EUROGRAPHICS 2002 / G. Drettakis and H.-P. Seidel
(Guest Editors)

Volume 21 (2002), Number 3

Surface reconstruction based on a dynamical system

N.N.

Abstract

We present an efficient algorithm that computes a manifold triangular mesh from a set of unorganized sample points in $\mathbb{R}^3$. The algorithm builds on the observation made by several researchers that the Gabriel graph of the sample points provides a good surface description. However, this surface description is only one-dimensional. We associate the edges of the Gabriel graph with index 1 critical points of a dynamical system induced by the sample points. Exploiting also the information contained in the critical points of index 2 provides a two-dimensional surface description which can be easily turned into a manifold.

1. Introduction

Surface reconstruction is a powerful modeling paradigm. To create a model of some solid in $\mathbb{R}^3$ one can just sample its surface and apply a surface reconstruction algorithm to the sample. Hence the task in the surface reconstruction problem is to transform a finite sample into a surface model. There are several obstacles a surface reconstruction algorithm might face. The sample could very large, noisy or to sparse too capture all the features of the solid.

The different approaches to the surfaces reconstruction problem can be divided broadly into two classes. Algorithms in the first class provide implicit surface models while the algorithms in the second class provide explicit surface models.

An implicit surface model is a function $f : \mathbb{R}^3 \to \mathbb{R}$ such that the zero set of f, i.e. $f^{-1}(0)$, is a surface that interpolates or approximates the sample. The signed distance function was introduced to this respect by Hoppe et al. [11]. This function or smoothed variants of it are used frequently in the implicit approach to surface reconstruction [6, 14, 15]. For rendering purposes an implicit surface is likely to be tranformed into a triangular mesh.

Most of the explicit approaches to surface reocnstruction compute a triangular mesh directly [1, 2, 3, 4, 10]. Here we want to have a closer look on two of these algorithms.

The algorithm of Mencl and Müller [12, 13] consists of three steps. In the first step the Euclidean minimum spanning tree of the sample points is computed. Then this tree is extended to the so called surface description graph. Finally the contours of the surface description graph are filled with triangles.

The Euclidean minimum spanning tree of the sample points is a subgraph of the Gabriel graph on the same set of points. Two sample points p and q are connected by a Gabriel edge if the open ball with diameter $\|p-q\|$ that has p and q in its boundary does not contain any sample point. The sample points together with the Gabriel edges build the Gabriel graph.

The algorithm of Attene and Spagnuolo [5] also builds on the observation that the Gabriel graph and the Euclidean minimum spanning tree provide a fairly good surface description. It removes Delaunay tetrahedra from the three-dimensional Delaunay triangulation if they are removable. The property to be removable is defined via the Gabriel graph of the sample points.

The approach we are going to present in this paper is explicit. But it also makes use of a distance function. As the signed distance function this function is defined using the Voronoi diagram of the sample points. But in contrast to the signed distance function we cannot use only the zero set of the function. Instead we are going to study all of its critical points, i.e. local minima, local maxima and saddle points. We provide a one-to-one correspondence between the Gabriel edges and some critical points of the distance function. It turns out that the Gabriel edges correspond to the index 1 saddle points of this function. We further show that the saddles of index 2 correspond to surfaces with boundaries. These boundaries consists of Gabriel edges. The collection of all these surfaces together with the Gabriel edges is a simplicial complex. This complex provides a two-dimensional surface description though not a manifold in general. Building on results of Edelsbrunner on critical points we show how to transform the complex into a manifold.

The main contributions of this paper are new insights in a natural distance function associated with a set of sample points. These insights lead to an efficient and robust algorithm for surface reconstruction. We have implemented the algorithm and evaluated its applicability. The algorithm turned out to be very robust. It is able to cope even with noisy and/or undersampled data where other explicit algorithms leave unpleasant holes. Our insights should be valuable also for other tasks in sample based modeling like sample decimation or feature extraction.

The paper is organized as follows. In the second section we introduce a distance function and its critical points. The distance function is defined via a set of sample points. The dependency on the sample points makes it suitable for our use in surface reconstruction. In the same section we introduce the notions of Voronoi diagram and Delaunay triangulation of a set of sample points. The third section introduces a dynamical system associated with the height function and the sample points. This dynamical system is used in the fourth section to define the flow complex. The flow complex is a simplicial complex on the sample points that almost provides a reconstruction of a solid if the points are sampled from the surface of this solid. In the fifth section we transform the flow complex into the reduced flow complex by using the concepts of pairing and cancellation. The reduced flow complex provides us with the reconstruction we are looking for. In the sixth section we report on an implementation of our reconstruction algorithm and present some experimental results.

2. Distance function and critical points

Let P be a set of unorganized points sampled from the surface of a solid embedded in $\mathbb{R}^3$. The *distance function* $h : \mathbb{R}^3 \to \mathbb{R}$ assigns to every point in $\mathbb{R}^3$ its least distance to any of the sample points, i.e.

$$h(x) = \min_{p \in P} \|x - p\|.$$

In contrast to the *signed distance function* which was studied by [11] in the context of surface reconstruction we cannot use only the zero set of h to determine the reconstruction. The zero set of h is the set P of sample points. This is also the set of local minima of h, i.e. the points in P are critical points of h. We are interested in all critical points of h. Besides the minima these are the maxima and saddle points. We will show later that the maxima and saddle points can also be computed easily.

