. Sci. Comput.* **21**, 2240–2256 (1999)
15. Hall, W.S.: *The Boundary Element Method*. Kluwer Academic Publishers, Dordrecht (1994)
16. Jiao, X.: Face offsetting: a unified approach for explicit moving interfaces. *J. Comput. Phys.* **220**, 612–625 (2007)
17. Jin, J.: *Surface reconstruction and deformation*. Doctoral Dissertation, The Hong Kong University of Science and Technology (2012)
18. Khayat, R.E.: Three-dimensional boundary element analysis of drop deformation in confined flow for Newtonian and viscoelastic systems. *Int. J. Numer. Methods Fluids* **34**, 241–275 (2000)
19. Koch, R.K., Gross, M.H., Carls, F.R., von Büren, D.F., Fankhauser, G., Parish, Y.I.H.: Simulating facial surgery using finite element methods. In: *SIGGRAPH*, pp. 421–428 (1996)
20. LeVeque, R.J.: High-resolution conservative algorithms for advection in incompressible flow. *SIAM J. Numer. Anal.* **33**, 627–665 (1996)
21. Liu, T., Shen, D., Davatzikos, C.: Deformable registration of cortical structures via hybrid volumetric and surface warping. *NeuroImage* **22**, 1790–1801 (2004)
22. Müller, M., Charypar, D., Gross, M.: Particle-based fluid simulation for interactive applications. In: *SIGGRAPH*, pp. 154–159 (2003)
23. Osher, S., Sethian, J.: Fronts propagating with curvature-dependent speed: algorithms based on Hamiltonian Jacobi formulations. *J. Comput. Phys.* **79**, 12–49 (1988)

24. Pauly, M., Keiser, R., Adams, B., Dutré, P., Gross, M., Guibas, L.J.: Meshless animation of fracturing solids. *ACM Trans. Graph.* **24**, 957–964 (2005)
25. Plantinga, S., Vegter, G.: Isotopic meshing of implicit surfaces. *Vis. Comput.* **23**, 45–58 (2007)
26. Pons, J., Boissonnat, J.D.: Delaunay deformable models: topology-adaptive meshes based on the restricted Delaunay triangulation. In: *CVPR*, 1–8 (2007)
27. Sethian, J.: *Level Set Methods and Fast Marching Methods*. Cambridge University Press, Cambridge (1999)
28. Tryggvason, G., Bunner, B., Esmaeeli, A., Juric, D., Al-Rawahi, N., Tauber, W., Han, J., Nas, S., Jan, Y.-J.: A front-tracking method for the computations of multiphase flow. *J. Comput. Phys.* **169**(2), 708–759 (2001)
29. Volino, P., Magnenat-Thalmann, N.: Comparing efficiency of integration methods for cloth simulation. In: *Proceedings of the International Conference on Computer Graphics*, pp. 265–272 (2001)
30. Wojtan, C., Thüey, N., Gross, M., Turk, G.: Deforming meshes that split and merge. *ACM Trans. Graph.* **28** (2009). Article 76