# Chapter 4

# Surface Reconstruction

In the previous chapter we learned that the restricted Delaunay triangulation is a good approximation of a densely sampled surface $\Sigma$ from both topological and geometric view point. Unfortunately, we cannot compute this triangulation as the restricted Voronoi diagram $\mathrm{Vor}\, P|_\Sigma$ cannot be computed without knowing $\Sigma$. As a remedy we approximate the restricted Voronoi diagram and compute a set of triangles that include all restricted Delaunay triangles and may be some more. This set is pruned to extract a manifold surface which is output as an approximation to the sampled surface $\Sigma$.

## 4.1 Algorithm

First we observe that each restricted Voronoi cell $V_p|_\Sigma$ is almost flat if the sampling density is sufficiently high. This follows from the Normal Variation Lemma (3.3) as the points in $V_p|_\Sigma$ cannot be far apart if $\varepsilon$ is small. In particular, $V_p|_\Sigma$ lies within a thin neighborhood of the tangent plane $\tau_p$ at $p$. So, we need two approximations, (i) an approximation to $\tau_p$ or equivalently to $\mathbf{n}_p$, (ii) an approximation to $V_p|_\Sigma$ based on the approximation to $\mathbf{n}_p$. The following definitions of *poles* and *cocones* are used for these two approximations.

### 4.1.1 Poles and Cocones

**Definition 4.1 (Poles.)** *The farthest Voronoi vertex, denoted $p^+$, in $V_p$ is called the positive* pole *of $p$. The negative pole of $p$ is the farthest point $p^- \in V_p$ from $p$ so that the two vectors from $p$ to $p^+$ and $p^-$ make an angle more than $\frac{\pi}{2}$. We call $\mathbf{v}_p = p^+ - p$, the* pole vector *for $p$. If $V_p$ is unbounded,*

*$p^+$ is taken at infinity, and the direction of $\mathbf{v}_p$ is taken as the average of all directions given by the unbounded Voronoi edges.*

The following lemma is a direct consequence of Normal Lemma (3.2). It says that the pole vectors approximate the true normals at the sample points.

**Lemma 4.1 (Pole.)** *For $\varepsilon < 1$, the angle between the normal $\mathbf{n}_p$ at $p$ and the pole vector $\mathbf{v}_p$ satisfies the inequality:*

$$\angle_a(\mathbf{n}_p, \mathbf{v}_p) \leq 2 \arcsin \frac{\varepsilon}{1-\varepsilon}.$$

PROOF. First, consider the case where $V_p$ is bounded. Since the Voronoi cell $V_p$ contains the centers of the medial balls at $p$, we have $\|p^+ - p\| \geq f(p)$. Thus, plugging $\mu = 1$ in the Normal Lemma (3.2) we obtain the result immediately.

Next, consider the case where $V_p$ is unbounded. In this case $\mathbf{v}_p$ is computed as the average of the directions of the infinite Voronoi edges. The angle $\angle_a(\mathbf{v}_p, \mathbf{n}_p)$ in this case cannot be more than the worst angle made by an infinite Voronoi edge with $\mathbf{n}_p$. An infinite Voronoi edge $e$ makes the same angle with $\mathbf{n}_p$ as the vector $p\vec{p}_\infty$ does, where the infinite endpoint of $e$ is taken at $p_\infty$. Again we have $\|p - p_\infty\| \geq f(p)$ and Normal Lemma (3.2) can be applied with $\mu = 1$ to give the result. □

The Pole Lemma (4.1) says that the pole vector approximates the normal $\mathbf{n}_p$. Thus, the plane $\tilde{\tau}_p$ passing through $p$ and with the pole vector as normal approximates the tangent plane $\tau_p$. The following definition of *cocone* accommodates a thin neighborhood around $\tilde{\tau}_p$ to account for the small uncertainty in the estimation of $\mathbf{n}_p$.