The distance function h is closely related to the *Voronoi diagram* of the set P of sample points. The Voronoi diagram of P is a cell decomposition of $\mathbb{R}^3$ in convex polytopes. Every *Voronoi cell* corresponds to exactly one sample point and contains all points of $\mathbb{R}^3$ that do not have a smaller distance to any other sample point, i.e. the Voronoi cell corresponding to $p \in P$ is given as follows

$$V_p = \{x \in \mathbb{R}^3 : \forall q \in P \quad \|x - p\| \leq \|x - q\|\}.$$

Thus V_p contains exactly the points where the value of h is determined by p. Closed facets shared by two Voronoi cells are called *Voronoi facets*, closed edges shared by three or more Voronoi cells are called *Voronoi edges* and the points shared by four or more Voronoi cells are called *Voronoi vertices*. The term *Voronoi object* can denote either a Voronoi cell, facet, edge or vertex. The Voronoi diagram is the collection of all Voronoi objects. See Figure 1 for a two-dimensional example of a Voronoi diagram.

We are going to determine the critical points of the distance function h from the Voronoi diagram of P and its dual diagram. The latter diagram is called *Delaunay diagram*. The Delaunay diagram of P is a cell complex that decomposes the convex hull of the points in P. The convex hull of four or more points in P defines a *Delaunay cell* if the intersection of the corresponding Voronoi cells is not empty and there exists no superset of points in P with the same property. Analogously, the convex hull of three or two points defines a *Delaunay face* or *Delaunay edge*, respectively, if the intersection of their corresponding Voronoi cells is not empty. Every point in P is a *Delaunay vertex*. The term *Delaunay object* can denote either a Delaunay cell, face, edge or vertex.

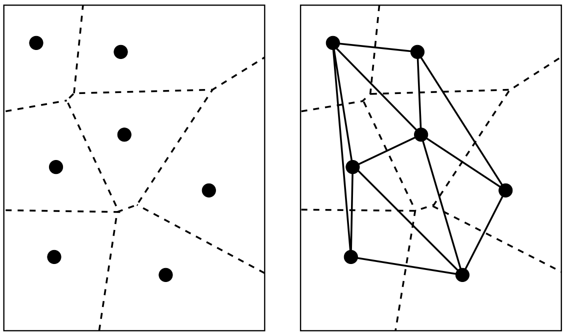

Figure 1: *Left: A set of sample points in the plane and its Voronoi diagram. Right: The Delaunay diagram of the same set of points.*

Note that from the definition of Delaunay objects we have a duality between Voronoi- and Delaunay objects. See Figure 1 for an example. That is, for every d-dimensional Voronoi object, $0 \leq d \leq 3$, there is a dual $(3-d)$-dimensional Delaunay object and vice versa. Using this duality we can characterize the critical points of h. In fact the following theorem can be proven, see also Figure 2:

Theorem 1 The critical points of h are the intersection points of Voronoi objects and their dual Delaunay objects.

We assign to every critical point of h an index. This index is the dimension of the Delaunay object used in the characterization of critical points by Theorem 1. We have already seen that the minima of h are the points in P. Since every point in P is a Delaunay vertex these points are the intersection points of Voronoi cells and their dual Delaunay vertices. Thus the minima are the critical points of index 0. It turns out that the maxima are the critical points of index 3. The remaining critical points are saddle points of index either 1 or 2. Saddle points of index 1 are intersection points of Voronoi facets and Delaunay edges. The latter edges are exactly the Gabriel edges. See Figure 2 for an example.

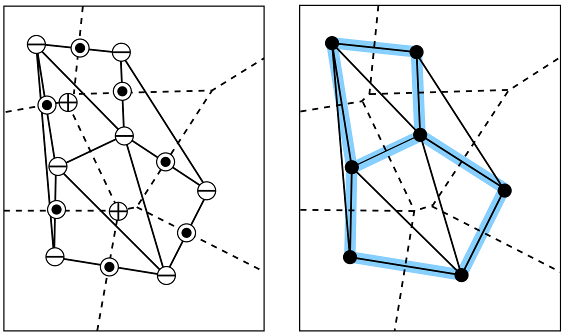

Figure 2: *Left: The minima $\ominus$, saddles $\odot$ and maxima $\oplus$ of the height function associated with the sample from Figure 1. Right: The Gabriel graph of the same set of points. The Gabriel edges are highlighted.*

3. A dynamical system

A smooth function $f : \mathbb{R}^3 \to \mathbb{R}$ gives rise to a smooth vectorfield on $\mathbb{R}^3$, namely the gradient vectorfield that maps every point x of $\mathbb{R}^3$ to the gradient of f at x. The gradient vector field in turn gives rise to an ordinary differential equation

$$\frac{d}{dt}\phi(t,x) = \text{grad}\, f(\phi(t,x)).$$

The theory of ordinary differential equations states that this equation can be integrated. The solution of the equation is a mapping

$$\phi : \mathbb{R} \times \mathbb{R}^3 \to \mathbb{R}^3,$$

which has two remarkable properties, first $\phi(0,x) = x$ and second $\phi(t,\phi(s,x)) = \phi(t+s,x)$. The first parameter of ϕ can be interpreted as time and the mapping itself tells how the points of $\mathbb{R}^3$ move in time. The two conditions state that, first the points have not moved at time zero and second any point x moves in time $t+s$ to the point where the point $\phi(s,x)$ moves in time s. The function ϕ is called a *dynamical system* or *flow* on $\mathbb{R}^3$. A point x with $\phi(x,t) = x$ for all $t \in \mathbb{R}$ is called a fixpoint of the dynamical system. The fixpoints of ϕ are exactly the critical points of f. Fixing the second parameter of ϕ provides for every $x \in \mathbb{R}^3$ a mapping $\phi_x : \mathbb{R} \to \mathbb{R}^3, t \mapsto \phi(t,x)$, which is called the orbit of x. The orbit describes the motion of the point x in time. This motion always follows the direction of steepest ascent of the function f, i.e. the orbits are integral curves to the gradient vector field of f.

