Ph.D Thesis Proposal: Construction and Deformation of Smooth Surfaces in $\mathbb{R}^3$

Ho-Lun Cheng

April 5, 2000

1 Introduction

Motivation Surface deformation arises in moving boundary problems of physical simulation, where they act as boundaries of spatial domains that grow and shrink with time, and with topology changes. An example is the boundary between the solid and liquid portions of metal during solidification [9]. Another example is the mold filling process. Such processes are simulated through numerical computations on a mesh representing the boundary as a 2-dimensional triangulation of the surface. We also interested in using deformable surfaces in molecular modeling. Like most organic objects, molecules have distinctive shapes and at the same time are able to change that shape without a limited but still infinite-dimensional space. More extreme deformations happen during protein folding when the protein go through the process of taking on their shape.

Background The approach to deforming surfaces in this work is based on the technical notion of the skin surfaces, as introduced in [4]. The main reason for this choice is the existence of fast combinatorial algorithms based on the alpha shape theory [3][5]. One of these algorithms is the automatic deformation of any one skin into any other.

Contributions In this work, we are going to construct a software which will generate the skin surface. The final goal is to construct the skin with the breathing algorithm, which compute the skin when it is deforming i.e. from one skin surface to one another. A 'static' version of skin construction is first designed for some testing on the operations of the triangulation. Also, it can be a comparison to the dynamic version.

Outline We will begin with giving some background on alpha shape theory and the skin surface definition in section 2 and 3. In section 4, we will give some properties of the skin which is useful for the proof of the General Homeomorphism Theorem in section 5. Section 6 and 7 will describe the algorithm for the static and dynamic skin construction algorithm. Software discussion will be given in section 8. Section 9 will discuss the performance of the current version of skin software and we will finish with the future works in section 10.
PART I. BACKGROUND

2 Alpha Shape

We will briefly introduce the weighted Delunay triangulation and alpha shape in this section. Readers can find further information in [3][5][8].

**Weighted Delunay Triangulation** Define $A$ to be a set of weighted points $\tilde{a} = (a, A^2)$ where $a \in \mathbb{R}^3$ with radius $A$, $A^2 \in \mathbb{R}$. We defined the Voronoi cell, $\nu_a$, of $\tilde{a}$ by:

$$\nu_a = \{x \mid \pi_a(x) \leq \pi_b(x), \forall b \in A\}$$

which $\pi_a(x)$ is the squared weighted distance from the point $x$ to $\tilde{a}$:

$$\pi_a(x) = |xa|^2 - A^2$$

We define the Voronoi cell of $X \subset A$ as:

$$\nu_X = \bigcap_{a \in X} \nu_a$$

The Voronoi complex of $A$ is

$$V_A = \{\nu_X \mid \nu_X \neq \emptyset\}$$

Let $X$ be the centers of the set of balls, $\tilde{X}$. The Delunay triangulation is the simplicial complex which consists of Delunay simplices, $\delta_X = \text{conv} \ (X)$:

$$D_A = \{\delta_X \mid \nu_X \in V_A\}$$

Note that for each Delunay simplex, $\delta_X$, there is a dual Voronoi cell $\nu_X$ which has an one-to-one correspondence.

**Weighted Alpha Complex** For $\tilde{a} \in A$, we define $\tilde{a}(\alpha)$ be the sphere with center $a$ and radius $r = \sqrt{A^2 + \alpha^2}$ with $\alpha^2 \in \mathbb{R}$. Note that $\tilde{a}(0) = \tilde{a}$. We define this set of ‘expanded’ spheres as

$$A(\alpha) = \{\tilde{a}(\alpha) \mid \tilde{a} \in A\}$$

The ball could be imaginary if $r^2 < 0$. However, even with the change of the radii, the Delunay triangulation remains the same, i.e. $D_A = D_{A(\alpha)}$. Now, let $B(\alpha) = \bigcup A(\alpha)$ be the unions of these balls. The alpha complex, $K(\alpha)$ at a certain $\alpha$ value is:

$$K(\alpha) = \{\delta_X \mid \nu_X \cap B(\alpha) \neq \emptyset\}$$

$K(\alpha)$ is a sub-complex of $D_A$ and $K(\alpha')$ is a sub-complex of $K(\alpha)$ if $\alpha' < \alpha$.

3 Skin Surface

**Sphere Algebra** We are going to define two operations on the set of spheres, the addition between two spheres, and the scalar multiplication of a sphere.

