Approximating Bounded, Non-orientable Surfaces from Points

Anders Adamson Marc Alexa
Department of Computer Science, Darmstadt University of Technology
Fraunhoferstr. 5, 64283 Darmstadt
aadamson@gris.informatik.tu-darmstadt.de

Abstract

We present an approach to surface approximation from points that allows reconstructing surfaces with boundaries, including globally non-orientable surfaces. The surface is defined implicitly using directions of weighted co-variances and weighted averages of the points. Specifically, a point belongs to the surface, if its direction to the weighted average has no component into the direction of smallest covariance. For bounded surfaces, we require in addition that any point on the surface is close to the weighted average of the input points. We compare this definition to alternatives and discuss the details and parameter choices. Points on the surface can be determined by intersection computations. We show that the computation is local and, therefore, no globally consistent orientation of normals is needed. Continuity of the surfaces is not affected by the particular choice of local orientation. We demonstrate our approach by rendering several bounded (and non-orientable) surfaces using ray casting.

Keywords: *point-sampled geometry, ray-surface intersection, manifold, boundary, non-orientable*

1 Introduction

Point sets without additional topological information are a popular representation of surfaces (see [3, 16, 32]). Real world shapes are sampled using scanners resulting in an initial surface representation consisting of points [21, 27]. Points are also a suitable primitive for the display of surfaces, especially as the complexity of the shape results in triangles being rendered into less than a pixel for the final display [26, 28, 18, 11, 10, 33].

For many modeling tasks, however, a (local) approximation of the surface based on the points is required. A variety of methods for the reconstruction of surfaces from points exist. Some of these are approaches are global in nature, others allow locally computing surface parts but are based on locally approximating a global implicit function.

All approaches seem to assume that the represented surface is a solid and, thus, unbounded and globally orientable. This might stem from the fact that point clouds are mostly acquired by scanning real-world (solid) objects, so that areas lacking points are considered inadequately sampled and to be fixed. However, with the rising interest in unstructured point sets as a general surface representation, we feel it is important to consider surfaces with boundary explicitly. Furthermore, enhancing reconstruction to faithfully deal with bounded, possibly non-orientable surfaces, increases the robustness against sampling errors and makes the algorithm more versatile.

We extend our framework for approximating surfaces from scattered points [1] to handle smooth boundaries and show that the surface could be (globally) non-orientable. Specifically, we make the following contributions:

Boundary definition: The surface is defined implicitly using compactly supported functions. We show how to compute and define a smooth and natural boundary using only the quantities we compute during surface approximation.

Boundary behavior and parameters: We prove several properties about the behavior of the computations involved in the estimation of the boundary. We use these approximations to solve the problem of parameter estimation and provide the user with an interval of reasonable parameters to choose from.

Orientation: The computation procedure for points requires only the intersection of a curve/ray with a plane. As the orientation of the plane's normal is irrelevant, a global orientation of normals is not required. We show that the continuity of the surface is not affected.

Despite these features, the point set could still define a solid and our implicit definition of the surface would allow defining inside and outside. Note that we cannot handle non-manifold surfaces, e.g., surfaces with self intersections. In the following, we briefly review some of the related work, define the surface and explain how to compute it efficiently,

Proceedings of the Shape Modeling International 2004 (SMI'04)
0-7695-2075-8/04 $20.00 © 2004 **IEEE**

before we explain in detail the definition of the boundary and how to choose the necessary parameters.

2 Related Work

A variety of works tackle the (ill-posed) problem of reconstructing a surface from an unstructured set of points.

Hoppe et al. [17] define a signed distance field from the points. For each point a normal direction is estimated and the normal is oriented. The signed distance to the surface is defined as the normal component of the distance to the closest point.

In a sense the surface definition of Hoppe et al. is related to Voronoi-based reconstruction techniques [8, 13, 4, 15, 6, 5] as the surface is defined by a closest point relationship. These techniques define the surface as a subset of the Delaunay triangulation of the point set. As such, they are all global methods and generate a C^0 surface.

Surfaces with higher order continuity from points are typically implicit. For the specific purpose of consolidating registered range images Curless et al. [12] used a discrete representation (see also [23] for a comparable approach). Radial basis functions with global [29, 9, 31] or local [22, 19, 25] support are a more general technique for approximating functions from scattered constraints. Note that all of these techniques are inherently global in nature – some of them explicitly consider the case of incomplete point sampling and explain how, despite the missing points, a globally unbounded and oriented surface would be reconstructed.

