6

Watertight Reconstructions

Most of the surface reconstruction algorithms face a difficulty when dealing with undersampled surfaces and noise. While the algorithm described in Chapter 5 can detect undersampling, it leaves holes in the surface near the undersampled regions. Although this may be desirable for reconstructing surfaces with boundaries, many applications such as CAD designs require that the output surface be *watertight*, that is, a surface that bounds a solid. Ideally, this means that the watertight surface should be a compact 2-manifold without any boundary. The two algorithms that are described in this chapter produce these types of surfaces when the input sample is sufficiently dense. However, the algorithms are designed keeping in mind that the sample may not be sufficiently dense everywhere. So, in practice, the algorithms may not produce a perfect manifold surface but their output is watertight in the following sense:

> *Watertight surface*: A 2-complex embedded in $\mathbb{R}^3$ whose underlying space is a boundary of the closure of a 3-manifold in $\mathbb{R}^3$.

Notice that the above definition allows the watertight surface to be nonmanifold. The closure of a 3-manifold can indeed introduce nonmanifold property; for example, a surface pinched at a point can be in the closure of a 3-manifold.

6.1 Power Crust

In Chapter 4, we have seen that the poles for a dense point sample lie quite far away from all samples (proof of the Pole Lemma 4.1) and hence from the surface. Indeed, they lie close to the medial axis. The Delaunay balls circumscribing the tetrahedra that are dual to the poles are called *polar balls*. These balls have their centers at the poles and they approximate the medial balls. The

PowerCrust algorithm is based on the observation that a solid is equal to the union of all the inner (outer) medial balls of its bounding surface. Since the polar balls approximate the medial balls, the boundary of the union of inner (outer) polar balls approximates the bounding surface.

For simplicity and also for a technical reason we assume that the sampled surface Σ has a single component in this section. Also, assume that Σ does not have any boundary. Such a surface partitions $\mathbb{R}^3$ into two components. The unbounded component of $\mathbb{R}^3 \backslash \Sigma$ is denoted Ω_O. The rest, that is, $\mathbb{R}^3 \setminus \Omega_O$ is denoted Ω_I. Notice that Ω_O is open where Ω_I is closed with Σ on its boundary. For the analysis we need the normals of Σ oriented. As before we orient the normal $\mathbf{n}_x$ at any point $x \in \Sigma$ outward, that is, toward Ω_O.

When Σ is connected, there are two sets of medial balls. The inner medial balls have their centers in Ω_I while the outer medial balls have their centers in Ω_O. The surface Σ bounds the union of both sets of medial balls. In other words, Σ consists of points where the inner medial balls meet with the outer ones. Likewise we can separate the polar balls into inner and outer ones. The inner ones have their centers in Ω_I whereas the outer ones have their centers in Ω_O. The union of inner polar balls does not necessarily meet the union of outer polar balls in a surface. Nevertheless, the points lying in both unions and the points lying in neither of the unions are close to Σ. A surface is extracted out of these points by using *power diagrams* of the polar balls.

6.1.1 Definition

Power Diagrams

A power diagram is a generalization of the Voronoi diagram where the input points and distances are weighted. A *weighted point* $\hat{p}$ is a point $p \in \mathbb{R}^3$ with a weight w_p, that is, $\hat{p} = (p, w_p)$. The weighted point $\hat{p}$ can be thought of as a ball B_{p,w_p}. Conversely, a ball $B_{p,r}$ can be thought as a weighted point $\hat{p} = (p, r)$. The *power distance* between two points $\hat{p}$ and $\hat{q}$ is given by

$$\pi(\hat{p}, \hat{q}) = \|p - q\|^2 - w_p^2 - w_q^2.$$

See Figure 6.1 for a geometric interpretation of the power distance of an unweighted point $\hat{x} = (x, 0)$ from a weighted one in $\mathbb{R}^2$. Let $\hat{P}$ denote a set of weighted points where P is the set of corresponding unweighted points. A *power cell* $V_{\hat{p}}$ for a weighted point $\hat{p} \in \hat{P}$ is defined as

$$V_{\hat{p}} = \{x \in \mathbb{R}^3 \mid \pi(x, \hat{p}) \leq \pi(x, \hat{q})\ \forall \hat{q} \in \hat{P}\}.$$

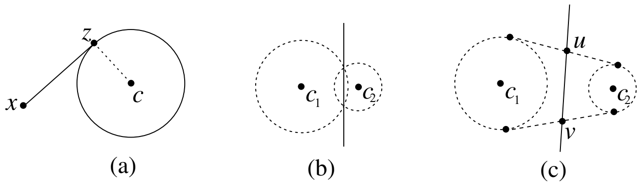


Figure 6.1. (a) $\pi(x, \hat{c})$ is the squared length of the tangent xz. (b) The cells of $\hat{c}_1$ and $\hat{c}_2$ are the half planes on the left and right respectively of the solid line. The line passes through the intersection of the two circles. (c) Similar to (b), the dividing line passes through the midpoints u and v of the bitangents.

The facets and hence all faces of a power cell are linear. A point x in such a facet satisfies the equation

$$\|p - x\|^2 - w_p^2 = \|q - x\|^2 - w_q^2$$
$$\text{or, } \|p\|^2 - 2p^T x - w_p^2 = \|q\|^2 - 2q^T x - w_q^2$$

which is an equation of a plane.

Definition 6.1. *For a set of weighted points $\hat{P} \subset \mathbb{R}^3$, the* power diagram Pow $\hat{P}$ *is the 3-complex made by the faces of the power cells $\{V_{\hat{p}} \mid \hat{p} \in \hat{P}\}$.*

If the two balls corresponding to the two weighted points determining a facet meet, the facet lies on the plane passing through the circle where the boundaries of the two balls meet. As the ordinary unweighted Voronoi diagram, the power diagram defines a dual triangulation called the weighted Delaunay triangulation. Precisely, for a weighted point set $\hat{P}$, the weighted Delaunay triangulation is a simplicial complex where a simplex σ is in the triangulation if the power cells in Pow $\hat{P}$ for the vertices of σ have a nonempty intersection.