The distance function h defined via a set of sample points is not a smooth function. It is only smooth in the interiors of the Voronoi cells of the sample points. Thus we cannot apply the theory of ordinary differential equations to get a dynamical system from h. But as in the case of smooth functions there is a unique direction of steepest ascent of h at every non-critical point of h. Assigning to the critical points to h the zero vector and to every other point of $\mathbb{R}^3$ the unique unit vector of steepest ascent gives rise to a vector field on $\mathbb{R}^3$. Note that this vector field is not smooth. We are going to construct a mapping

$$\phi : [0,\infty) \times \mathbb{R}^3 \to \mathbb{R}^3,$$

such that at every point $(t,x) \in [0,\infty) \times \mathbb{R}^3$ the right derivative

$$\lim_{t \leftarrow t'} \frac{\phi(t,x) - \phi(t',x)}{t - t'}$$

exists and is equal to the unique unit vector of steepest ascent at x.

For the construction of the dynamical system ϕ, which we also call an *induced flow*, the following definition turns out to be helpful.

Driver. Given $x \in \mathbb{R}^3$. Let V be the lowest dimensional Voronoi object in the Voronoi diagram of a finite set of sample points P that contains x and let σ be the dual Delaunay object of V. The driver of x is the point on σ closest to x.

The flow ϕ induced by a finite set of sample points P is given as follows: For all critical points x of the distance function associated with P we set:

$$\phi(t,x) = x, \;\text{ for all } t \in [0,\infty)$$

Otherwise let y be the driver of x and R be the ray originating at x and shooting in the direction $x-y$. Let z be the first point on R whose driver is different from y. Note that such a z need not exist in $\mathbb{R}^3$. In this case let z be the point at infinity. We set:

$$\phi(t,x) = x + t\frac{x-y}{||x-y||}, \; t \in [0, ||z-x||]$$

For $t > ||z-x||$ the flow is given as follows:

$$\begin{aligned} \phi(t,x) &= \phi(t - ||z-x|| + ||z-x||, x) \\ &= \phi(t - ||z-x||, \phi(||z-x||, x)) \end{aligned}$$

It is not completely obvious, but ϕ can be shown to be well defined on the whole of $[0,\infty) \times \mathbb{R}^3$. Furthermore, the following two properties of dynamical systems hold for induced flows, first $\phi(0,x) = x$ and second $\phi(t,\phi(s,x)) = \phi(t+s,x)$. Furthermore, as in the smooth case the fixpoints of ϕ are exactly the critical points of h. Thus we can speak of the minima, maxima and saddles of ϕ.

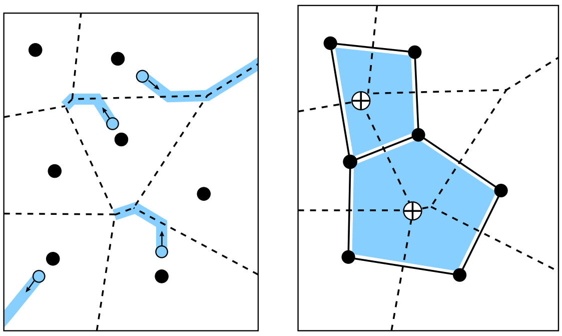

Figure 3: *Left: Some orbits of the flow induced by the sample points from Figure 1. Right: The two regions that flow in the maxima of the height function induced by the same set of points.*

4. Flow complex

The flow induced by a finite set of sample points P describes how the points of $\mathbb{R}^3$ move in time. For any point $x \in \mathbb{R}^3$ the orbit ϕ_x describes this motion. Similar to the smooth case the orbits are integral curves to the vector field of steepest ascent of the distance function h. Orbits end in the fixpoints of ϕ, i.e. in the critical points of h. Given a critical point x of h we are interested in the set of all points whose orbit ends in x, i.e. the set of all points that flow into x. This set is called the *stable manifold* of x. The stable manifold of a minimum is easy to determine it is just the minimum itself. In the following we will also characterize the stable manifolds of the critical points with positive index. Later we will use the stable manifolds of a subset of the index 2 saddles to design a surface reconstruction algorithm. As a minor technical detail we should mention that we are going to deal with smoothed versions of the stable manifolds. For all practical purposes this means taking the topological closure.

Index 0 critical points. We already mentioned that the stable manifold of a local minimum consists only of the minimum itself, i.e. it is a single point.

Index 1 critical points. An index 1 saddle is the intersection point of a Delaunay edge and its dual Voronoi facet. The stable manifold of such a saddle is the Delaunay edge that contains the saddle. Remember that such Delaunay edges are Gabriel edges and vice versa. Thus the union of all stable manifolds of index 1 saddles is the Gabriel graph of the set P of all sample points.