**Definition 4.2 (Cocone.)** *The set $C_p = \{y \in V_p \ : \ \angle_a(p\vec{y}, \mathbf{v_p}) \geq \frac{3\pi}{8}\}$ is called the cocone of $p$. In words, $C_p$ is the complement of a double cone that is clipped within $V_p$. This double cone has $p$ as the apex and the pole vector $\mathbf{v}_p$ as the axis and an opening angle of $\frac{3\pi}{8}$ with the axis. See Figure 4.1 for an example of a cocone.*

As an approximation to $V_p|_\Sigma$, cocones meet all Voronoi edges that are intersected by $\Sigma$. So, if we compute all triangles dual to the Voronoi edges

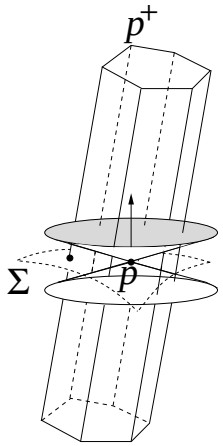


Figure 4.1: The positive pole $p^+$ helps estimating the normal $\mathbf{n}_p$ and the cocone helps estimating $V_p|_\Sigma$.

intersected by cocones, we obtain all restricted Delaunay triangles and possibly a few others. We call this set of triangles *cocone triangles*. We will see later that all cocone triangles lie very close to $\Sigma$. A cleaning step is necessary to weed out some triangles from the set of cocone triangles so that a 2-manifold is computed as output. This is accomplished by a *manifold extraction* step.

COCONE($P$)
1 Compute Vor $P$;
2 $T = \emptyset$;
3 `for` each $p \in P$ `do`
4 $T_p$ =COCONETRIANGLES($V_p$);
5 $T := T \cup T_p$
6 `endfor`
7 $E$ :=EXTRACTMANIFOLD($T$)
8 `return` $E$.

Let us now look into the details of the two steps COCONETRIANGLES() and EXTRACTMANIFOLD().

In order to check if a Voronoi edge $e = (a, b)$ intersects $C_p$ we consider the three vectors $\mathbf{v}_p$, $\mathbf{a} = (a - p)$, $\mathbf{b} = (b - p)$, and two conditions I and II:

$$I.\ \ 0 \leq \frac{\mathbf{v}_p \cdot \mathbf{a}}{\|\mathbf{v}_p\|\|\mathbf{a}\|} \leq \cos\frac{3\pi}{8} \text{ or } 0 \leq \frac{\mathbf{v}_p \cdot \mathbf{b}}{\|\mathbf{v}_p\|\|\mathbf{b}\|} \leq \cos\frac{3\pi}{8},$$

$$II.\ \ \frac{\mathbf{v}_p \cdot \mathbf{a}}{\|\mathbf{v}_p\|\|\mathbf{a}\|} < 0 \text{ and } \frac{-\mathbf{v}_p \cdot \mathbf{b}}{\|\mathbf{v}_p\|\|\mathbf{b}\|} < 0.$$

Condition I checks if any of the vertices $a$ and $b$ of the Voronoi edge $e$ lies inside $C_p$. Condition II checks if both $a$ and $b$ lie outside $C_p$, but the edge $e$ crosses it.

CoconeTriangles($V_p$)
1 $T_p := \emptyset$
2 `for` each Voronoi edge $e = (a, b) \subset V_p$ `do`
3 `if` Condition I or II holds
4 $T_p = T_p \cup \text{dual}\, e$
5 `endif`
6 `endfor`
7 `return` $T_p$

The set $T = \cup T_p$ of cocone triangles enjoys some interesting geometric properties which we exploit in the manifold extraction step as well as in the proofs of geometric and topological guarantees of Cocone. Of course, the sample has to be sufficiently dense for these properties to hold. In the rest of the chapter we assume that $\varepsilon \leq 0.06$ which satisfies Condition A stated in chapter 3 enabling us to apply the results therein.

### 4.1.2 Cocone triangles

First we show that all triangles in $T$ have a small empty ball circumscribing it, i.e., the radius of this ball is small compared to the local feature sizes at their vertices. This also means that their diametric ball is small. This fact together with Triangle Normal Lemma (3.5) implies that all cocone triangles lie almost flat to the surface.