See Figure 6.2 for a power diagram of a set of points in the plane. Notice that the power cell of a weighted point may be empty or may not contain the point. However, such anomalies occur only when a weighted point lies completely inside another one. In our case these situations will not arise.

Let P be a sufficiently dense sample of Σ. Recall that each sample point $p \in P$ defines two poles in the Voronoi diagram Vor P. These two poles lie in different components of $\mathbb{R}^3$ separated by Σ. The pole in the unbounded component Ω_O is called the *outer pole* and the pole in the bounded component Ω_I is called the *inner pole*. The set of inner poles, C_I, defines a weighted point set $\hat{C}_I$ where the weight of each pole is equal to the radius of the corresponding polar ball. Similarly, the set of outer poles, C_O, defines a weighted point set $\hat{C}_O$. The entire set of poles $C = C_I \cup C_O$ defines the weighted point set $\hat{C} = \hat{C}_I \cup \hat{C}_O$.

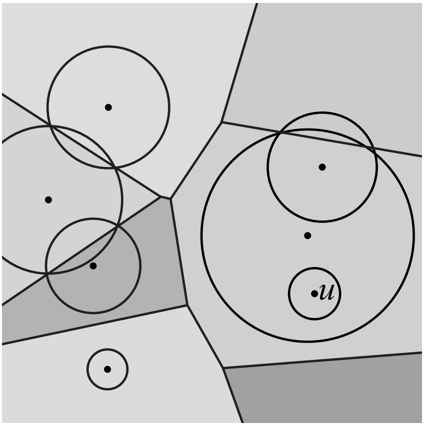


Figure 6.2. Power diagram of a set of points in the plane. A power cell of a weighted point may be empty or may not contain the point; $\hat{u}$ is such a weighted point.

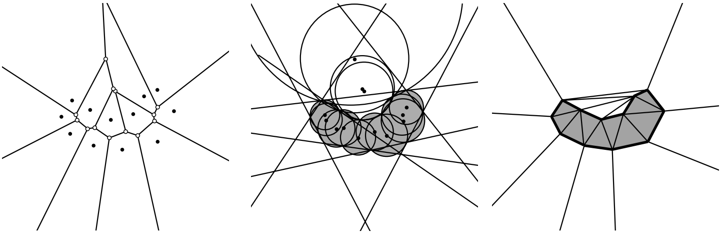


Figure 6.3. Power crust of a set of points in the plane: (left) the Voronoi diagram of the points, (middle) the polar balls including the infinite ones; the inner polar balls are shaded, (right) the inner power cells are shaded; the power crust edges drawn with thick segments separate the inner cells from the outer ones.

Definition 6.2. *The power crust* Pwc$P$ *of* P *is defined as the subcomplex in* Pow $\hat{C}$ *where a face* $F \in$ Pow $\hat{C}$ *is in* Pwc$P$ *if a cell corresponding to an inner pole and a cell corresponding to an outer pole meet in* F.

Figure 6.3 illustrates the concept of the power crust for a set of points in the plane.

According to the definition, Pwc$P$ is a collection of vertices, edges, and facets where a cell for an inner pole meets a cell for an outer pole. Consider such an edge. The sequence of cells around it should at least change from an inner pole to an outer one and again from an outer pole to an inner one. This implies the following lemma.

Lemma 6.1. *Each edge in* Pwc$P$ *has an even number (greater than zero) of facets incident to it.*

The above lemma implies that Pwc$P$ cannot have an edge with a single or no facets incident to it. This contributes to the watertightness of Pwc$P$. Also, since each point of P has an inner pole and an outer pole, it is incident to cells in Pow $\hat{C}$ where at least one corresponds to an inner pole and another corresponds to an outer pole. Therefore, we have:

Lemma 6.2. *Each point of a ε-sample P belongs to* Pwc P *if ε is sufficiently small.*

6.1.2 Proximity

The power crust Pwc$P$ is geometrically close to Σ. Also, the normals to the facets of Pwc$P$ match closely with the normals of Σ at nearby points. We will see that these properties together ensure that the underlying space of Pwc$P$ is homeomorphic to Σ.

The union of the inner and outer polar balls plays a vital role in establishing the geometric and normal proximity of Pwc$P$ to Σ. Let U_I and U_O denote the union of the inner and outer polar balls respectively.

Geometric Proximity

First, observe that any point on Pwc P is either in both of U_I and U_O or in neither of them. Specifically, let x be any point on a facet F in Pwc$P$. The facet F belongs to two cells for two poles, say $c_1 \in C_I$ and $c_2 \in C_O$. By the property of the power diagram, the facet F is either in both polar balls if they intersect, or in neither of them if they do not. In the first case, clearly $x \in U_I \cap U_O$. In the other case x cannot belong to any polar ball since the power distance $\pi(x, \hat{c}_1)$ of x for the polar ball $\hat{c_1}$ is minimum among all poles and it is positive.

The next lemma states that points which belong to both of U_I and U_O or to neither of their interiors are very close to Σ. We will skip the proof (see Exercise 8).

Lemma 6.3. *Let x be a point so that either $x \in U_I \cap U_O$ or $x \notin \mathrm{Int}U_I \cup \mathrm{Int}U_O$. For a sufficiently small ε, $\|x - p\| = \tilde{O}(\varepsilon) f(p)$ where $p \in P$ is the nearest sample point to x.*

We already know that any point in the underlying space $|\mathrm{Pwc}P|$ of Pwc$P$ is in either $U_I \cap U_O$ or in $\mathbb{R}^3 \setminus (U_I \cup U_O)$. Therefore, the above lemma along with the ε-sampling condition leads to the following theorem.

Theorem 6.1 (PC-Hausdorff). *For a sufficiently small ε, let P be a ε-sample of Σ. Each point $x \in |\mathrm{Pwc}P|$ is within $\tilde{O}(\varepsilon) f(\tilde{x})$ distance of Σ where $\tilde{x}$ is* *the closest point to x on Σ. Conversely, each point $x \in \Sigma$ is within $\tilde{O}(\varepsilon) f(x)$ distance of* $|\mathrm{Pwc}P|$.