Hierarchical methods based on blending individual local implicit approximations such as [24] as well as moving least squares type surface approximation [20, 3, 14] are local representations. While the MLS-based approaches explicitly require the surface to be sampled sufficiently dense from a closed manifold, hierarchical methods are rather concerned with the problem of adequately filling un-sampled regions.

Schaufler and Jensen [30] define the intersection of a ray and the point set as follows: In each point a disk is constructed using the point normal. A cylinder around the ray is intersected with the disks. The intersection is computed as a weighted average of disks whose centers are inside the cylinder. Note that the reconstructed surface could have boundary, because not every cylinder contains points. On the other hand, the boundary of the surface would depend on the particular rays chosen.

Our approach is also based on intersecting rays with the surface [1], however, it could be understood as a robust computation of a particular implicit surface (which is identical to the MLS definition [20]). Implicit surfaces typically represent solids. According to Bloomenthal et al. [7], boundaries are generally represented by defining additional functions on the space and requiring that these functions are, e.g., positive. Our approach could, indeed, be interpreted in this way.

3 Surface definition

We first give a brief definition of the surface as defined in [1] and establish several notations for later use.

Given points $\mathcal{P} = \{\mathbf{p}_i \in \mathbb{R}^3, i \in \{1, \ldots, N\}\}$ on or close to a surface $\mathcal{S}$. The point set surface $\mathcal{S}_\mathcal{P}$ is defined as the zero-set of the implicit function

$$f(\mathbf{x}) = \mathbf{n}(\mathbf{x}) \cdot (\mathbf{x} - \mathbf{a}(\mathbf{x})), \tag{1}$$

which describes the distance of $\mathbf{x}$ to a plane with normal $\mathbf{n}(\mathbf{x})$ through the weighted average of points $\mathbf{a}(\mathbf{x})$. For the surface definition to be reasonable (i.e. the surface is continuously differentiable), we require the mappings defining normal direction $\mathbf{n} : \mathbb{R}^3 \to \mathbb{S}^2$ and weighted average $\mathbf{a} : \mathbb{R}^3 \to \mathbb{R}^3$ to be smooth in their input at least in a sufficient neighborhood Ω around $\mathcal{P}$. We will give particular choices for the mappings and the neighborhood below. These choices are useful for efficient and local computation of the surface, and they lead to natural definition of boundaries.

Let $\theta : \mathbb{R} \to \mathbb{R}$ be a monotone decreasing function with local support r_θ, which is strictly positive inside the support, i.e. $\theta(\chi) = 0 \Leftrightarrow \chi > r_\theta$. We will use θ as a weight function determining the influence of a point $\mathbf{p}_i$ on a point in space $\mathbf{x}$. The neighborhood Ω is consequently defined as the region in space that is affected by the points:

$$\Omega = \left\{ \mathbf{x} | \sum_i \theta(||\mathbf{p}_i - \mathbf{x}||) > 0 \right\} \tag{2}$$

Alternatively, we can define the distance of a point $\mathbf{x}$ to the point set as the minimum distance of $\mathbf{x}$ to any point in $\mathcal{P}$:

$$d_\mathcal{P}(\mathbf{x}) = \min_i ||\mathbf{x} - \mathbf{p}_i||. \tag{3}$$

Using this definition, Ω could be defined as the set of points with distance $d_\mathcal{P}$ less than r_θ, which also describes the union of r_θ-balls around the points in $\mathcal{P}$. Note that all these definitions of Ω are equivalent because we assume θ to be strictly positive in its support.

The weighted average of points $\mathbf{a}$ is defined (for points in Ω only) as

$$\mathbf{a}(\mathbf{x}) = \frac{\sum_i \mathbf{p}_i \theta(||\mathbf{p}_i - \mathbf{x}||)}{\sum_i \theta(||\mathbf{p}_i - \mathbf{x}||)}, \mathbf{x} \in \Omega. \tag{4}$$

Normal directions are defined as directions of smallest weighted co-variance, using the same weights as before. This could be understand as a least squares fit of a plane with unit normal $\mathbf{n}$ through $\mathbf{x}$, i.e. the minimizer of

$$\min_{||\mathbf{n}||=1} \frac{\sum_i \langle \mathbf{n}, \mathbf{p}_i - \mathbf{x} \rangle^2 \theta(||\mathbf{p}_i - \mathbf{x}||)}{\sum_i \theta(||\mathbf{p}_i - \mathbf{x}||)}. \tag{5}$$