For each weighted point $\tilde{a}$, we have a weighted distance function $\pi_{\tilde{a}}(x)$. The zero-set of $\pi_{\tilde{a}}$ is the sphere with radius $A$ and center $a$ (Similar situation with $\tilde{a}(\alpha)$). We define the sphere $\tilde{a} + \tilde{b}$ as the zero-set of $\pi_{\tilde{a}} + \pi_{\tilde{b}}$ and $\gamma \cdot \tilde{a}$ as the zero-set of $\gamma \cdot \pi_{\tilde{a}}$. With the two operations, we can define the convex hull of $n$ spheres in $A(\alpha) = \{\tilde{a}(\alpha)_1, \tilde{a}(\alpha)_2, \ldots, \tilde{a}(\alpha)_n\}$

$$\text{conv } A(\alpha) = \{\sum_{\tilde{a}_i \in A(\alpha)} \gamma_i \cdot \tilde{a}_i \mid \sum_{i=1}^{n} \gamma_i = 1, \gamma_i \geq 0, i \in [1..n]\}$$

Besides adding and multiplying, we need also to shrink the spheres. We define $\sqrt{\alpha}$ to be the sphere with the same center but $\frac{1}{\sqrt{\alpha}}$ times the original radius of $\tilde{a}$. The application of the shrinking operation to all spheres in a set $A$ is denoted as $\sqrt{A} = \{\sqrt{a} \mid a \in A\}$. We take all the spheres in the convex hull and shrink each of them. Take the union of all these spheres, the body of the skin surface is defined as

$$\text{body } A(\alpha) = \bigcup \sqrt{\text{conv } A(\alpha)}$$

So, the skin of $A(\alpha)$ is

$$\text{skin } A(\alpha) = \text{env } \sqrt{\text{conv } A(\alpha)}$$
Mixed Cells  Recalled in section 2, each $\delta_X \in D_{\mathcal{A}(\alpha)}$ has a correspondent $\nu_X \in V_{\mathcal{A}(\alpha)}$. We define the mixed cell, $\mu_X$, as

$$\mu_X = (\delta_X + \nu_X)/2$$

Figure 3 shows examples of the four different types of mixed cells corresponding to different cardinalities of $X$. The collection of mixed cells forms a partition of $\mathbb{R}^3$, which is called the mixed complex.

Skin Decomposition  Within each mixed cell, $\mu_X$, the skin surface is completely determined by at most four weighted points in $X$, see [4]. Specifically

$$\text{skin } \mathcal{A}(\alpha) \cap \mu_X = \text{env } \sqrt{\text{conv } X \cap \mu_X}$$

For card $X = 1$ and 4, the envelope within the mixed cell of conv $X \cap \mu_X$ is a sphere and a hyperboloid of revolution for card $X = 2$ and 3. In each case, we define the center as the intersection of the affine subspace defined by $\nu_X$ and that defined by $\delta_X$. In the case of hyperboloid this is the the apex of the asymptotic double-cone, and in the case of the sphere it is the center. It may or may not belong to the mixed cell since $\nu_X \cap \delta_X$ may be empty.
Complementarity We can also reverse the orientation of the skin surface by finding another finite collection of weighted points. These new set of weighted points define the same skin and a complementary body. We define $\mathcal{A}^-(\alpha)$ with skin $\mathcal{A}(\alpha) = \text{skin } \mathcal{A}^+(\alpha)$ and body $\mathcal{A}(\alpha) \cup \text{body } \mathcal{A}^+(\alpha) = \mathbb{R}^3$. Specifically, $\mathcal{A}^-(\alpha)$ contains a weighted point at every Voronoi vertex, $\nu_X$, in $V_A$ (card $X = 4$) and its weight is chosen so that it will be orthogonal to the weighted points in $\hat{X}$. Specifically

\[
\mathcal{A}^-(\alpha) = \{ \hat{a} | \hat{a} = (\nu_X, |ab|^2 - B^2), \text{card}(\hat{X}) = 4, \hat{b} = (b, B^2) \in \hat{X} \text{ and } \hat{X} \subseteq \mathcal{A}(\alpha) \}
\]

Sandwiching Spheres We close this section by stating a rather special and important property of skin surface heavily exploited in this paper.

Sandwich Property. For every point $x$ on the skin surface of $\mathcal{A}(\alpha)$, there are unique spheres $S_x \in \sqrt{\text{conv } \mathcal{A}(\alpha)}$ and $T_x \in \sqrt{\text{conv } \mathcal{A}^+(\alpha)}$ that pass through $x$. Furthermore, $S_x$ and $T_x$ have the same radius.