Normal Proximity

The boundaries of U_I and U_O remain almost parallel to the surface Σ. To describe this phenomenon precisely, we need to define a normal for each point on the union boundaries. For a polar ball, orient the normal to its boundary to point outward, that is, away from the center. The normals on the boundaries of U_I and U_O are well defined for points that belong to a single polar ball. The points that belong to more than one polar ball do not have well defined normals. Therefore, when we talk about a normal at a point x on the boundary of U_I (or U_O), we mean the normal to the boundary of any polar ball incident to x. The following lemma is the key to establishing the normal proximity. We will not prove this lemma here though a general version of this lemma is proved in Section 7.3 in the context of noisy samples. The corollary of the General Normal Theorem 7.1 has the same condition as that of the following lemma with $\delta = \tilde{O}(\varepsilon)$.

Lemma 6.4. *Let x be any point in an inner polar ball $B_{c,r}$ where $\|x - \tilde{x}\| = \tilde{O}(\varepsilon) f(\tilde{x})$ for a sufficiently small ε. One has $\angle(\mathbf{n}_{\tilde{x}}, \overrightarrow{cx}) = \tilde{O}(\sqrt{\varepsilon})$.*

In the above lemma, $B_{c,r}$ is an inner polar ball containing x. It is possible that x does not belong to the cell of $c \in C_I$. The next lemma considers the polar ball whose power distance to x is the least, that is, x belongs to the cell of the center of that polar ball.

Lemma 6.5. *For a sufficiently small ε, let x be a point within $\tilde{O}(\varepsilon) f(\tilde{x})$ distance from its closest point $\tilde{x}$ in Σ. If c is the inner pole where x belongs to the cell of c, $\angle(\mathbf{n}_{\tilde{x}}, \overrightarrow{cx}) = \tilde{O}(\sqrt{\varepsilon})$.*

Proof. Let $B_{c,r}$ be the inner polar ball centering c. If $x \in B_{c,r}$, Lemma 6.4 establishes the claim.

Consider the other case when x does not belong to $B_{c,r}$. In that case it can be shown that the distance of x to $B_{c,r}$ is $\tilde{O}(\varepsilon) f(\tilde{x})$. The closest point of x to $B_{c,r}$ lies on the segment xc. Let this point be z. The point z is $\tilde{O}(\varepsilon) f(\tilde{x})$ away from its closest point $\tilde{z}$ on Σ since this distance is no more than the sum of the distances of x from Σ and of z from x. By Lipschitz property of f this distance is also no more than $\tilde{O}(\varepsilon) f(\tilde{z})$. Now Lemma 6.4 applied to z gives that the angle between $\mathbf{n}_{\tilde{z}}$ and $\overrightarrow{cz}$ is $\tilde{O}(\sqrt{\varepsilon})$. Since the normals at $\tilde{z}$ and $\tilde{x}$ make at most $\tilde{O}(\varepsilon)$ angle (Normal Variation Lemma 3.3), the claimed bound follows. ■

Notice that the above Lemma also holds for outer poles.

The above lemma and some more observations together lead to the following result about the normals of the facets in Pwc P.

Theorem 6.2 (PC-Normal). *Let P be an ε-sample of Σ where ε is sufficiently small. Let $\mathbf{n}_F$ be the normal to any facet F in* Pwc P *and x be the point in Σ closest to F. Then, $\angle_a(\mathbf{n}_F, \mathbf{n}_x) = \tilde{O}(\sqrt{\varepsilon})$.*

Proof. Let y be the point in F closest to x. We know from PC-Hausdorff Theorem 6.1 that $\|x - y\| = \tilde{O}(\varepsilon) f(x)$. Let c and c' be the inner and outer poles respectively whose cells share F. Lemma 6.5 implies $\angle(\mathbf{n}_x, \overrightarrow{cy}) = \tilde{O}(\sqrt{\varepsilon})$ and $\angle(\mathbf{n}_x, \overrightarrow{yc'}) = \tilde{O}(\sqrt{\varepsilon})$. This implies that $\angle(\overrightarrow{cy}, \overrightarrow{c'y}) = \pi - \tilde{O}(\sqrt{\varepsilon})$ which also means $\angle(\overrightarrow{cc'}, \overrightarrow{cy}) = \tilde{O}(\sqrt{\varepsilon})$. We are done since the line containing cc' is perpendicular to the plane of F and hence

$$\begin{aligned}\angle_a(\mathbf{n}_F, \mathbf{n}_x) = \angle_a(\overrightarrow{cc'}, \mathbf{n}_x) &\leq \angle_a(\overrightarrow{cc'}, \overrightarrow{cy}) + \angle_a(\overrightarrow{cy}, \mathbf{n}_x)\\ &= \tilde{O}(\sqrt{\varepsilon}).\end{aligned}$$

■

6.1.3 Homeomorphism and Isotopy

Lemma 6.5 is the key to establishing the homeomorphism between Σ and $|\text{Pwc } P|$, the underlying space of Pwc P. Consider the function $\nu \colon \mathbb{R}^3 \setminus M \to \Sigma$ where $\nu(x) = \tilde{x}$ and M is the medial axis of Σ. We show that the restriction of ν to $|\text{Pwc } P|$ realizes this homeomorphism.

Theorem 6.3. $|\text{Pwc } P|$ *is homeomorphic to Σ where P is a sufficiently dense sample of Σ.*