Index 2 critical points. The stable manifolds of index 2 saddles are piecewise linear surfaces. An index 2 saddle s is the intersection point of a Delaunay facet F and its dual Voronoi edge E. We show how to construct the surface explicitly. We start by constructing a polygon P whose interior points all flow into s. This polygon contains s and is contained itself in F. Under mild assumptions there are three Voronoi facets incident to every Voronoi edge. We are going to construct a polyline for each of the three Voronoi facets incident to E. These three polylines together make up the boundary of the polygon P. The drivers of the Voronoi facets incident to E are points on their dual Delaunay edges. These Delaunay edges are all in the boundary of F. Note that it is possible that such a driver is a saddle of index 1. First, consider a driver d which is not a saddle of index 1. The line segment that connects d with s is contained in F and intersects the boundary of the corresponding Voronoi facet in two points, namely in s and in a second point s'. We get a polyline from the two segments that connect s' to the two Delaunay vertices incident to the Delaunay edge that contains d. Second, if the driver of the Voronoi facet is a saddle of index 1 we take its dual Delaunay edge as the polyline. That is, we get three polylines all contained in F, one for each Voronoi facet incident to E. Let P be the polygon whose boundary is given by these polylines. P is contained in F and all its interior points flow into s. It can be triangulated by connecting s with the points s' and the Delaunay vertices incident to F. Figure 4 shows two examples of two such polygons P.

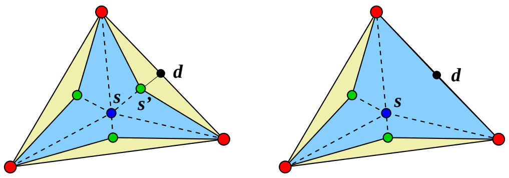

Figure 4: *Two examples of polygons that are contained in a Delaunay triangle that is intersected by its dual Voronoi edge in s. The interior points of these polygons flow into s. The polygon in the figure on the right has one index 1 critical point on its boundary.*

Let s' be a point as constructed above for a Voronoi facet that is not driven by a saddle of index 1. By construction s' is contained in a Voronoi edge E'. Furthermore, by our assumption it has to be an interior point of E'. We can assume again that E' is incident to three Voronoi facets. For one of these Voronoi facets we have already computed a polyline. For the remaining two we do it exactly the same way we did it above for P. Thus we have again three polylines, one for each Voronoi facet incident to E'. Two of these polylines always intersect in a common Delaunay vertex. That is, the three polylines together form a polyline which is homeomorphic to to the circle $\mathbb{S}^1$. The latter polyline need not be contained in a hyperplane but it can be triangulated by connecting the point s' with newly computed points s' and to the Delaunay vertices incident to the Delaunay facet dual to E'. This gives us a new triangulated surface patch whose interior points all flow into s.

We continue with the above construction until there are no more points s' left for which we have not already constructed a surface patch. The surface of points that flow into the index 2 saddle s is made up from all the patches. By construction the boundary of this surface consists of Gabriel edges, i.e. Delaunay edges that contain an index 1 saddle. Figure 5 shows an example of the stable manifold of some index 2 saddle. The stable manifold in this example is made up from five surface patches.

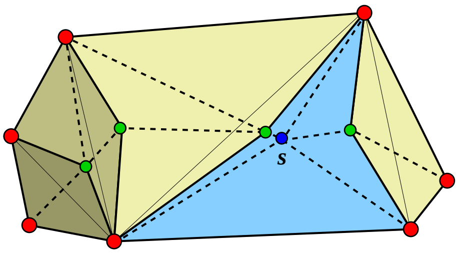

Figure 5: *In this example the stable manifold of s is made up from five surface patches. Note that the surface patches need not be planar.*

Flow complex. Given a finite set of points in $\mathbb{R}^3$. The stable flow complex of the point set is given by the simplicial complex build by the Gabriel graph and the triangulated surfaces whose points flow into index 2 saddles.

Index 3 critical points. The critical points of index 3 are the local maxima of the distance function h. We can prove the following theorem which completely characterizes the stable manifolds of the local maxima.

Theorem 2 The stable manifolds of the local maxima of the distance function h associated with a finite set P of sample points in $\mathbb{R}^3$ are the bounded regions of the flow complex.

Note that analogous statements also hold for critical points of smaller index, i.e. the boundary of a stable manifold of an index d critical point, $1 \leq d \leq 3$, is made up from stable manifolds of index $d-1$ critical points. That is, there exists a nice recursive structure. So far we used this structure only to characterize the stable manifolds of the local maxima by stable manifolds of index 2 critical points. Later we will use this structure to transform the flow complex into a manifold.

Another observation is that the stable manifold of an index d saddle is always d-dimensional.

The flow complex is at the heart of our reconstruction algorithm. It contains the essential information of the stable manifolds of all critical points. We show next how to compute the triangles of the flow complex associated with a finite set P of sample points. The following algorithm is derived directly from our characterization of the stable manifolds of index 2 saddles.

```
FLOWCOMPLEX($P$)
 1  $F := \emptyset$
 2  compute the Voronoi- and Delaunay diagram of $P$.
 3  compute the set $S$ of index 2 saddles.
 4  for each $s \in S$ do
 5    $f :=$ Delaunay facet that contains $s$.
 6    $Q := \emptyset$
 7    for each Delaunay edge $e$ incident to $f$ do
 8      $Q$.push( $(s,e)$ )
 9    end for
10    while $Q \neq \emptyset$
11      $(v,e) := Q$.pop
12      $u,w :=$ endpoints of $e$.
13      if $e$ contains a saddle of index 1 do
14        $F := F \cup \{uvw\}$
15      else do
16        $f :=$ Voronoi facet dual to $e$.
17        $d :=$ driver of the interior points of $f$.
18        $v' :=$ first point on the ray from $d$ to $v$
              that is contained in $f$.
19        $F := F \cup \{vv'u, vv'w\}$
20        $f' :=$ Delaunay facet dual to the Voronoi
              edge that contains $v'$.
21        for each edge $e'$ incident to $f'$ besides $e$ do
22          $Q$.push( $(v',e')$ )
23        end for
24      end if
25    end while
26  end for
27  return $F$
```