![Figure 6: $S_x$ and $T_x$ are the sandwiching spheres of point $x$ on the skin surface](image)

We refer $S_x$ and $T_x$ as the sandwiching spheres at $x$ because they squeeze the surface flat in a neighborhood of $x$. They also determine the curvatures at $x$, which we will see at section 4. The fact that $S_x$ and $T_x$ are equally large follows from Lemma 7 in [4].

4 Geometric Properties of Skin

Maximum Curvature For a point $x$ on the skin surface and with a tangent $t_x$, the normal curvature is that of a geodesic passing through $x$ in the direction $t_x$ and let $\kappa(x)$ to be the maximum normal curvature on $x$. For a hyperboloid or a sphere in the form

\[
x^2 + y^2 + z^2 = \pm r^2
\]

we have $1/\sqrt{r^2}$ and $1/\sqrt{\pm r^2 + 2z^2}$ for the maximum curvature of sphere and hyperboloid respectively. Substituting (1) back into the curvatures we get $\kappa(x) = \frac{1}{\rho}$.

Iso-Curvature Lemma. The sphere with radius $c$ and center at the origin is the set of points $x$ on hyperboloids (and spheres) in standard form for which $\kappa(x) = \frac{1}{\rho}$

Also it is convenient to define the length scale, $\rho(x) = 1/\kappa(x)$.

Curvature Properties. Recall that there are two sandwiching spheres of $x$, $S_x \in \sqrt{\text{conv } \mathcal{A}(\alpha)}$ and $T_x \in \sqrt{\text{conv } \mathcal{A}^+(\alpha)}$.

Curvature Sandwich Lemma. For every point $x$ on the skin, $\frac{1}{\kappa(x)}$ is the common radius of the sandwiching spheres $S_x$ and $T_x$.

![Figure 7: The circles $S_x$ and $T_x$ sandwich the hyperbola. Depending on whether we revolve the hyperbola around the vertical or the horizontal axis, we get a two-sheeted or a one-sheeted hyperboloid.](image)
maximum curvature varies continuously over the skin surface. Moreover,

**Curvature Variation Lemma.** If $x$ and $y$ are points on the skin surface then $|\varphi(x) - \varphi(y)| \leq |xy|$.

**Proof.** For sphere case, the curvature will remain constant wherever $x$ and $y$ are. For the case of hyperbola, we can see in figure 8 that $|xy| \geq |xy'|$. Note that $|xy'| = |\varphi(x) - \varphi(y)|$ since $||x|| = \varphi(x)$ and $||y'|| = \varphi(y)$.

![Figure 8: Proof of the Curvature Variation Lemma of the hyperbola case.](image)

**Normal Variation** We try to give a bound on the normal variation. Let the normal of the point $x$ on the skin surface be $\mathbf{n}_x$.

**Normal Variation Lemma.** Let $x, y$ be two points on the skin surfaces and $|xy| < \varphi(x)/2$. Then the angle between the normal vectors of $x$ and $y$ is

$$\angle n_x n_y < 2 \arcsin \frac{|xy|}{\varphi(x) - |xy|}.$$  

**Proof.** Consider the sphere with center $x$ and radius $|xy|$ illustrated in figure 9. All points on the skin surface within the sphere have local length scale at least $\varphi(x) - |xy|$. We find a geodesic path from $x$ to $y$. The normal variation is largest if the total local length scale along the path is as small as possible. The total local length scale of the path is greater than that of the circle with radius $R = \varphi(x) - |xy| > \varphi(x)/2$.

![Figure 9: The path from $x$ to $y$ has less total curvature than the circle passing through $x$ and $y$.](image)

Draw a circle with this radius passing through $x$ and $y$. The length of the arc connecting $x$ and $y$ inside the sphere is

$$L(x, y) = 2R \arcsin \frac{|xy|}{2R}.$$  

The angle between the surface normals at $x$ and $y$ is

$$\angle n_x n_y < L(x, y)/R \leq 2 \arcsin \frac{|xy|}{2(\varphi(x) - |xy|)},$$

as claimed.

**Part II. Meshing**

### 5 Triangulation

Let $N$ be the skin surface skin $A(\alpha)$. A finite set of points $P \subseteq N$ is an $\varepsilon$-sampling if for every point $x \in N$ there is a vertex $p \in P$ which $|xa| < \varepsilon \cdot \varphi(x)$. The goal of this section is to prove that an $\varepsilon$-sampling for skin $A(\alpha)$ defines a restricted Delaunay triangulation homeomorphic to the skin surface, provided $\varepsilon$ satisfies the following inequality.

$$\frac{2\varepsilon}{1 - \varepsilon} \leq 2 \cos(\arcsin \frac{\sqrt{3\varepsilon}}{1 - \varepsilon} + 2 \arcsin \frac{\varepsilon}{2(1 - \varepsilon)})$$