Proof. Consider the restriction $\nu' \colon |\text{Pwc } P| \to \Sigma$ of ν. Since $|\text{Pwc } P|$ avoids the medial axis (PC-Hausdorff Theorem 6.1) for a sufficiently small ε, the map ν' is well defined. It is also continuous since ν is. We show that ν' is one-to-one. If not, at least two points x, x' exist on $|\text{Pwc } P|$ where $\nu'(x) = \nu'(x') = \tilde{x}$. The point x, x', and $\tilde{x}$ lie on the line ℓ normal to Σ at $\tilde{x}$. By PC-Hausdorff Theorem 6.1, both x and x' are within $\tilde{O}(\varepsilon) f(\tilde{x})$ distance from $\tilde{x}$. We claim that ℓ intersects $|\text{Pwc } P|$ only at a single point within $\tilde{O}(\varepsilon) f(\tilde{x})$ distance, thereby contradicting the existence of x and x' as assumed. To prove the claim assume without loss of generality that x is further from $\tilde{x}$ than x' is. Consider two functions Π_I and Π_O that assign to each point $z \in \mathbb{R}^3$ the minimum power distance to any pole in C_I

and C_O respectively. Consider moving a point z from $\tilde{x}$ toward x. Lemma 6.5 implies that the vectors from the nearest (in terms of power distance) pole in C_I and C_O to z make an angle $\tilde{O}(\sqrt{\varepsilon})$ with the line ℓ. This means that either the function Π_I monotonically increases while Π_O monotonically decreases or Π_I monotonically increases while Π_O monotonically decreases as z moves from $\tilde{x}$ to x. It follows that the two functions become equal only at a single point. Since each point on $|\mathrm{Pwc}\, P|$ has equal minimum power distances to the poles in C_I and to the poles in C_O, we must have $x = x'$.

Consider the set $\Sigma' = \nu'(|\mathrm{Pwc}\, P|)$. Obviously, ν' maps $|\mathrm{Pwc}\, P|$ surjectively onto Σ'. Since $|\mathrm{Pwc}\, P|$ is compact, so is Σ'. The inverse map of ν' from Σ' to $|\mathrm{Pwc}\, P|$ is continuous since the inverse of a continuous map between two compact spaces is also continuous. Therefore, ν' is a homeomorphism between $|\mathrm{Pwc}\, P|$ and Σ'. The only thing that remains to be shown is that $\Sigma' = \Sigma$.

Notice that Σ' is a manifold without boundary since Pwc P does not have any edge with a single facet incident on it (Lemma 6.1). Then Σ' being a submanifold of Σ can differ from it by a component. But, that is impossible since we assume that Σ has a single component. ■

We already indicated in Section 1.1 that two homeomorphic surfaces can be embedded in $\mathbb{R}^3$ in ways that are fundamentally different. So, it is desirable that we prove a stronger topological relation between Σ and $|\mathrm{Pwc}\, P|$. We show that, not only are they homeomorphic but are isotopic as well. This means one can deform $\mathbb{R}^3$ continuously so that $|\mathrm{Pwc}\, P|$ is taken to Σ.

Theorem 6.4 (PC-Isotopy). *Let P be an ε-sample of Σ. $|\mathrm{Pwc}\, P|$ is isotopic to Σ if ε is sufficiently small.*

Proof. For isotopy we define a map $\xi : \mathbb{R}^3 \times [0, 1] \rightarrow \mathbb{R}^3$ so that $\xi(|\mathrm{Pwc}\, P|, 0) = |\mathrm{Pwc}\, P|$ and $\xi(|\mathrm{Pwc}\, P|, 1) = \Sigma$ and $\xi(\cdot, t)$ is a continuous, one-to-one, and onto map for all $t \in [0, 1]$. Consider a tubular neighborhood N_Σ of Σ as

$$N_\Sigma = \{x \mid d(x, \Sigma) \leq c\varepsilon f(\tilde{x})\}$$

where each point y of $|\mathrm{Pwc}\, P|$ is within $c\varepsilon f(\tilde{y})$ distance and $c\varepsilon < 1$ is sufficiently small. For a sufficiently small ε, such a c exists by the PC-Hausdorff Theorem 6.1. In $\mathbb{R}^3 \setminus N_\Sigma$ we define ξ to be identity for all $t \in [0, 1]$. For any point $x \in N_\Sigma$ we define ξ as follows. Consider the line segment g passing through x and normal to Σ with endpoints g_i and g_o on the two boundaries of N_Σ. Since $c\varepsilon < 1$, the tubular neighborhood N_Σ avoids the medial axis and hence g intersects Σ in exactly one point, say at u. Also, by arguments in the

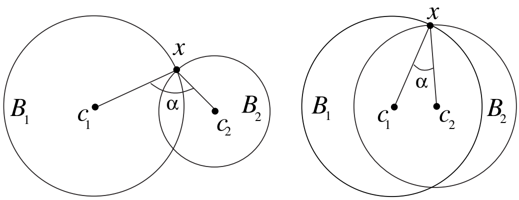


Figure 6.4. Shallow intersection (left) and deep intersection (right) between two balls.

proof of Theorem 6.3, g intersects $|\text{Pwc } P|$ only at a single point, say at w. In N_Σ we define $w_t = \xi(w, t) = tu + (1 - t)w$ and let $\xi(\cdot, t)$ linearly map the segments $g_i w$ to $g_i w_t$, and $w g_o$ to $w_t g_o$. That is,

$$\begin{aligned}\xi(x,t) &= w_t + (x - w)\frac{g_i - w_t}{g_i - w}, \qquad x \text{ is in } g_i w\\ &= w_t + (x - w)\frac{g_o - w_t}{g_o - w}, \qquad x \text{ is in } g_o w.\end{aligned}$$

Clearly, ξ is continuous, one-to-one, and onto for each $t \in [0, 1]$ with

$$\xi(|\text{Pwc } P|, 0) = |\text{Pwc } P| \quad \text{and} \quad \xi(|\text{Pwc } P|, 1) = \Sigma.$$

■

6.1.4 Algorithm

We compute the power crust by identifying the cells corresponding to the inner and outer poles and then computing the facets separating a cell of an inner pole from a cell of an outer pole. The poles are labeled inner or outer by computing how deeply the corresponding polar balls intersect. It is important for this labeling algorithm that the surface Σ has a single connected component.

The labeling algorithm is a simple traversal of the Delaunay graph structure. It utilizes the following properties of the polar balls. If two polar balls intersect, their depth of intersection depends on their types. If one of them is inner and the other is outer, the intersection is shallow. On the other hand, if both of them are inner or outer, the intersection is deep . We formalize this idea.

Let two balls B_1 and B_2 intersect and x be any point on the circle in which their boundaries intersect. We say that B_1 and B_2 intersect at an angle α if the vectors $\overrightarrow{c_1x}$ and $\overrightarrow{c_2x}$ make an angle α where c_1 and c_2 are the centers of B_1 and B_2 respectively (see Figure 6.4).