The algorithm FLOWCOMPLEX takes a finite set P of points in $\mathbb{R}^3$ as input and works as follows: In line 1 a set F that is used to store the triangles of the flow complex is initialized with the empty set. In line 2 the Voronoi and Delaunay diagrams of P are computed. These are used in line 3 to compute the index 2 saddles of the flow induced by P. Lines 4 and 26 enclose the main loop of the algorithm. In this loop all smoothed stable manifolds of all index 2 saddles are computed. When we characterized these manifolds we have seen that they are piecewise linear surfaces. By definition the triangles of the flow complex are just the collection of all triangles in such manifolds. The loop goes through all saddles of index 2. In line 5 the Delaunay facet that contains the saddle s is computed. In line 6 a queue Q is initialized with the empty set. This queue is going to store tuples of of points and Delaunay edges. Lines 10 and 25 enclose the loop in which the smoothed stable manifold of s is computed. While Q is not empty we pop its first element (v,e) in line 11. If the Delaunay edge e with endpoints u and w contains a saddle of index 1 we insert in line 14 the triangle with corner points u, v and w into F. Otherwise we execute lines 16 to 18. In line 17 we compute the driver d of the interior points of the Voronoi facet f dual to e. It can be shown that all these points have the the same driver. In line 18 we determine the first point v' on the line segment from d to v that is contained in f. In line 19 the triangles $vv'u$ and $vv'w$ are inserted into F. In lines 20 to 23 the Delaunay facet dual to the Voronoi edge that contains v' is computed and for every edge e' incident to f' besides e the tuples (v', e') are pushed to Q. Finally F is returned in line 27.

5. Pairing and cancellation

Edelsbrunner [8] introduced the concept of pairing and cancellation of critical points for surface simplification in a slightly different context. Here we adopt his ideas to tranform the flow complex associated with a sample P into a manifold. Therefore we make use of the recusive structure of the stable manifolds of the critical points. That is, we want to use the fact that the boundary of a stable manifold of an index d critical point, $1 \leq d \leq 3$, is made up from stable manifolds of index $d-1$ critical points.

We exploit the recursive structure of stable manifolds by building pairs of index 2 saddle points and local maxima. An index 2 saddle point a and a local maximum build a pair (a,b) if the stable manifold of a is contained in the boundary of the stable manifold of b. Note that a saddle of index 2 is always paired with two maxima one of which might be a maximum at infinity.

Next we are going to describe how to cancel pairs of index 2 saddles and local maxima. Let (a,b) be such a pair and let c be the second maximum paired with a. Cancellation of (a,b) does not just mean its removal. The idea behind the cancellation of (a,b) is to merge the stable manifolds of b and c. These manifolds share the stable manifold of a as a common boundary. In the cancellation process we remove the common boundary of these stable manifolds and consider the resulting cell as the new stable manifold of c. The index 2 saddles different from a that were paired with b but not with c now have to be paired with c since their stable manifolds are now contained in the new boundary of the stable manifold of c. Instead of pair we will use the term *valid pair* in the following. A pair of an index 2 saddle a and a local maximum b is valid if the stable manifold of s is contained in the boundary of the stable manifold of b. We keep the notion of valid pairs also after cancellation for merged stable manifolds of local maxima. Cancellation of a valid pair (a,b) means:

1. For every index 2 saddle s such that (s,b) is a valid pair declare (s,b) invalid. If also (s,c) is valid declare it invalid. Otherwise declare (s,c) valid.
2. Remove the stable manifold of the saddle a from the flow complex induced by the sample points.

Figure 6 shows an one-dimensional example for pairing and cancellation. The basic ideas remain valid in one dimension, but since there are no saddle points in one dimension local minima and local maxima are paired in this example.

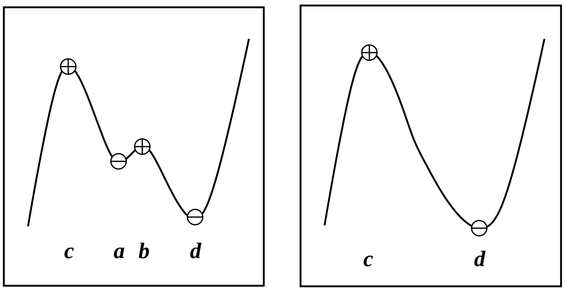

Figure 6: *Left: An one-dimensional example for a pairing of critical points according to the height function h. In this example local minima and local maxima are paired. The local minima ⊖ are at positions a and d and the local maxima ⊕ are at positions b and c. The valid pairs in this configuration are (a,b), (a,c) and (d,b). Right: The critical points from after the cancellation of the local minimum at position a with the local maximum at position a. The pairs (a,b), (a,c) and (d,b) are declared invalid and the pair (d,a) is declared valid instead.*

Of course it makes no sense to cancel valid pairs in arbitrary order nor does it make sense to cancel every valid pair. Since we want to use the cancellation process to tranform the flow complex of some sample P into a manifold we cancel a valid pair (a,b) only if there exists a Gabriel edge in the boundary of the stable manifold of a which is incident to the stable manifolds of at least three maxima. That is, we stop cancelling valid pairs if there is no valid pair left that fulfills this condition.