**Restricted Delaunay Triangulation** Let $P \subseteq \mathbb{R}^3$ be a finite set of points on the skin surface.
For each points \( p \in P \), the Voronoi polyhedron \( \nu_p \) is defined. The corresponding restricted Voronoi polyhedron is the intersection with the skin surface, skin \( \mathcal{A}(\alpha) \cap \nu_p \). The restricted Delaunay triangulation, \( D_P \), is the nerve of the collection of restricted polyhedra:

\[
D_P = \{ \text{conv } U \mid U \subseteq P, N \cap \bigcap_{p \in U} \nu_p \neq \emptyset \}
\]

From \([6][1]\), if \( \nu_U \) has the close ball property, then the underlying space of \( D_P \) is homeomorphic to skin \( \mathcal{A}(\alpha) \).

**Closed Ball Property** We say \( V_P = \{ \nu_p \mid p \in P \} \) has the close ball property if the intersection of \( k+1(=1,2,3,4) \) Voronoi polyhedra with \( N \) are

- case \( k = 0 \): a closed disk,
- case \( k = 1 \): empty or a closed interval,
- case \( k = 2 \): empty or a single point,
- case \( k = 3 \): empty.

**Lemmas and Claims** Figure 22 in appendix B shows a map of the proof of the theorem. We will not give the proofs of all the lemmas and claims, in stead, we will state those lemmas and explain briefly.

From the Sandwich Property, Iso-Curvature Lemma, Curvature Sandwich Lemma and the Curvature Variation Lemma, we could prove the following five lemmas about distance and normal on the surface.

**Short Distance Claim.** If points \( x, y \in \text{skin } \mathcal{A}(\alpha) \) belong to a common Voronoi polyhedron then \( |xy| < \frac{2\varepsilon}{1-\varepsilon} \cdot \theta(x) \).

**Long Distance Claim.** If a line meets skin \( \mathcal{A}(\alpha) \) in two points \( x \) and \( y \) and forms an angle smaller than \( \xi \) with the surface normal at \( x \) then \( |xy| > 2 \cos \xi \cdot \theta(x) \).

**Edge Normal Lemma.** The angle between an edge \( ab \) and the surface normal at \( a \) is \( \geq \frac{\pi}{2} - \arcsin \frac{|ab|}{2\theta(a)} \).

**Triangle Normal Lemma.** The angle between the normal of \( abc \) and the surface normal at \( a \) is \( \leq \arcsin \frac{R_{abc}}{2\theta(a)} \), which \( R_{abc} \) is the radius of circumcircle of triangle \( abc \).

With together with condition (I), we can prove

**Voronoi Edge Lemma.** A Voronoi edge of \( V \) intersects the skin surface in at most one point.

**Voronoi Polygon Lemma.** The intersection of a Voronoi polygon of \( V \) with the skin surface is either empty or a closed interval.

**Voronoi Polyhedron Lemma.** The intersection of a Voronoi polyhedra of \( V \) with the skin surface is a closed disk.

The three lemmas are sufficient to prove \( V_P \) has the close property. Thus, we proved the General Homeomorphism Theorem for the skin.

### 6 Surface Construction

Before the real breathing algorithm, a ‘static’ version of the skin triangulation is constructed. The reason is we can first test some operations on the skin surface which is used in the dynamic version. We will talk about the static version first. The main steps in the static version are:

1. Place points on the surface according to the density
2. Construct the triangulation of the skin surface by the point set.
3. Refine the triangulation by edge contraction and point insertion

**Point Distribution** The goal is generating points on the surface with a local density proportional to the local curvature, \( \kappa(x) \). We will first define the weighted area of the skin surface. Then we will define a point density, \( c \), base on the weighted area. Having this constant, we are able to break down the skin surface within mixed cells and generate points independently within each mixed cell.
The *weighted area*, $W$, is the integration of all points on surface times the squared curvature of each of them:

$$W = \int \kappa^2(x) dA$$

Let's take the case for a unit sphere, each point on the sphere has the same curvature value, which is $1$. Thus the weighted area is $4\pi$. Note that the weighted area does not change for different radius. As the sphere radius changed to $r$, which makes the surface area changed in $r^2$, the curvature of each point decreases in the rate of $1/r^2$.