Lemma 6.6. *Let B_1 and B_2 be two polar balls that intersect. For a sufficiently small ε the following hold.*

(i) If B_1 is an inner and B_2 is an outer polar ball, they intersect at an angle of $\pi - \tilde{O}(\sqrt{\varepsilon})$.

(ii) If both of B_1 and B_2 are inner or outer polar balls, and there is a facet F in Pow $\hat{C}$ *between the cells of c_1 and c_2 with a point $x \in F$ where $\|x - \tilde{x}\| = \tilde{O}(\varepsilon) f(\tilde{x})$, then B_1 and B_2 intersect at an angle of $\tilde{O}(\sqrt{\varepsilon})$.*

Proof. Consider (i). Let y be a point on the circle where the boundaries of B_1 and B_2 intersect. Since y belongs to $U_I \cap U_O$, Lemma 6.3 asserts that $\|y - \tilde{y}\| = \tilde{O}(\varepsilon) f(\tilde{y})$.

Let $\mathbf{n}_{\tilde{y}}$ be the oriented outer normal. By Lemma 6.4 we get that the angle between $\mathbf{n}_{\tilde{y}}$ and the vector $\overrightarrow{c_i x}$ is $\tilde{O}(\sqrt{\varepsilon})$ if c_i is an inner pole, and the angle is $\pi - \tilde{O}(\sqrt{\varepsilon})$ if c_i is an outer pole. Since c_1 and c_2 are poles with opposite labels in case (i), the claimed angle bound follows.

Consider (ii). The poles have same labels, say inner in case (ii). Both polar balls have a point, namely x with $\|x - \tilde{x}\| = \tilde{O}(\varepsilon) f(\tilde{x})$. Therefore, the vectors $\overrightarrow{c_1 x}$ and $\overrightarrow{c_2 x}$ make an angle of $\tilde{O}(\sqrt{\varepsilon})$ with $\mathbf{n}_{\tilde{x}}$ according to Lemma 6.4. The claimed angle bound is immediate. ■

Now we describe the labeling algorithm for the poles. The angle at which two polar balls centering c and c' intersect is denoted $\angle c, c'$.

```
LabelPole(Vor $P$,$C$,Pow $\hat{C}$)
label all poles in $C$ outer;
choose any sample $p$ on Conv $P$;
mark the finite pole $c$ of $p$ inner;
push $c$ into a stack $S$;
while $S \neq \emptyset$ do
    $c :=$ pop $S$;
    mark $c$ processed;
    for each pole $c'$ adjacent to $c$ in Pow $\hat{C}$ do
        if ($\angle c, c' > \frac{2\pi}{3}$) and ($c'$ is not processed)
            label $c'$ inner;
            push $c'$ into $S$;
        endif
    endfor
endwhile
return $C$ with labels.
```

Notice that we chose an angle threshold of $\frac{2\pi}{3}$ somewhat arbitrarily to decide if two balls are intersecting deeply or not. The point is that any angle $\tilde{O}(\sqrt{\varepsilon})$

will work as long as ε is sufficiently small. Assuming $\varepsilon < 0.1$, the angle of $\frac{2\pi}{3}$ is a safe choice. The correctness of LabelPoles follows from the next lemma.

Lemma 6.7 (Label). *Each pole is labeled correctly by* LabelPole.

Proof. Consider the graph of the power crust edges. Since Σ has a single component, this graph is connected. We call two inner polar balls adjacent if they contribute a facet in Pow $\hat{C}$ and the facet has an edge in the power crust.

The first inner polar ball marked by LabelPole is correctly labeled as the finite pole of a sample point on the convex hull is necessarily inner. This sample point, as all others, lie on the power crust. Since the graph of the power crust edges is connected, all inner polar balls contributing an edge on the power crust are labeled by LabelPole. Also, since each polar ball has a sample point on its boundary which appears as an endpoint of a power crust edge, Lemma 6.6 can be applied assuring that all of them are labeled correctly. ■

Now we enumerate the steps of the power crust.

```
PowerCrust($P$)
  compute Vor $P$;
  compute all poles $C$ in Vor $P$;
  compute Pow $\hat{C}$;
  $C$ :=LabelPole(Vor $P$, $C$, Pow $\hat{C}$);
  mark each facet of Pow $\hat{C}$ separating a cell of an
     inner pole from that of an outer pole;
  output the 2-complex made by the marked facets.
```

The PC-Hausdorff Theorem 6.1, the PC-Normal Theorem 6.2, the PC-Isotopy Theorem 6.4, and the Label Lemma 6.7 make the following theorem.

Theorem 6.5. *Given an ε-sample of a smooth, compact surface Σ without boundary, the* PowerCrust *computes a 2-complex* Pwc P *with the following properties if ε is sufficiently small.*

(i) Each point $x \in |\mathrm{Pwc}\, P|$ has $\|x - \tilde{x}\| = \tilde{O}(\varepsilon) f(\tilde{x})$.

(ii) Each facet $F \in \mathrm{Pwc}\, P$ has a normal $\mathbf{n}_F$ with $\angle_a(\mathbf{n}_F, \mathbf{n}_x) = \tilde{O}(\sqrt{\varepsilon})$ where x is the point in Σ closest to F.

(iii) $|\mathrm{Pwc}\, P|$ is isotopic to Σ.

The PowerCrust algorithm computes two Voronoi diagrams. The first one in step 1 takes $O(n^2)$ time and space in the worst case for a set of n points.

The power diagram computation in step 3 takes $O(m^2)$ time and space if m is the number of poles. Since we have at most two poles for each input point, we have $m \leq 2n$. Therefore, step 3 takes $O(n^2)$ time and space in the worst case. The complexity of all other steps are dominated by the Voronoi diagram computations. Therefore, PowerCrust runs in $O(n^2)$ time and space in the worst case.