The order in which we cancel valid pairs is given by difference of the distance function evaluated at the two points in the pair. Let (a,b) a valid pair and h the distance function induced by P. If $h(a)$ and $h(b)$ do not differ much then the local maximum b is relatively close to the boundary of its stable manifold. The deeper a local maximum lies in its stable manifold the better this stable manifold captures the shape represented by the sample. Thus we always cancel the valid pair (a,b) on which the height function h differs the least provided this pair fulfills the topological constraint described above, i.e. $h(b)-h(a)$ is minimal among all valid pairs that fulfill the topology constraint.

Reduced flow complex. Given a flow complex. We transform this flow complex by cancelling valid pairs as long as this is possible. The complex we get after all these cancellations is called the *reduced flow complex*.

The reduced flow complex is almost the reconstruction we are looking for. In fact we made the following observation.

Observation 1 The reduced flow complex of a well sampled solid is the surface of some solid.

However, it can happen that the surface meets itself in a point or curve. Figure 7 shows a such a locally non-regular situation. It can be resolved by cutting the surface along the curve, i.e. doubling the curve and moving the two resulting caps a little apart. Resolving these non-regularities finally provides us with a topologically correct surface which is our reconstruction.

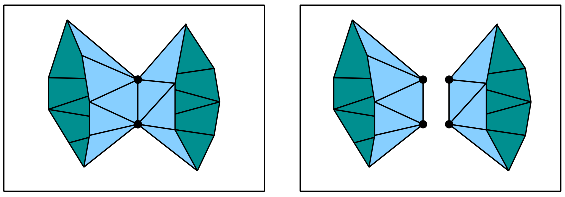

Figure 7: *Left: A surface that meets itself in a curve (line segment) is shown. Right: The same surface after cutting along the curve.*

The procedure we have described above directly leads to an algorithm to compute the reduced flow complex. Here we give a pseudocode formulation of this algorithm.

```
REDUCEDFLOWCOMPLEX($P$)
 1  compute the flow complex $F$ of $P$.
 2  compute the set $S$ of index 2 saddles.
 3  compute the set $V$ of valid pairs of $F$.
 4  while exists pair in $V$ that fulfills the topology
            constraint do
 5    $(a,b) :=$ pair in $V$ that fulfills the topology
                constraint and minimizes the
                distance value difference.
 6    remove the stable manifold of $a$ from $F$.
 7    $c :=$ local maximum such that $(a,c) \in V$ and $c \neq b$.
 8    for each $s \in S$ do
 9      if $(s,b) \in V$ do
10        $V := V - \{(s,b)\}$
11        if $(s,c) \in V$ do
12          $V := V - \{(s,c)\}$
13        else
14          $V := V \cup \{(s,c)\}$
15        end if
16      end if
17    end for
18  end while
19  return $F$
```

The algorithm REDUCEDFLOWCOMPLEX takes a finite set P of points in $\mathbb{R}^3$ as input and works as follows: In line 1 we compute the stable flow complex F of P, e.g. by calling the algorithm FLOWCOMPLEX. In lines 2 and 3 we compute the set S of index 2 saddles and the set V of valid pairs, respectively. The main loop of the algorithm is enclosed by lines 4 and 18. In this loop we cancel a valid pair as long as there exists a pair in F that fulfills the topology constraint that we introduced above. In line 5 we determine the valid pair (a,b) that we will cancel. In line 6 we remove the stable manifold of a from F. The second local maximum c that forms a valid pair together with a is determined in line 7. In the loop enclosed by lines 8 and 17 the set S of valid pairs is updated. If an index 2 saddle s and the local maximum b form a valid pair the this pair is removed from V in line 10. If the pair (s,c) is also valid it will be removed from V in line 12. Otherwise (s,c) is inserted into V in line 14. Finally we return the modified flow complex F in line 19.

The reconstruction that we want to propose here essentially computes the reduced flow complex and resolves topological non-regularities which we found to be very rare.

```
RECONSTRUCTION($P$)
 1  compute the reduced flow complex $F$ of $P$.
 2  resolve all topological non-regularities in $F$.
 3  return $F$.
```

The algorithm RECONSTRUCTION takes a finite set P of sample points as input. It computes the reduced flow complex F in line 1 and afterwards resolves all topological non-regularities of F in line 2.

Figure 8 demonstrates the whole reconstruction process.