For the case of hyperboloid, we cannot take the whole hyperboloid since the area is infinite. So we just take a part of the hyperboloid, which is sufficient enough because all of the hyperboloids are clipped within mixed cells. We will take a 2-sheeted hyperboloid as an example. Let the hyperboloid be a revolution with a hyperbolic curve modeled by the Klein model. Let the points on the hyperboloid be $(u(p), v(p))$, s.t. $p \in [0,1]$. We have

$$K(p) = \begin{bmatrix} u(p) \\ v(p) \end{bmatrix} = \begin{bmatrix} \frac{p}{\sqrt{1-p^2}} \\ \frac{\sqrt{1-p^2}}{\sqrt{1-p^2}} \end{bmatrix}$$

$$\frac{\partial K}{\partial p} = \begin{bmatrix} \frac{\partial u}{\partial p} \\ \frac{\partial v}{\partial p} \end{bmatrix} = \begin{bmatrix} \frac{1}{\sqrt{(1-p^2)\sqrt{1-p^2}}} \\ \frac{-1}{\sqrt{(1-p^2)\sqrt{1-p^2}}} \end{bmatrix}$$

$$\vartheta^2(p) = u^2(p) + v^2(p) = \frac{1 + p^2}{1 - p^2}$$

With distance, $s$, the derivatives of $s$ related to $p$ is

$$ds^2 = du^2 + dv^2 = \frac{1 + p^2}{(1 - p^2)^3} dp^2$$

Let $p_{max}$ be the value of $p$ such that $v(p_{max})$ is the maximum height of the hyperboloid. Now the weighted area is

$$r(p) = u(p) = \frac{p}{\sqrt{1-p^2}}$$

$$W(p_{max}) = \int_0^{p_{max}} \frac{1}{\vartheta^2(p)} dA$$

$$= 2\pi \int_0^{p_{max}} \frac{r(p)}{\vartheta^2(p)} ds$$

$$= 2\pi \int_0^{p_{max}} \frac{p}{(1-p^2)^{3/2}} dp$$

$$= \frac{\sqrt{2\pi}}{2} \left( \arctanh \frac{1 + p}{\sqrt{2(1+p^2)}} \right)_{0}^{p_{max}}$$

With weighted area of the surface, we define the constant, $c$, which is the number of points per unit weighted area. According to the size constraint of edge and triangle that will be discussed later, $c = 16$ is a feasible solution. With this constant, we can generate points independently for each mixed cell according to curvature, see figure 11.

For sphere case, we generate $c \cdot 4\pi$ points and keep them if they are within the mixed cell. For hyperboloids, weighted area will be computed within the portion with top and bottom clipped. Then, with $c$, points are generated and being kept if they are within the mixed cell.

Figure 10: A hyperboloid rotated with the radius $u(p)$
Triangulation The first step is constructing the Delunay triangulation with the point set generated. According to the Voronoi edge lemma, a triangle is in the restricted Delunay triangulation iff the Voronoi edge intersect with the skin on a point. This implies the corresponding Voronoi center of the tetrahedron with that triangle is inside the skin body. So, after we compute the Delunay triangulation, we keep those tetrahedra with their centers within the skin body. Then we just take the boundary of these tetrahedra to get the skin triangulation.

Refinement The goal of the algorithm is to locally triangulate with edges and triangles of size roughly proportional to the length scale. The size of an edge $ab$ is defined to be half of its length, $R_{ab} = |ab|/2$. The size of a triangle $abc$ is the radius of the circumcircle and denoted as $R_{abc}$. We want to achieve

[E] $R_{ab} > \frac{C}{Q} \cdot \rho_{ab}$ for every edge $ab \in D$.

[T] $R_{abc} < C Q \cdot \rho_{abc}$ for every triangle $abc \in D$.

which $\rho_{ab} = \max\{\varrho(a), \varrho(b)\}$ and $\rho_{abc} = \min\{\varrho(a), \varrho(b), \varrho(c)\}$.

For edges we are mostly concerned with not letting them get too short and for triangle we worry about growing them too large. For constant $C$ and $Q$, $C$ controls how closely the triangulation approximates the skin surfaces roughly. $Q$ controls the quality of the triangles. $C = \frac{1}{Q}$ and $Q = 2$ is a feasible assignment. So the algorithm will enforce both invariants by contracting short edges and adding dual vertices of large triangles.

```
loop
  while \exists edge ab with $R_{ab} \leq \frac{C}{Q} \cdot \rho_{ab}$ do
    contract ab
  endwhile;
  while \exists triangle abc with $R_{abc} \geq C Q \cdot \rho_{abc}$ do
    add dual Voronoi vertex
  endwhile
forever.
```

7 Surface Maintenance

Because the skin can deform with the time, we want to design the algorithm so that 'the algorithm breathes'. That means it maintains a triangulation dynamically by adapting density to curvature, geometric location to shape, and connectivity to topology. So it can construct the skin triangulation by starting with the empty triangulation and growing components from nothing, or deform from one skin to another skin.