6.2 Tight Cocone

The output of PowerCrust has each input sample point as a vertex. However, each of the vertices of Pwc P is not necessarily a point in P. The vertices of the power crust facets are the points whose nearest power distance to the poles is determined by three or more poles. Each sample point satisfies this property and so do other points. As a result, the number of vertices in the output surface is usually greater than the number of input points. In some cases, this increase in size can be a prohibitive bottleneck, especially when dealing with large data sets. Further, the power crust is a subcomplex of a Voronoi diagram and is not necessarily triangular. Of course, its polygonal facets can be triangulated but at the expense of increased size and many coplanar triangles. These limitations of PowerCrust are remedied by the TightCocone algorithm.

The overall idea of TightCocone is to label the Delaunay tetrahedra computed from the input sample as *in* or *out* according to an initial approximation of the surface and then peeling off all *out* tetrahedra. This leaves the *in* tetrahedra, the boundary of whose union is output as the watertight surface. The output of BoundCocone described in Chapter 5 is taken as the initial approximated surface possibly with holes and other artifacts.

Since the output of TightCocone is the boundary of the union of a set of tetrahedra, it is watertight by definition. However, apart from being watertight, the output also should approximate the geometry of the original sampled surface. For any such theoretical guarantee, we need the sample to be dense enough. Since the main motivation for designing TightCocone is to consider undersampled point sets, we do not prove any guarantee about TightCocone except that all guarantees for Cocone also hold for TightCocone when the sample is sufficiently dense.

Although we do not attempt to design TightCocone with theoretical guarantees, we make some decisions in the algorithm based on the assumption that the undersampling is not arbitrary. This means that BoundCocone computes most of the intended surface except with holes that are locally repairable. Of course, if this assumption is not obeyed, the output surface, though watertight may not be close to the original one and may even be empty.

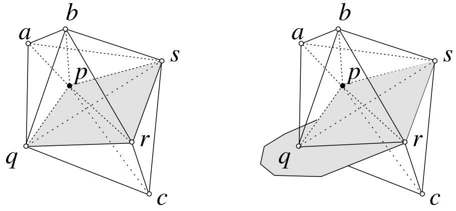


Figure 6.5. The umbrella of p has three triangles pqr, prs, and pqs. This umbrella separates the tetrahedra incident to p into two clusters: the upper cluster $\{absp, asqp, abpq, bqrp, brsp\}$ and the lower cluster $\{cqrp, csrp, cqsp\}$. Suppose the walk entered p with the pair $(p, bprs)$. The right picture shows that the unexplored point q has an umbrella. Therefore, the pair $(q, bprq)$ is entered into the stack since q is a good and unexplored point.

6.2.1 Marking

The BoundCocone algorithm as described in Chapter 5 computes a preliminary surface possibly with holes and other artifacts at the undersampled regions. The sample points in the well-sampled regions have their neighborhoods well approximated. Specifically, the set of surface triangles incident to these points form a *topological disk*. We call the points *good* whose incident surface triangles form a topological disk. The rest of the points are called *poor*.

Definition 6.3. *The union of surface triangles incident to a good point p is called its* umbrella *denoted as U_p.*

The algorithm to mark tetrahedra walks through the Delaunay triangulation in a depth first manner using the vertex and triangle adjacencies. It maintains a stack of pairs (p, σ) where p is a good point and σ is a tetrahedron incident to p which has been marked *out*. Suppose the pair (p, σ) is currently popped out from the stack. The umbrella U_p locally separates the tetrahedra incident to p into two clusters, one on each side (see Figure 6.5). The cluster that contains σ is marked *out* since σ is already marked *out*. The other cluster gets the marking *in*. This is done by initiating a local walk from σ that traverses all tetrahedra through triangle adjacency without ever crossing a triangle in U_p and marking each tetrahedron as *out*. The rest of the tetrahedra that are not encountered in this walk get the *in* marking. During this local walk in the *out* cluster, when a vertex q of U_p is reached through a tetrahedron σ', the pair (q, σ') is pushed into the stack if q is good and is not explored yet (see Figure 6.5).

Now we face the question of initiating the stack. For this we assume that Del P is augmented with "infinite" tetrahedra that are incident to a triangle on the convex hull of P and a point at infinity. The stack is initiated with a good point on the convex hull paired with an incident infinite tetrahedron.

```
MARK(Del $P$)
push an infinite tetrahedron incident to a good point
on the convex hull to stack $S$;
let $T_p$ be the set of tetrahedra incident to $p$ in Del $P$;
while $S \neq \emptyset$ do
    $(p, \sigma) :=$ pop $S$;
    mark $p$ processed;
    $G := \{\sigma\}$;
    while $(\exists \sigma \in G)$ and $(\sigma' \in T_p \setminus G)$ and $(\sigma \cap \sigma' \notin U_p)$ do
        $G := G \cup \{\sigma'\}$;
        for all good vertex $q$ of $\sigma'$ do
            if $(q \in U_p)$ and ($q$ not processed)
                push $(q, \sigma')$ to $S$;
        endfor
    endwhile
    mark each $\sigma \in G$ *out*;
    mark each $\sigma \in T_p \setminus G$ *in*;
endwhile.
```

For most of the data in practice, the surface computed by BOUNDCOCONE is well connected, that is, all triangles incident to good points can be reached from any other good point via a series of triangle adjacencies. Assuming this connectivity of the preliminary surface computed by BOUNDCOCONE, the above procedure marks all tetrahedra that are incident to at least one good sample point. However, the tetrahedra whose vertices are all poor are not marked by this step. We call them *poor tetrahedra*. The intended output surface is the boundary of the union of a set of tetrahedra. Accordingly, the poor tetrahedra should be marked *in* or *out*. We follow a heuristic here based on the assumption that the undersampling is local.

In the justification of poor tetrahedra marking we assume an intended ideal surface that is the boundary of a set of tetrahedra. These tetrahedra are referred as the ones lying inside the intended surface and the rest lying in its outside. Of course, here we deviate from the mathematical precision, but it makes the description more intuitive.