**Lemma 4.2 (Small Triangle.)** *Let $t$ be any cocone triangle and $r$ denote the radius of the smallest empty ball circumscribing $t$. Then, for each vertex $p$ of $t$ and $\varepsilon \leq 0.06$,*

*(i) $r \leq \frac{1.15\varepsilon}{1-\varepsilon} f(p)$, and*

*(ii) circumradius of $t$ is at most $\frac{1.15\varepsilon}{1-\varepsilon} f(p)$.*

PROOF. Let $y$ be any point in $V_p$ so that

$$\angle_a(\mathbf{n}_p, \vec{py}) > \frac{3\pi}{8} - 2\arcsin\frac{\varepsilon}{1-\varepsilon}. \tag{4.1}$$

First we claim that, for any such point $y$, we have $\|y-p\| < \frac{1.15\,\varepsilon}{1-\varepsilon} f(p)$ if $\varepsilon \leq 0.06$.

If $\angle_a(\mathbf{n}_p, \vec{py}) > \theta = \arcsin\frac{\varepsilon}{\mu(1-\varepsilon)} + \arcsin\frac{\varepsilon}{1-\varepsilon}$, then $\|y-p\| \leq \mu f(p)$ according to Normal Lemma (3.2). With $\mu = \frac{1.15\varepsilon}{1-\varepsilon}$ and $\varepsilon \leq 0.06$ we have

$$\theta \;=\; \arcsin\frac{1}{1.15} + \arcsin\frac{\varepsilon}{1-\varepsilon} < \frac{3\pi}{8} - 2\arcsin\frac{\varepsilon}{1-\varepsilon}. \tag{4.2}$$

Thus, from inequalities 4.1 and 4.2 we have

$$\angle_a(\mathbf{n}_p, \vec{py}) \;>\; \frac{3\pi}{8} - 2\arcsin\frac{\varepsilon}{1-\varepsilon} > \theta. \tag{4.3}$$

Therefore, $\|y-p\| \leq \frac{1.15\varepsilon}{1-\varepsilon} f(p)$.

Now let $t$ be any cocone triangle with $p$ being any of its vertices and $e = \mathrm{dual}\, t$ being its dual Voronoi edge. For $t$ to be a cocone triangle, it is necessary that there is a point $y \in e$ so that $\angle_a(\mathbf{v}_p, \vec{py}) > \frac{3\pi}{8}$. Taking into account the angle $\angle_a(\mathbf{v}_p, \mathbf{n}_p)$, this necessary condition implies $\angle_a(\mathbf{n}_p, \vec{py}) > \frac{3\pi}{8} - 2\arcsin\frac{\varepsilon}{1-\varepsilon}$ which satisfies the inequality 4.3. Hence, the previous claim gives $\|y-p\| \leq \frac{1.15\,\varepsilon}{1-\varepsilon} f(p)$ for $\varepsilon \leq 0.06$.

The ball $B_{y,\|y-p\|}$ is empty and circumscribes $t$ proving (i). The claim in (ii) follows immediately from (i) as the circumradius of $t$ cannot be larger than the radius of any ball circumscribing it. □

The next lemma proves that all cocone triangles lie almost parallel to the surface. The angle bounds are expressed in terms of $\alpha(\varepsilon)$ and $\beta(\varepsilon)$ that are defined in chapter 3.

**Lemma 4.3 (Cocone Triangle Normal.)** *Let $t$ be a cocone triangle and $\mathbf{n}_t$ be its normal. For any vertex $p$ of $t$ we have $\angle_a(\mathbf{n}_p, \mathbf{n}_t) \leq \alpha(\frac{2.3\varepsilon}{1-\varepsilon}) + \beta(\frac{1.15\varepsilon}{1-\varepsilon})$ when $\varepsilon \leq 0.06$.*

PROOF. Let $q$ be a vertex of $t$ with a maximal angle of $t$. The circumradius of $t$ is at most $\frac{1.15\varepsilon}{1-\varepsilon} f(q)$ by Small Triangle Lemma (4.2). Then, by Triangle Normal Lemma (3.5),

$$\angle_a(\mathbf{n}_q, \mathbf{n}_t) \leq \beta(\frac{1.15\varepsilon}{1-\varepsilon}) \text{ for } \varepsilon \leq 0.06.$$