Proceedings of the Shape Modeling International 2004 (SMI'04)
0-7695-2075-8/04 $20.00 © 2004 IEEE

It is well known that this constrained minimization problem can be solved using the eigenvectors of the matrix $W(\mathbf{x}) = \{w_{jk}\}$ of weighted co-variances in the directions of an orthonormal basis of $\mathbb{R}^3$. Let $\mathbf{e}_i, i \in \{0, 1, 2\}$ be such a basis, then the coefficients of the matrix are given as

$$w_{jk} = \sum_i \langle \mathbf{e}_j, \mathbf{p}_i - \mathbf{x} \rangle \langle \mathbf{e}_k, \mathbf{p}_i - \mathbf{x} \rangle \theta(||\mathbf{p}_i - \mathbf{x}||). \quad (6)$$

Let $\{\mathbf{v}_i\}$ be the eigenvectors of $W(\mathbf{x})$ corresponding to the eigenvalues $\lambda_0 \leq \lambda_1 \leq \lambda_2$, we set $\mathbf{n} = \mathbf{v}_0$ for any $\mathbf{x} \in \Omega$. Using this representation it can be deduced that $\mathbf{n}(\mathbf{x})$ is a smooth function on Ω if $\lambda_0 < \lambda_1$ for all $\mathbf{x} \in \Omega$. We call point sets for which this is true *well sampled* (see [1] for more information).

Note that the continuity arguments are independent of the orientation of the normal: The computation is local and inside a ball with sufficiently small radius we could orient the normals consistently so that $\mathbf{n}(\mathbf{x})$ is indeed a smooth vector-valued function of $\mathbf{x}$ inside the ball. If the normals are now inverted inside this ball, the surface defined as $\mathbf{n}(\mathbf{x}) \cdot (\mathbf{x} - \mathbf{a}(\mathbf{x})) = 0$ is unaffected. Since we made no special assumptions on the location of the ball this argument extends to the whole surface.

For θ we typically assume a Gaussian-shaped function

$$\theta(d) = e^{-d^2/h^2}, \quad (7)$$

where h is the anticipated feature size, which roughly corresponds with spacing among points in $\mathcal{P}$. In practice, we compute h as the average Euclidean k-nearest neighbor distance, where we use $k = 6$. To get a locally supported weight function, we approximate the Gaussian by a spline with appropriate local support and continuity. We will, nevertheless, sometimes use the feature size h in our arguments.

4 Computing points on the surface

A ray-surface (or, more generally, curve-surface) intersection approach is used to efficiently compute points of the surface (see [1] for more details). The method is iterative: For a point $\mathbf{x}_i$ on the ray obtained in the i-th iteration, a normal direction $\mathbf{n}(\mathbf{x}_i)$ and a local constant surface approximation through $\mathbf{a}(\mathbf{x}_i)$ are computed. The local surface approximation is then intersected with the ray yielding $\mathbf{x}_{i+1}$. In practice, this iteration quickly converges, if we start with points in the proximity of $\mathcal{S}_\mathcal{P}$.

To determine a good starting point for the iterations efficiently, a set of enclosing balls $\{B_i\}$ is constructed by placing a ball B_i of radius r_B around each input point $\mathbf{p}_i$. The union of the balls

$$\mathcal{B} = \{\mathbf{x} | d_\mathcal{P}(\mathbf{x}) < r_B\} = \bigcup_i B_i, B_i = \{\mathbf{x} : ||\mathbf{x} - \mathbf{p}_i|| < r_B\} \quad (8)$$

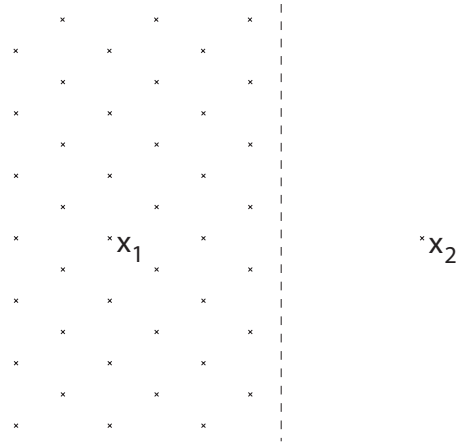

Figure 1. The rationale of defining bounded surface: While all points are in the same plane, we expect $\mathbf{x}_2$ not to part of the reconstructed surface.