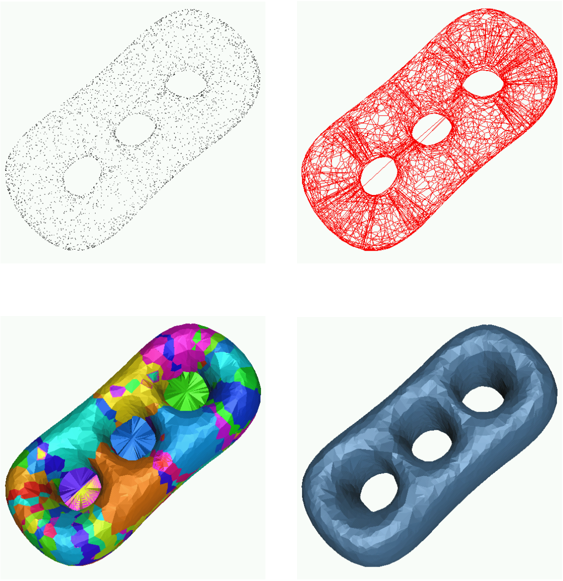

Figure 8: *Top - left: The dataset* 3HOLES. *Top - right: The Gabriel graph of the same dataset. Bottom - left: The flow complex of the dataset (the boundaries between stable manifolds of different maxima are colored differently). Bottom - right: The reduced flow complex.*

6. Implementation and results

We implemented the algorithm RECONSTRUCTION using C++. Our implementation is based on the Computational Geometry Algorithms Library *CGAL* [7] which includes fast and robust Delaunay triangulations in three dimensions.

We tested our algorithm on several examples. All tests were performed on a 480 Mhz Sun Ultra Sparc II. Screenshots of some reconstructed models can be found in Figures 8 and 10. We used *Geomview* [9] for the rendering of the models. In Table 1 we summarize additional data of the shown models.

Model	Points	Triangles	Non-reg's	Time
3HOLES	12008	24024	0	2.32
KNOT	30000	60000	0	13.03
TEETH	87415	174826	1	19.3
OILPUMP	92789	185574	0	19.99
DRAGON	299410	598820	2	82.74
HIP	397386	794772	0	113.74
BUDDHA	433396	866832	9	118.35

Table 1: *Basic data for several datasets that we reconstructed. In the column named "Non-reg's" we report the number of non-regularities encountered in the reduced flow complex. The running times are given in seconds. The datasets* HIP *and* TEETH *are courtesy of Cyberware Inc. The datasets* BUDDHA *and* DRAGON *were taken from the Stanford graphics repository.*

One of our findings was that non-regularities are not very frequent. We explicitly checked the models for topological correctness before and after resolving non-regularities. An indication for the topological correctness of the reconstructed models is that they all satisfy Euler's formula

$$\#\,\text{triangles} \;=\; 2\cdot\#\,\text{vertices} + 4\cdot(\text{genus} - 1).$$

Satisfying Euler's formula is only a necessary condition for topological correctness. So we also checked if every point in a model has a neighborhood homeomorphic to an open disc which was always the case.

The output of our algorithm is visually pleasing. Furthermore it seems to be quite robust against noise and undersampling. For noisy datasets the visual quality of the reconstruction suffers of course, but the algorithm could still produce a fairly good and topologically correct reconstruction. See Figure 9 for an example.

7. Conclusion

We presented an efficient and robust algorithm for surface reconstruction from unorganized sample points in three dimensions. The algorithm is based on a dynamical system induced by the sample points. We think that the ideas which lead us to the formulation of this dynamical system can be useful also for other applications in sample based modeling. Especially we want to explore how one can make use of these ideas for sample decimation and feature extraction.

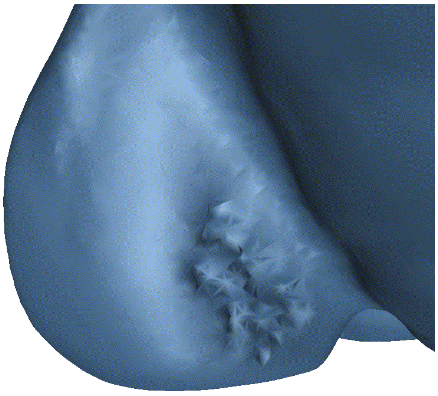

Figure 9: *The dataset* DRAGON *contains some noise at one the legs. This noise does not disturb our reconstruction algorithm to the extent that it fails to produce a reconstruction, but the visual quality suffers of course.*

The tests we performed with our implementation of the reconstruction algorithm showed that it is quite robust against undersampling and noise.

References

1. N. Amenta and M. Bern. Surface reconstruction by Voronoi filtering. *Discr. Comput. Geom.*, **22**, pp. 481–504, (1999) 1

2. N. Amenta, M. Bern and M. Kamvysselis. A new Voronoi-based surface reconstruction algorithm. *Proc. SIGGRAPH 98*, pp. 415-421, (1998) 1

3. N. Amenta, S. Choi, T.K. Dey and N. Leekha. A simple algorithm for homeomorphic surface reconstruction. In *Proc. 16th. ACM Sympos. Comput. Geom.*, pp. 213–222, (2000) 1

4. N. Amenta, S. Choi, R.K. Kolluri. The power crust, unions of balls, and the medial axis transform. *Comp. Geom.: Theory and Applications*, **19**(2-3), pp. 127-153, (2001) 1

5. M. Attene and M. Spagnuolo. Automatic surface reconstruction from point sets in space. *Computer Graphics Forum, Proceedings of Eurographics*, **19**(3), pp. 457-465, (2000) 1

6. J.D. Boissonnat and F. Cazals. Smooth shape reconstruction via natural neighbor interpolation of distance

functions. In *Proc. 18th ACM Sympos. Comput. Geom.*, pp. 223–232, (2000) 1

7. Computational Geometry Algorithms Library `http://www.cgal.org` 8

8. H. Edelsbrunner, J. Harer and A. Zomorodian. Hierarchical Morse Complexes for Piecewise Linear 2-Manifolds. In *Proc. 17th ACM Sympos. Comput. Geom.*, pp. 70–79, (2001) 6