The distance between $p$ and $q$ is no more than the diameter of the circumcircle of $t$, i.e., $\|p-q\| \leq \frac{2.3\varepsilon}{1-\varepsilon} f(p)$ (Small Triangle Lemma (4.2)). By Normal Variation Lemma (3.3), $\angle(\mathbf{n}_p, \mathbf{n}_q) \leq \alpha(\frac{2.3\varepsilon}{1-\varepsilon})$. The desired bound for $\angle_a(\mathbf{n}_p, \mathbf{n}_t)$ follows since it is no more than the sum $\angle(\mathbf{n}_p, \mathbf{n}_q) + \angle_a(\mathbf{n}_q, \mathbf{n}_t)$. □

### 4.1.3 Pruning

Prior to the extraction of a 2-manifold from the set of cocone triangles, some of them are pruned. An edge $e$ is *sharp* if any two consecutive cocone triangles around it form an angle more than $\frac{3\pi}{2}$. See Figure 4.2. Edges with a single triangle incident to them are also sharp by this definition. Also, we will show later that the cocone triangles include all restricted Delaunay triangles for a sufficiently dense sample. The set of restricted Delaunay triangles cannot be incident to sharp edges. This implies that we can prune triangles incident to sharp edges and still retain the set of restricted Delaunay triangles. In fact, we can carry out this pruning in a cascaded manner. By deleting one triangle incident to a sharp edge, we may create other sharp edges. Since no restricted Delaunay triangle is pruned, none of their edges become sharp. Therefore, it is safe to delete the new sharp edges with all of their incident triangles.

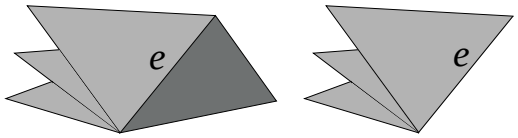


Figure 4.2: The edge $e$ is not sharp in the left picture. It is sharp in the right picture.

This pruning step weeds out all triangles incident to sharp edges, but the remaining triangles still may not form a surface. They may form "layers" creating a non-manifold. A manifold surface is extracted from this possibly layered set by *walking* outside the space covered by them. The manifold extraction step depends heavily on the fact that cocone triangles contain all restricted Delaunay triangles none of whose edges is sharp. We prove this fact below.

**Theorem 4.1 (Restricted Delaunay.)** *For $\varepsilon \leq 0.06$, the following conditions hold:*

*(i) cocone triangles contain all restricted Delaunay triangles, and*

*(ii) no restricted Delaunay triangle has a sharp edge.*

PROOF. Consider (i). Let $y$ be any point in any restricted Voronoi cell $V_p|_\Sigma$. We claim that $\angle_a \mathbf{n}_p, \vec{py}$ is larger than $\frac{\pi}{2} - \arcsin\frac{\varepsilon}{2(1-\varepsilon)}$. The distance $\|y-p\| \leq \varepsilon f(y)$ since $y \in V_{p,\Sigma}$ and $P$ is an $\varepsilon$-sample. By Feature Translation Lemma (1.3) $\|y-p\| \leq \frac{\varepsilon}{1-\varepsilon} f(p)$. We can therefore apply the proof of Edge Normal Lemma (3.4) to establish the claim.

Let $t$ be any restricted Delaunay triangle and $e = \text{dual}\, t$ be the dual Voronoi edge. Consider the point $y = e \cap \Sigma$. We have $y \in V_p|_\Sigma$ for each of the three points $p \in P$ determining $e$. For each such $p$, the angle $\angle_a(\mathbf{n}_p, \vec{py})$ is larger than $\pi/2 - \arcsin\frac{\varepsilon}{2(1-\varepsilon)}$. Therefore

$$\begin{aligned} \angle_a(\vec{py}, \mathbf{v}_p) &> \angle_a(\vec{py}, \mathbf{n}_p) - \angle_a(\mathbf{n}_p, \mathbf{v}_p) \\ &> \frac{\pi}{2} - \arcsin\frac{\varepsilon}{2(1-\varepsilon)} - \theta \end{aligned}$$

where $\theta = \angle_a(\mathbf{n}_p, \mathbf{v}_p)$. By Pole Lemma (4.1) we have

$$\begin{aligned} \theta + \arcsin\frac{\varepsilon}{2(1-\varepsilon)} &\leq 2\arcsin\frac{\varepsilon}{1-\varepsilon} + \arcsin\frac{\varepsilon}{2(1-\varepsilon)} \\ &\leq \frac{\pi}{8} \text{ for } \varepsilon \leq 0.06. \end{aligned}$$

So, $\angle_a(\vec{py}, \mathbf{v}_p) > \frac{3\pi}{8}$. Therefore, the point $y$ is in the cocone $C_p$ by definition. Hence the triangle $t$ is a cocone triangle.