Let $t < u$ be moments in time and $D(t)$ the restricted Delaunay triangulation at time $t$. The algorithm updates $D(t)$ locally and changes it into $D(u)$. This section describes the overall algorithm and presents the details for adapting the triangulation to the changing shape of the surface.

Moving vertices. The intuition for moving vertices is taken from Morse theory, which considers differential structures that arise in sweeping a smooth manifold $M$, see e.g. Milnor [7]. The skin surface is the cross-section at a moment in time during the sweep, and the manifold is the stack of cross-sections in the time direction. A topological change in the cross-section corresponds to a critical point of $M$. For critical points in a time interval $[t_0, t_1]$ free of critical points, we can construct a 1-parameter family of diffeomorphisms from the integral lines of the
ordinary differential equation defined by the gradient flow \( \nabla_t M = \partial M/\partial t \). These diffeomorphisms \( \varphi_t : N(t_0) \to N(t) \), with \( t \in [t_0, t_1] \), can be composed to diffeomorphically connect any two cross-sections in the time interval,

\[
\varphi_u = \varphi_u \circ \varphi^{-1}_t : N(t) \to N(u).
\]

To do the step from time \( t \) to \( u \), we assume we can compute integral lines and move a vertex \( a \in D(t) \) along that line to \( a' = \varphi_u(a) \in D(u) \). For the canonical deformation between two skin surfaces discussed in [2], integral lines can indeed be computed. Note however that moving vertices along integral lines is convenient but not essential for the algorithm, and an approximation of that movement will in general suffice. For small time steps, the triangulation changes only a small amount and can be maintained as described in Section 6.

**Algorithmic time warp.** Moving the entire triangulation forward by one time step can be expensive, because we need to adjust the coordinates of every single vertex. This is a waste of effort in cases where the time step is forced to be small at a few vertices while most others could afford much larger steps. We thus follow an alternative approach and take different time steps at different locations. This is done by prioritizing the local transformations that change the triangulation in time. There are three types pertaining to the three adaptation tasks. We save effort by adjusting the coordinates only for the vertices involved in a transformation. This results in a time-warped surface with different pieces reflecting the state at different times. To bring the entire surface to the present time \( t \), we simply adjust all the vertex coordinates, and by assumed correctness of the prioritization this requires no other changes in the triangulation.

A any moment in time \( t \), we consider the collection of possible next transformations. Let \( t_i > t \) be the time such a transformation \( \tau_i \) would happen if the vertices moved along integral lines and no other transformations preceded \( \tau_i \). We store the \( \tau_i \) in a priority queue ordered by \( t_i \). The deformation ends at time 1.

\begin{verbatim}
loop \( \tau_i = \text{NextTransformation};
  if \( t_i < 1 \) then \( D = \text{Apply}(\tau_i) \) else stop endif forever.
\end{verbatim}

Function \text{Apply} changes \( D \) according to \( \tau_i \), and simultaneously updates the priority queue by inserting new transformations made possible by the changes caused by \( \tau_i \). The changes may make some of the transformations in the priority queue inapplicable. For example, the edge of an edge flip may disappear from \( D \). Instead of deleting these transformations immediately, we use a lazy strategy that checks a transformation when it reaches the top of the priority queue.

\begin{verbatim}
Transformation \text{NextTransformation};
  repeat \( \tau = \text{ExtractMin} \) until \( \text{IsOK}(\tau) \);
  return \( \tau \).
\end{verbatim}

**Topological Changes.** When the surface moves, the curvature will eventually increase and approach to infinity in certain portion of the skin. When there is a topology change, the curvature will becomes infinity at the critical point, such as the time when a vertex just appear, and the skin body of it is a sphere with radius zero. Take the example of a hyperbola, it will begin with a two-sheeted hyperboloid. The two pieces will move towards each other until they become a double cone, then become an 1-sheeted hyperboloid afterwards. The curvature will be extremely high.

![Figure 12: Curvature of a hyperbola increase at the tip as the skin body is close to a topological change.](image)
high at certain moment, so large number of points are needed to adapt that curvature. The disadvantages are, first, usually these area are growing smaller and smaller and not very significant to the whole skin surface. The second reason is such large number of points is a pain to the computation and may even be uncontrollable and crash the program. So we introduce the notion of hot spheres. We place very small spheres at each of centers of the mixed cells. A 2-dimensional diagram is shown in figure 13. Points entering the hot spheres have curvature higher than certain value. Instead of maintaining it with the refinement, we dropped those point and reconstruct it with a double cone, or a 'cylinder'. We proved the restricted Delunay triangulation will still be homeomorphic to the skin surface by this special sampling, with the condition that the special sampling maintain enough points at the boundary between the special hot portion and the rest of the skin outside the hot ball.