The poor tetrahedra whose vertices lie in a single undersampled region tend to be small when undersampling is local. We choose to mark them *in* and the

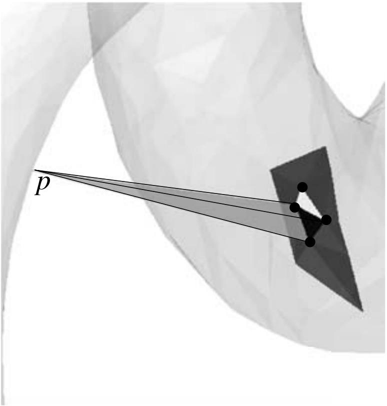


Figure 6.6. The four vertices marked with dark circles border a hole. The poor tetrahedron with these four vertices is marked *in*. The sharp tip p of the long tetrahedron is a good point which marks it *out*. When this tetrahedron is peeled, the triangle opposite to p fills the hole partially. The other triangle of the hole also gets into the output by a similar peeling.

peeling process later is not allowed to peel them away. This allows the surface to get repaired in the undersampled region. See Figure 6.6 for an illustration.

Other poor tetrahedra that connect vertices from different undersampled regions tend to be big. If such a big poor tetrahedron lies outside the intended surface, we need to take it out. So, it should be marked *out*. On the other hand, if this big poor tetrahedron lies inside the intended surface, we need to mark it as *in*. Otherwise, a large void/tunnel in the surface is created by taking out this tetrahedron. We eliminate this dilemma using the assumption that undersampling is local. Call a triangle in a tetrahedron *small* if its circumradius is the least among all triangles in the tetrahedron. If a poor tetrahedron has a triangle with vertices from the same undersampled region, then that triangle is small. The poor tetrahedra lying inside the intended surface have to be reached by the peeling process that peels away all *out* marked tetrahedra. This means that the inner poor tetrahedra have to be reached through the small triangle. We take this observation into account during peeling while dealing with the poor tetrahedra and defer designating them during the marking step.

6.2.2 Peeling

After the marking step, a walk is initiated to peel off the tetrahedra that are marked *out* and some others. The boundary of the union of the remaining

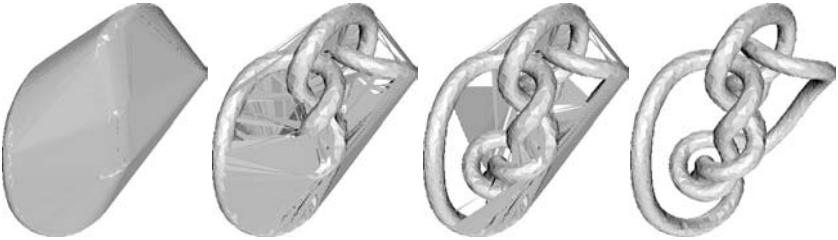


Figure 6.7. The boundary of the union of peeled tetrahedra as peeling process progresses.

tetrahedra form the watertight surface. This is also the surface of the union of the peeled tetrahedra (see Figure 6.7).

The walk maintains a stack of surface triangles that form the boundary of the union of the tetrahedra peeled so far. It is initiated with all convex hull triangles. At any generic step, a triangle, say t, is popped out from the stack. One of the tetrahedra incident to t is already peeled. If the other incident tetrahedron, say σ, is also already peeled the triangle t separates two *out* tetrahedra and is not put in the output. Otherwise, there are two possibilities. If σ is not poor and marked *in*, we put t in the output list. In the other case either σ is marked *out* or σ is poor. When σ is marked *out* the walk should move into σ through t, which is done by replacing t with the other three triangles of σ into the stack. If σ is a poor tetrahedron, the walk is also allowed to move into σ through t only if t is not the small triangle in σ. This is done to protect peeling of the inner poor tetrahedra as we discussed before. Notice that if σ is a poor tetrahedron outside the intended surface, it will be eventually reached by the peeling process at triangles other than the small one. But, if σ is a poor tetrahedron inside, it can only be reached from outside through its small triangle due to the assumption that undersampling is local. The walk terminates with the surface triangles in the output list when there are no more triangles to process from the stack.

```
PEEL(Del $P$)
push all convex hull triangles in Del $P$ to stack $S$;
mark all infinite tetrahedra peeled;
$T := \emptyset$;
while $S \neq \emptyset$ do
    $t :=$ pop $S$;
    if $\exists \sigma \in$ Del $P$ incident to $t$ and not marked peeled
        if ($\sigma$ is not poor) and (marked in)
            $T := T \cup t$;
```

```
      else
        if ($\sigma$ is marked *out*) *or* ($\sigma$ is poor and $t$ is
          not the small triangle in $\sigma$)
            mark $\sigma$ *peeled*;
            push all triangles of $\sigma$ other than $t$ to $S$;
        endif
      endif
    endif
endwhile
return $T$.
```