Consider (ii). Let $t_1$ and $t_2$ be adjacent triangles in the restricted Delaunay triangulation with $e$ as their shared edge, and let $p \in e$ be any of their shared vertices. Since $t_1$ and $t_2$ belong to the restricted Delaunay triangulation, they have circumscribing balls $B_1$ and $B_2$, respectively, centered at points $v_1, v_2$ of $\Sigma$.

The boundaries of $B_1$ and $B_2$ intersect in a circle $C$ contained in a plane $H$, with $e \subset H$. The plane $H$ separates $t_1$ and $t_2$, since the third vertex of each triangle lies on the boundary of its circumscribing ball, and $B_1 \subseteq B_2$ on one side of $H$, while $B_2 \subseteq B_1$ on the other. See Figure 4.3. The line through $v_1, v_2$ is perpendicular to $H$. Both $v_1$ and $v_2$ belong to the Voronoi facet dual to $e$. This means $v_1$ and $v_2$ belong to a restricted Voronoi cell, and the distance $\|v_1 - v_2\| \leq \frac{2\varepsilon}{(1-\varepsilon)} f(v_1)$ by Short Distance Lemma (3.6). So the segment $v_1v_2$ forms an angle of at least $\pi/2 - \arcsin\frac{\varepsilon}{1-\varepsilon}$ with $\mathbf{n}_{v_1}$ (proof of Edge Normal Lemma (3.4)). This normal differs, in turn, from

$\mathbf{n}_p$ by an angle of at most $\frac{\varepsilon}{1-3\varepsilon}$ (Normal Variation Lemma (3.3)). So, the angle between $H$ and $\mathbf{n}_p$ is at most $\frac{\varepsilon}{1-3\varepsilon} + \arcsin\frac{\varepsilon}{1-\varepsilon}$. For small $\varepsilon$, they are nearly parallel. In particular, if $\varepsilon \leq 0.06$, $H$ makes at most $9°$ with $\mathbf{n}_p$. Similarly, plugging $\varepsilon \leq 0.06$ in the angle upper bound of Cocone Triangle Normal Lemma (4.3), one gets that both $t_1$ and $t_2$ differ from the surface normal at $p$ by at most $14°$.

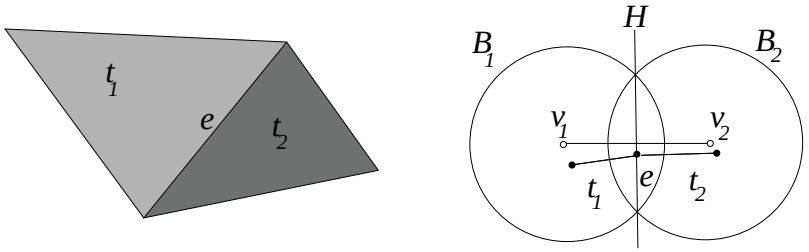


Figure 4.3: Illustration for Restricted Delaunay Theorem.

Thus we have $t_1$ on one side of $H$, $t_2$ on the other, and the smaller angle between $H$ and either triangle is at least $67°$. Hence the smaller angle between $t_1$ and $t_2$ is at least $134°$, and $e$ is not sharp. □

### 4.1.4 Manifold extraction

A simplicial complex with an underlying space of a 2-manifold is extracted out of the pruned set of cocone triangles. Let $\Sigma' \subseteq \Sigma$ be any connected component of the sampled surface. Since cocone triangles are small (Small Triangle Lemma 4.2), they cannot join points from different components of $\Sigma$. Let $T'$ be the pruned set of cocone triangles with vertices in $\Sigma'$. Consider the medial axis of $\Sigma'$. The triangles of $T'$ lie much closer to $\Sigma'$ than to its medial axis. Furthermore, $T'$ includes the restricted Delaunay triangulation $\mathrm{Del}\, P|_{\Sigma'}$ (Restricted Delaunay Theorem 4.1). Therefore, if $|T'|$ denotes the underlying space of $T'$, the space $\mathbb{R}^3 \setminus |T'|$ has precisely two disjoint open sets $O_{in}$ and $O_{out}$ containing the inner and outer medial axis of $\Sigma'$ respectively. The manifold extraction step computes the boundary of the closure of $O_{out}$, which we simply refer to as the boundary of $O_{out}$.