9. `http://www.geomview.org` 8

10. M. Gopi, S. Krishnan, C.T. Silva. Surface Reconstruction based on Lower Dimensional Localized Delaunay Triangulation. *Computer Graphics Forum, Proceedings of Eurographics*, **19**(3), pp. 467-478, (2000) 1

11. H. Hoppe, T. DeRose, T. Duchamp, J. McDonald and W. Stuetzle. Surface reconstruction from unorganized points. In *Proc. SIGGRAPH 92*, pp. 71-78, (1992) 1, 2

12. R. Mencl. A graph-based approach to surface reconstruction. *Computer Graphics Forum, Proceedings of Eurographics*, **14**(3), pp. 445-456, (1995) 1

13. R. Mencl and H. Müller. Graph-Based Surface Reconstruction Using Structures in Scattered Point Sets. In *Proc. CGI'98*, (1998) 1

14. H.K. Zhao, S. Osher, B. Merriman and M. Kang. Implicit and non-parametric shape reconstruction from unorganized points using variational level set method. *Computer Vision and Image Understanding*, **80**(3), pp. 295-319, (2000) 1

15. H.K. Zhao, S. Osher and R. Fedkiw, Fast Surface Reconstruction using the Level Set Method. In *Proc. 1st IEEE Workshop on Variational and Level Set Methods, in conjunction with 8th Int. Conf. on Comp. Vision (ICCV)*, pp. 194-202, (2001) 1

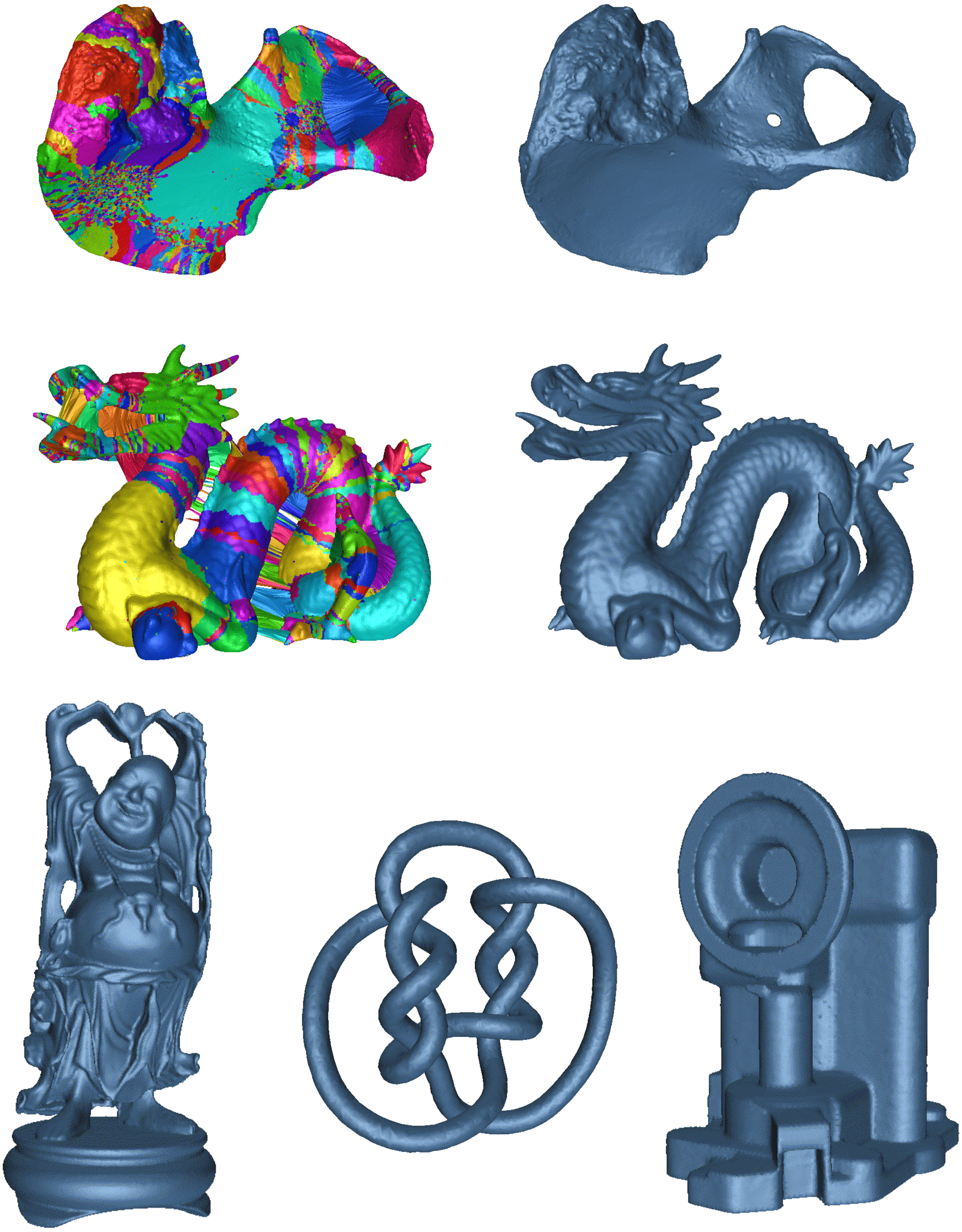

Figure 10: *Screenshots of the flow complex and the reduced flow complex of several models.*