is expected to contain the surface, i.e. $\mathcal{S}_\mathcal{P} \subset \mathcal{B}$, which is achieved by choosing the global radius $r_B \approx 1.5h$ (note that this value has been determined experimentally). All potential ray-surface intersections can be found by first intersecting $\mathcal{B}$ (a spatial data structure should be used to accelerate the computation necessary to find the right balls). To evaluate whether a ball B_i contains a ray-surface intersection, the following three steps are repeatedly applied, starting with the intersection $\mathbf{x}_0$ of the ray and B_i:

1. **Support plane:** The matrix of weighted co-variance w.r.t. $\mathbf{x}_i$ is constructed and the eigenvectors are used as a local frame in $\mathbf{x}_i$; the eigenvector corresponding to the smallest eigenvalue is the normal direction $\mathbf{n}(\mathbf{x}_i)$.
2. **Evaluation:** The weighted average $\mathbf{a}(\mathbf{x}_i)$ is computed and $f(\mathbf{x}_i)$ is evaluated. If $|f(\mathbf{x}_i)|$ is sufficiently small, an intersection is found.
3. **Approximation and intersection:** If $|f(\mathbf{x_i})|$ is large, a planar approximation of the surface is constructed by a plane with normal $\mathbf{n}(\mathbf{x}_i)$ and an offset $f(\mathbf{x}_i)$ relative to $\mathbf{x}_i$ in normal direction. The next point on the ray $\mathbf{x}_{i+1}$ is computed by intersecting the ray with that plane. If $\mathbf{x}_{i+1}$ is outside the ball B_i the iterations are stopped and the next intersected ball is examined.

In a typical setting, 3 to 5 iterations are sufficient to determine a point on the surface. Note that due to our choice of weight function there can only be one intersection inside a ball B_i [2].

While we have used the term *normal* in the description of the algorithm, the orientation of this normal is irrelevant.

Proceedings of the Shape Modeling International 2004 (SMI'04)
0-7695-2075-8/04 $20.00 © 2004 **IEEE**

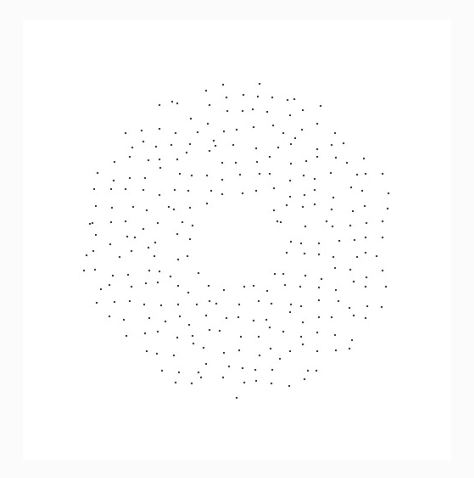

Figure 2. A set of points representing a planar disk with a hole.

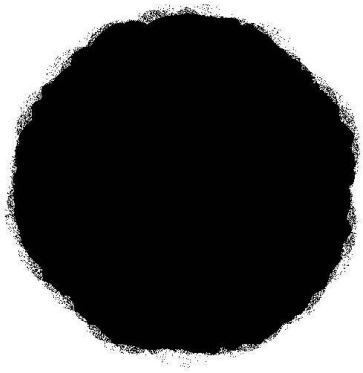

Figure 3. A rendering resulting from ray casting the points shown in Fig. 2. The support radius of the weight functions r_θ is chosen relatively small; the breakdown of the evaluation of points close to $\partial\Omega$ can be clearly observed.

The local constant surface approximation is independent of the normal orientation. Because the ray-surface intersection is computed inside a ball B_i with small radius r_B, the computation is local as required in Section 3.

5 Defining the boundary

For defining a boundary, we illustrate our reasoning using Figure 1: The input points are distributed equally over a half-plane. The two points of evaluation $\mathbf{x}_1$ and $\mathbf{x}_2$ as well as all points in $\mathcal{P}$ are assumed to be co-planar. The direction of smallest weighted co-variance of $\mathbf{x}_1$ and $\mathbf{x}_2$ is, thus, the constant normal to the plane. Thus, $\mathbf{x}_1$ is part of the surface, and $\mathbf{x}_2$ is part of the surface if $\mathbf{x}_2 \in \Omega$.