Let $E'$ be the boundary of $O_{out}$. We claim that $E'$ is a 2-manifold. Let $p$ be any vertex of $E'$. Orient the normal $\mathbf{n}_p$ so that it points towards $O_{out}$. Consider a sufficiently small ball $B$ centering $p$. Call the point where the ray of $\mathbf{n}_p$ intersects the boundary of $B$ the *north pole*. Obviously the north pole is in $O_{out}$. Let $T_p$ denote the set of triangles in $T'$ that are visible from the

north pole within $B$. The triangles of $T_p$ are in the boundary of $O_{out}$. Since there is no sharp edge in $T'$, the set of triangles $T_p$ makes a topological disk. We argue that $T_p$ is the only set of triangles in the boundary of $O_{out}$ which are incident to $p$.

Let $q \neq p$ be a vertex of a triangle $t \in T_p$. The triangle $t$ is also in $T_q$. If not, the line of the normal $\mathbf{n}_p$, when moved parallelly through the edge $pq$ towards $q$, must hit an edge in $T'$ that is sharp. The assumption to this claim is that the normals $\mathbf{n}_p$ and $\mathbf{n}_q$ are almost parallel and hence the visibility directions at $p$ and $q$ are almost parallel. Since $T'$ does not have any sharp edge, $t$ is also in $T_q$. This means that all topological disks at the vertices of $E'$ are compatible and they form a 2-manifold. This 2-manifold separates $O_{out}$ from $T'$ implying that $E'$ cannot have any other triangles from $T'$ other than the ones in the topological disks described above.

We compute $E'$ from $T'$ as a collection of triangles by a depth first walk in the Delaunay triangulation $\mathrm{Del}\, P$. The walk starts with a seed triangle in $T'$. The routine SEED computes this seed triangle for each component $T'$ of the pruned set by another depth first walk in the Delaunay triangulation. At any generic step, SEED comes to a triangle $t$ via a tetrahedron $\sigma$ and performs the following steps. First, it checks if $t$ is a cocone triangle. If so, it checks if it belongs to a component $T'$ for which a seed has not yet been picked. If so, the pair $(\sigma, t)$, also called the *seed pair*, is put into the seed set. Then, it marks all triangles of $T'$ so that any subsequent check can identify that a seed for $T'$ has been picked. Next, it continues walking through the triangles and their adjacent tetrahedra in a depth first manner till another cocone triangle is hit. SEED starts the walk at any convex hull triangle $t$. If $t$ is a cocone triangle, it puts the pair $(\sigma, t)$ into a seed set where $\sigma$ is the infinite tetrahedron incident to $t$. The initiation of the walk from a convex hull triangle ensures that the first triangle encountered in a component is on the outside of that component or equivalently on the boundary of $O_{out}$ defined for that component. Assuming the function SEED, a high level description of EXTRACTMANIFOLD is given below.

```
EXTRACTMANIFOLD (T)
1  T := Pruned T;
2  SD := SEED(T);
3  for each tuple (σ, t) ∈ SD do
4     E':= SURFTRIANGLES(σ, t)
5     E:= E ∪ E'
6  endfor
7  return the simplicial complex of E.
```