**PART III. SOFTWARE**

**8 Data Structure and Algorithm**

The software is under development. It is based on the Delunay triangulation code by Raindrop Geomagic. In order to construct the skin surface, several functionalities had been added to the code. We first modified it to be able to compute the weighted Delunay triangulation. Moreover, two addition data structures are added. COMPDB (Complex Database) compute information we needed on the Delunay complex, like centers of the mixed cells. SKINMESH compute the skin surface using the COMPDB data structures. In the future, we will also add refinement and addaptive meshing onto the software.

**Weighted Delunay Triangulation** Two features are added to change the code into weighted case, the local Delunay test (INSPHERE()) and 4:1-flip (FLIPVERTEX()).

In unweighted case, we have a triangle $uvw$ with incident tetrahedra $uvwx$ and $uwxy$. It is locally Delunay if $y^+$ is above the hyperplane through $u^+, v^+, w^+$ and $x^+$, which $x^+$ is the lifting of the point $x$ to $\mathbb{R}^4$ as $(x_1, x_2, x_3, x_1^2 + x_2^2 + x_3^2)$. So an orientation test can be used to determine on which side of this plane $y^+$ lies and so is equivalent to the local Delunay test. So a determine

$$
\begin{vmatrix}
  u_1 & u_2 & u_3 & u_1^2 + u_2^2 + u_3^2 & 1 \\
  v_1 & v_2 & v_3 & v_1^2 + v_2^2 + v_3^2 & 1 \\
  w_1 & w_2 & w_3 & w_1^2 + w_2^2 + w_3^2 & 1 \\
  x_1 & x_2 & x_3 & x_1^2 + x_2^2 + x_3^2 & 1 \\
  y_1 & y_2 & y_3 & y_1^2 + y_2^2 + y_3^2 & 1 \\
\end{vmatrix} = 0
$$

is used to determine whether the triangle is locally Delunay. In weighted case, we modify it into a diff-

![Figure 13: Dotted Voronoi diagram, dashed Delaunay triangulation, solid mixed complex, hollow centers, and shaded hot portion of space.](image)

![Figure 14: Head on view of start, middle, end configuration generated by special sampling strategy taking a 2-sheeted to a 1-sheeted hyperboloid.](image)
ferent lifting including the weight:

\[
\begin{align*}
    u_1 & \quad u_2 & \quad u_3 & \quad 1 & \quad u_1^2 + u_2^2 + u_3^2 - r_u \\
    v_1 & \quad v_2 & \quad v_3 & \quad 1 & \quad v_1^2 + v_2^2 + v_3^2 - r_v \\
    w_1 & \quad w_2 & \quad w_3 & \quad 1 & \quad w_1^2 + w_2^2 + w_3^2 - r_w \\
    x_1 & \quad x_2 & \quad x_3 & \quad 1 & \quad x_1^2 + x_2^2 + x_3^2 - r_x \\
    y_1 & \quad y_2 & \quad y_3 & \quad 1 & \quad y_1^2 + y_2^2 + y_3^2 - r_y
\end{align*}
\]

which \( r_x \) is the weight of \( x \) and so as \( u, v, w, y \).

Another modification is the 4:1-flipping which does not occur in unweighted cases. Such case happened when a vertex is redundant. Figure 15 shows the decision tree for the four types of flipping. Tests are taken in this order because of minimization of geometric tests. Beginning with a triangle \( abc \) which is non-locally delone, we first test if the 4:1-flip is possible. This test is purely done without geometric tests, i.e. only need to check if the 3 edges of triangle \( abc \) is degree three. Following is the edge flip test, which first make sure if the edge is degree three, before we try to perform a geometric test making sure the edge is flippable. If all tests failed, we try to test if all three edges are convex, and attempt to make a 2:3-flip. There will be a chance that this triangle is not currently flippable and we mark it red and try to flip it later.

**COMPDB** There are two main goals of this data structure, enumerate all simplices and compute information for later use in skin construction. For each of \( d \)-dimensional simplices, an unique index is given in \([1..n_d]\), which \( n_d \) is the number of simplices in that dimension. For each dimension we create an array for storage of information, which are only centers at this point of time. The indices of these records are inserted into the TRIST data structure so for each triangle, we have 3 indices of edges and 2 (or 1) tetrahedron indices. Vertices and triangle indices do not need separate organization because the TRIST data structure had already handled them. So, the access time for the records will be constant but the preparation time for this data will be \( O(n) \) which \( n \) is the number of all simplices.