Remember that the surface is defined only inside Ω, which is bounded because weight functions are locally supported. Using $\partial\Omega$ for the definition of the boundary is not a good idea in practice: the computations involved in evaluating Eq. 1 break down near the boundary of Ω, because the sum of weights in the denominators tends to zero (cf. Eqs. 4 and 5). For illustration purposes, we will use an irregular point set representing a planar disk with a hole (see Figure 2). The numerical breakdown of our current implementation close to $\partial\Omega$ can be clearly observed in Fig. 3, showing a rendering of the surface generated by ray casting.

The numerical problem could be avoided by requiring the sum of weights to exceed a certain threshold, i.e. $\sum_i \theta(||\mathbf{p}_i - \mathbf{x}||) > \epsilon$. This might lead to a definition of boundaries that are robustly computed, however, we feel that this is a bad idea: The boundary would depend on the sampling density. To illustrate this, imagine any point set representing a surface with holes. A given threshold on the sum of weights defines the boundary. Note that this boundary could be arbitrarily shifted towards $\partial\Omega$ by simply adding points in the locations of the existing points.

The use of an enclosing ball structure $\mathcal{B}$ for the computation with radius $r_B < r_\theta$ has been motivated mainly by efficiency arguments in Sec. 4. The numerical breakdown close to $\partial\Omega$ is another reason. In practice, all points in the set of enclosing spheres could be robustly evaluated. One might use $\partial\mathcal{B}$ to define the boundary of $\mathcal{S}_\mathcal{P}$, however, this results in piecewise circular boundary that is only C^0 where circular arcs meet.

We propose a different approach to obtain boundaries, that are smooth, easy to compute, and almost insensitive to variations in the sampling density. Refer again to Figure 1: To distinguish the location of $\mathbf{x}_1$ and $\mathbf{x}_2$ we inspect the relative location of the weighted average $\mathbf{a}$ in the points. Note that the weighted average of points is necessarily contained in their convex hull. Thus, for points far away from the point set the distance $||\mathbf{x} - \mathbf{a}(\mathbf{x})||$ increases, while we expect this distance to be rather small for locations close to (or "inside") the points. For further use we will call

$$c(\mathbf{x}) = ||\mathbf{x} - \mathbf{a}(\mathbf{x})|| \tag{9}$$

the *off-center* value of $\mathbf{x}$.

The main idea for defining a boundary is to require the off-center value to be less than a user-specified threshold.

Proceedings of the Shape Modeling International 2004 (SMI'04)
0-7695-2075-8/04 $20.00 © 2004 **IEEE**

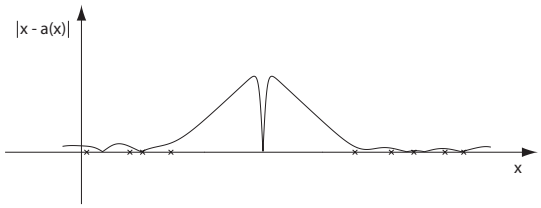

Figure 4. A plot of $c(\mathbf{x}) = ||\mathbf{a}(\mathbf{x}) - \mathbf{x}||$ for a slightly noisy point set with a gap.

More precisely, the surface $\mathcal{S}_\mathcal{P}$ is defined as

$$\mathcal{S}_\mathcal{P} = \{\mathbf{x} \in \Omega | f(\mathbf{x}) = 0 \wedge c(\mathbf{x}) < \epsilon_c\} \qquad (10)$$

In the following section we will discuss reasonable choices for ϵ_c.

6 Choosing the off-center limit

For the following discussion we use a series of plots of $c(\mathbf{x})$ in one dimension and corresponding renderings produced with ray casting and color coding $c(\mathbf{x})$ of the disk point set shown in Figure 2. The input points in one dimension could be interpreted as a one-dimensional slice through the disk point set: The points result from regularly sampling the line, adding Gaussian noise to the locations, and removing some points to generate a gap. We assume the space inside the gap/hole is in Ω, i.e. r_θ is sufficiently large. The graph of $c(\mathbf{x})$ in one dimension is shown in Figure 4.