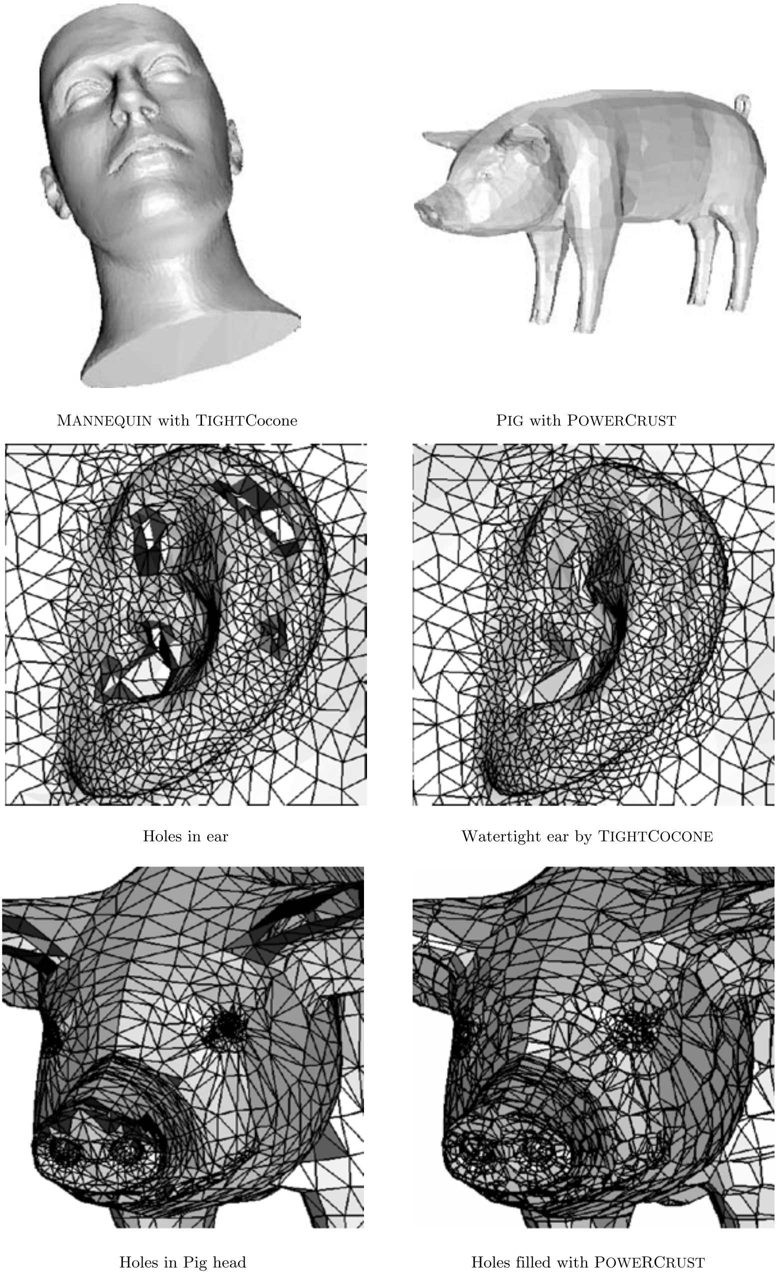
```
TightCocone($P$)
1   compute Del $P$;
2   Mark(Del $P$);
3   $T$ := Peel(Del $P$);
4   output $T$.
```

The complexity of TightCocone is dominated by the Delaunay triangulation computation in step 1. The marking and peeling steps are mere traversal of the Delaunay triangulation data structure. Therefore, in the worst case TightCocone takes $O(n^2)$ time and space for a set of n input points. However, unlike PowerCrust, it computes the Delaunay/Voronoi data structure only once.

6.3 Experimental Results

In Figure 6.8, we show the results of PowerCrust and TightCocone on two data sets. In Mannequin there are undersamplings in eyes, lips, and ears which produce holes. TightCocone closes all these holes. In particular, in the ear there is a relatively large hole since points cannot be sampled for occlusion. This hole is nicely filled. The Pig data has severe undersampling in the hoofs, ears, and nose. They are mostly due to the fact that these thin and highly curved regions should have more sample points to capture the features properly. PowerCrust fills all holes and produces a watertight surface for this difficult data set.

The time and space complexities of both TightCocone and PowerCrust are $O(n^2)$ where n is the number of input points. However, in practice this quadratic behavior is not observed. Table 6.1 shows the timings of PowerCrust and TightCocone for four data sets on a PC with 733 MHz Pentium III CPU and 512 MB memory.



MANNEQUIN with TIGHTCocone

PIG with POWERCRUST

Holes in ear

Watertight ear by TIGHTCOCONE

Holes in Pig head

Holes filled with POWERCRUST

Figure 6.8. Results of PowerCrust and TightCocone. Holes in the surface computed by BoundCocone are filled. Triangles bordering the holes are shaded darker.

Table 6.1. *Time data*

Object	# points	PowerCrust (s)	TightCocone (s)
Pig	3,511	11	5
Manneq	12,772	54	23
Club	16,864	55	43
Horse	48,485	163	152

6.4 Notes and Exercises

Amenta, Choi, and Kolluri [7] designed the PowerCrust algorithm. The material on the power crust in this chapter is taken from this paper. The labeling algorithm in the original paper is intended for surfaces with multiple components. However, the proof given in the paper does not work for surfaces with multiple components. This is why we choose to describe the algorithm for surfaces with a single component.

In the paper [7], Amenta, Choi, and Kolluri also prove that the poles approximate a subset of the medial axis. Specifically, the Hausdorff distance between the poles and the medial axis approaches zero as ε does so. In a simultaneous work Boissonnat and Cazals [16] also establish this result. The Exercises 3 and 4 are set keeping this result in mind. In Chapter 7, we will establish a similar result for noisy samples.

The material on the TightCocone algorithm is taken from the paper by Dey and Goswami [33]. The main advantage of this algorithm is that it does not introduce any extra points as vertices in the output surface. The PowerCrust, on the other hand, introduces extra points. For example, for the Pig data set, PowerCrust generates 28,801 points from an input set of 3,511 points.

Both PowerCrust and TightCocone can handle a small amount of noise. However, it may happen that these algorithms fail completely when the noise is beyond a certain limit.

Exercises

1. Prove that the plane of a facet shared by two cells of $\hat{c}$ and $\hat{c'}$ in a power diagram is perpendicular to the line containing c and c'.
2. Let bd U_I denote the boundary of the union of inner polar balls. Show that bd U_I is homeomorphic to Σ when the input sample P is sufficiently dense.

3^h. Let $B_{m,r}$ be a finite medial ball which touches Σ at x_1 and x_2. Let $\angle x_1 m x_2 \geq \varepsilon^{\frac{1}{3}}$. Let p be the nearest sample point to m. Prove that the distance of m to the nearest pole of p is $\tilde{O}(\varepsilon^{\frac{2}{3}})r$ where the sampling density ε is sufficiently small.

4. Prove the converse of the statement in 3, that is, if c is the center of a polar ball, then there is a medial ball $B_{m,r}$ so that the distance of m to c is $\tilde{O}(\varepsilon^{\frac{2}{3}})r$ when ε is sufficiently small.
5. Prove or disprove that any 2-complex is watertight if and only if each edge has even number of facets incident to it.
6. Prove that the output of TightCocone is homeomorphic to Σ if the input point set P is sufficiently dense.
7. Redesign TightCocone so that the marking and peeling phases are combined, that is, both are done simultaneously.
8. Prove Lemma 6.3 and Lemma 6.4 [7].

9^{o}. Design a labeling algorithm for poles where the sampled surface may have multiple components. Prove that the algorithm labels all poles correctly.