**SKINMESH** We first generated the points on the skin surface, then construct the skin mesh. For generation of points, we generate points within each mixed cell and clipped within it. Then we perform the Delunay triangulation on the points and mark those tetrahedra which have their centers within the skin body, through the function \( \text{INBODY()} \). The \( \text{INBODY()} \) function is a general function for testing whether a point inside the skin body by local search. Then we find the boundaries of the union of these tetrahedra by deleting triangles. Triangles are qualified to stay if

1. It has exactly two tetrahedra cofaces which one of them having its center in skin body and the other does not, or..
2. The triangle is on the boundary of the convex hull and the only tetrahedron

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{flipping_sequence.png}
\caption{Flipping sequence}
\end{figure}
coface is inside the skin body.

The remaining mesh is stored as TRIST data structure.

9 Performance

Time Here is some examples run by the current software. Computation time is measured and compared with Wrap. We save the points and perform the 'wrap' operation on Geomagic Wrap. Time of generating points in skin software is subtracted in order to make a comparison. Note that Wrap cannot exactly compute the skin surface, see figure 20 and 21. Moreover, Wrap will not care about voids inside the skin body. Experiments are ran on a Pentium 450MHz with 256MB RAM.

Figure 16: Body Center Cube. Number of points: 3415 Skin Time: 20s Wrap Time: 11s.

Figure 17: Number of points: 3470 Skin Time: 19s Wrap Time: 9s.

Figure 18: Number of points: 4800 Skin Time: 20s Wrap Time: 12s.

Figure 19: Gramicidin A Molecule. Number of points: 43145 Skin Time: 5min 6s Wrap Time: 2min 50s.

Triangle Quality Without refinement, triangle quality cannot be guaranteed, such as slim triangles, see figure 21. Moreover, points distribution are not exactly proportional to the curvature. It will soon be corrected by the refinement of edge contraction and point insertion algorithm.
REFINEMENT. Edge contraction and point insertion in maintaining the skin mesh according to condition (I), [E] and [T]

BREATHING SKIN ALGORITHM. Construct and maintain the skin surface dynamically. A priority queue will be used to handle the events of modification on the fly.

TOPOLOGICAL CHANGES. Dealing with triangles after they entered the hot spheres.

DATA COLLECTION Data such as time of computation and triangle qualities are collected as a comparison.

Further more, we would like to use the skin surface for some applications, such as molecular model and the mold filling process.

Figure 20: Portion of Gramicidin A molecule enlarged, generated by Wrap

Figure 21: Portion of Gramicidin A molecule enlarged, the skin surface after "push deep" operation on Wrap

10 Future Works

Here is a list of items to be finished:
References


Appendix A: Figures of Lemmas

Figure 22: A map for the proof of GHT.

Appendix B: Tables of Lemmas

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SP</td>
<td>Sandwich Property</td>
</tr>
<tr>
<td>ICL</td>
<td>Iso-Curvature Lemma</td>
</tr>
<tr>
<td>CSL</td>
<td>Curvature Sandwich Lemma</td>
</tr>
<tr>
<td>CVL</td>
<td>Curvature Variation Lemma</td>
</tr>
<tr>
<td>NVL</td>
<td>Normal Variation Lemma</td>
</tr>
<tr>
<td>ENL</td>
<td>Edge Normal Lemma</td>
</tr>
<tr>
<td>TNL</td>
<td>Triangle Normal Lemma</td>
</tr>
<tr>
<td>SDC</td>
<td>Short Distance Claim</td>
</tr>
<tr>
<td>LDC</td>
<td>Long Distance Claim</td>
</tr>
<tr>
<td>(I)</td>
<td>Condition on $\epsilon$</td>
</tr>
<tr>
<td>[E]</td>
<td>Invariant on edge length</td>
</tr>
<tr>
<td>[T]</td>
<td>Invariant on triangle size</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NSEL</td>
<td>No Short Edge Lemma</td>
</tr>
<tr>
<td>SL</td>
<td>Sampling Lemma</td>
</tr>
<tr>
<td>VEL</td>
<td>Voronoi Edge Lemma</td>
</tr>
<tr>
<td>VGL</td>
<td>Voronoi Polygon Lemma</td>
</tr>
<tr>
<td>VHL</td>
<td>Voronoi Polyhedron Lemma</td>
</tr>
<tr>
<td>GHT</td>
<td>General Homeomorphism Theorem</td>
</tr>
</tbody>
</table>

Table 1: Dictionary of properties, claims, lemmas, theorems, invariants, conditions.