As expected, the off-center value $c(\mathbf{x})$ is small in-between the points (in fact, for equidistant input points far away from the boundary of Ω it is $c(\mathbf{x}) = 0$) and it increases almost linearly far away from the points. This is also reflected in the color coded renderings of the disk (see Figures 3, 5, and 6), where blue indicates small values and red indicates large values of $c(\mathbf{x})$. Note that the off-center value is particularly small in the barycenter of points – including the barycenter of holes. Figure 6 shows what this behavior entails if the radius of the sphere structure r_B is relatively large: Inside the hole a small island appears. Other potential problems are

- holes in areas of the surface that seem to be adequately sampled. This results from a value for ϵ_c that is too small (see Fig. 5a),
- piecewise circular boundaries resulting from a too large value for ϵ_c (see Fig. 5b).

It is clear that we have to define ϵ_c relative to r_B. Before we can do so we need to establish a connection between $c(\mathbf{x})$ and the distance of $\mathbf{x}$ to the point set $d_\mathcal{P}(\mathbf{x})$. We can show (see Appendix A) that

1. $c(\mathbf{x}) > \sqrt{d_\mathcal{P}^2(\mathbf{x}) - h^2(2-\sqrt{3})}$ as $\mathbf{x}$ moves away from the points and
2. $\lim_{d_\mathcal{P}(\mathbf{x})\to\infty} d_\mathcal{P}(\mathbf{x}) = c(\mathbf{x})$ (assuming that θ has very large radius of support).

In other words, $c(\mathbf{x})$ behaves roughly like the distance to the point set, however, is larger when $\mathbf{x}$ moves away from the surface (a notable exception is around the barycenter of holes, where $d_\mathcal{P}$ is maximal, while $c(\mathbf{x})$ is small).

For an exact characterization of ϵ_c we consider two different case for the radius r_B of the enclosing ball structure. First, consider r_B to be chosen in a natural way, i.e. so that a portion of the hole/gap is not covered by $\mathcal{B}$ (Fig. 5. In order to avoid unwanted holes in the surface ϵ_c needs to be chosen large enough, however, $\partial\mathcal{B}$ defines an upper bound on ϵ_c: Because $c(\mathbf{x}) > \sqrt{d_\mathcal{P}^2(\mathbf{x}) - h^2(2-\sqrt{3})}$ and r_B reflects a bound on $d_\mathcal{P}$ we require $\epsilon_c < \sqrt{r_B^2 - h^2(2-\sqrt{3})}$.

To bound ϵ_c from below, we need to find a reasonable maximum of $c(\mathbf{x})$ in cases where we would consider the surface closed. In our setting, the only reasonable definition of a closed surface is to assume it is closed where it is contained in $\mathcal{B}$. Thus, the case where the enclosing ball structure just covers a hole is the relevant limiting case (see Fig. 6 for an illustration). In order to avoid the island (which could as well be seen as an unwanted hole in the surface), ϵ_c has to be larger than the maximum of $c(\mathbf{x})$ in this region.

Note that because $c(\mathbf{x})$ is larger than $d_\mathcal{P}(\mathbf{x})$ as we move away from the boundary of a hole, it is **not** necessary that an $\epsilon_c < r_B$ that is larger than $c(\mathbf{x})$ in small gaps (i.e. those that are covered by $\mathcal{B}$) exists. To show that such an ϵ_c exists in practice, we need to consider the practical choice of r_B based on h, and the way we determine h from the distribution of the points. In particular, by assuming Gaussian weights of the form $\theta(d) = e^{-d^2/h^2}$ we estimate $c(\mathbf{x})$ to be bounded by h (see Appendix B). Given our practical choice of r_B this leads to the following bounds for ϵ_c:

$$\frac{1+4\sqrt{3}}{9} r_B > \epsilon_c > \frac{2}{3} r_B \qquad (11)$$

Figures 5c and 6b show reasonable values for ϵ_c for both setting of radii r_B.

7 Results

We have tested our approach by ray casting several surfaces represented by unstructured point sets using the ray intersection method described in Section 4. The point sets all represent surfaces with boundaries, and some are globally non-orientable.