The main task in EXTRACTMANIFOLD is done by SURFTRIANGLES that takes a seed pair $(\sigma, t)$ as input. First, we initialize the surface $E'$ with the seed triangle $t$ which is definitely in $E'$ (line 1). Next, we initialize a stack *Pending* with the triple $(\sigma, t, e)$ where $e$ is an edge of $t$ (lines 3 and 4). As long as the stack *Pending* is not empty, we pop its top element $(\sigma, t, e)$. If the edge $e$ is not already processed we call the function SURFACENEIGHBOR to compute a tetrahedron-triangle pair $(\sigma', t')$ (line 9). The tetrahedron $\sigma'$ is adjacent to $t'$ and intersects $O_{out}$ while $t'$ is in $E'$ and is adjacent to $t$ via $e$. The triangle $t'$ is inserted in $E'$. Then two new triples $(\sigma', t', e')$ are pushed on the stack *pending* for each edge $e' \neq e$ of $t'$ (lines 11 to 13). Finally we return $E'$ (line 16).

```
SURFTRIANGLES (σ, t)
 1  E' := {t}
 2  Pending := ∅
 3  Let e be any edge of t
 4  push (σ, t, e) on Pending.
 5  while Pending ≠ ∅
 6    pop (σ, t, e) from Pending
 7    if e is not marked processed
 8      mark e processed
 9      (σ', t') := SURFACENEIGHBOR (σ, t, e)
10      E' := E' ∪ {t'}
11      for each edge e' ≠ e of t' do
12        push (σ', t', e) on Pending.
13      endfor
14    endif
15  endwhile
16  return E'
```

The question is how to implement the function SURFACENEIGHBOR$(\sigma, t, e)$. It has to output a tuple $(\sigma', t')$ where $t'$ is the neighbor of $t$ on the surface given by $E'$ and $\sigma'$ is an adjacent tetrahedron intersecting $O_{out}$. One can compute the surface neighbor $t'$ of $t$ using some numerical computations involving some dot product computations of vectors. However, these computations often run into trouble due to numerical errors with finite precision arithmetics. In particular, triangles of certain types of flat tetrahedra called *slivers* tend to contribute to these numerical errors and slivers are not uncommon in the Delaunay triangulation of a sample from a surface.

A robust and faster implementation of the function SURFACENEIGHBOR avoids numerical computations by exploiting the combinatorial structure of the Delaunay triangulation. Every triangle in the Delaunay triangulation has two incident tetrahedra if we account for the infinite ones incident to the convex hull triangles. SURFACENEIGHBOR is called with a triple $(\sigma, t, e)$. It circles over the tetrahedra and triangles incident to the edge $e$ starting from $t$ and going towards the other triangle of $\sigma$ incident to $e$. This circular walk stops when another cocone triangle $t'$ is reached. If $t'$ is reached via the tetrahedron $\sigma'$, we output the pair $(\sigma', t')$. Assuming inductively that $\sigma$ intersects $O_{out}$, the tetrahedron $\sigma'$ also intersects $O_{out}$. For example, in Figure 4.4, SURFACENEIGHBOR is passed on the triple $(T_1, t, e)$ and then it circles through the tetrahedra $T_1, T_2, T_3$ and their triangles till it reaches $t'$. At this point it returns $(T_3, t')$ where both $T_1$ and $T_3$ lie outside, i.e., in $O_{out}$. The SURFTRIANGLES with this implementation of SURFACENEIGHBOR is robust since no numerical decisions are involved, see Figure 4.4. The latter is also the reason why it is fast provided the Delaunay triangulation is given in a form which allows to answer queries for neighboring tetrahedra quickly.

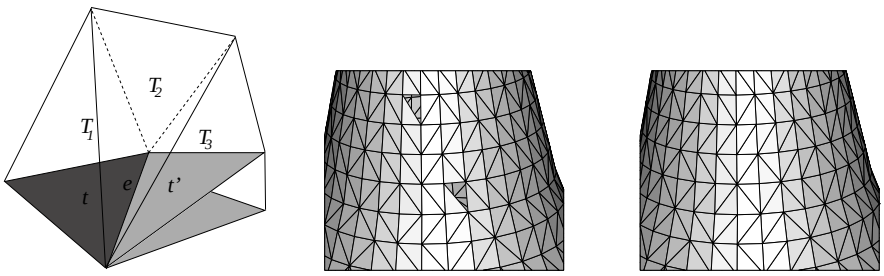


Figure 4.4: A stable computation of SURFACENEIGHBOR (left), a zoom on a reconstruction after an instable computation with numerical errors (middle), and a stable computation without any numerical error (right).