In all our examples we have estimated the feature size h by finding the 6 nearest neighbors for all points and then

Proceedings of the Shape Modeling International 2004 (SMI'04)
0-7695-2075-8/04 $20.00 © 2004 **IEEE**

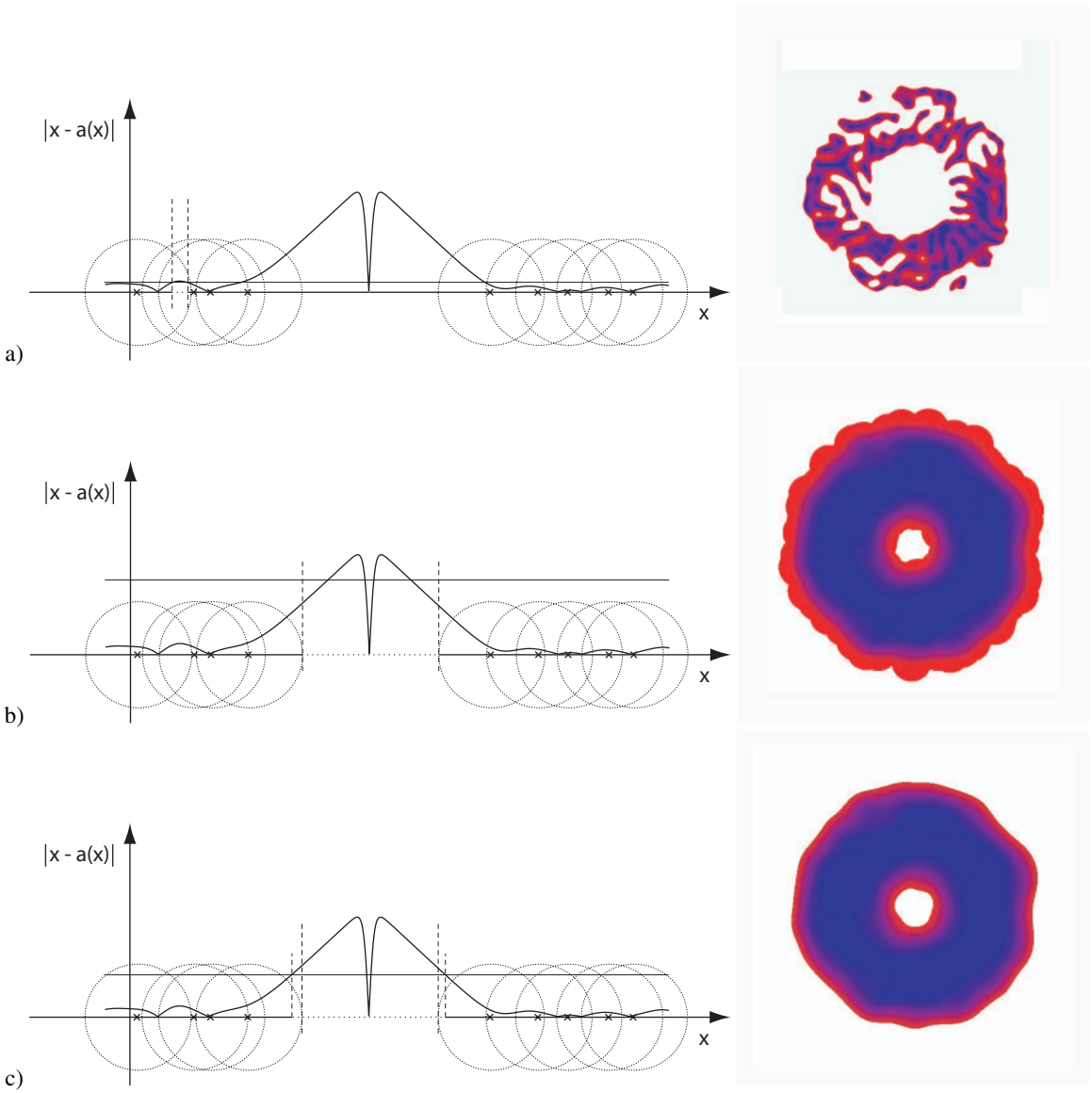

Figure 5. Different choices for the off-center value given a reasonable setting for the radius of the enclosing ball structure. In (a) the off-center value is chosen too small and holes appear in the surface where it is expected to be connected. In (b) the off-center value is too large so that surface is bounded only by the enclosing sphere structure. The off-center value in (c) shows a reasonable choice.

Proceedings of the Shape Modeling International 2004 (SMI'04)
0-7695-2075-8/04 $20.00 © 2004 **IEEE**

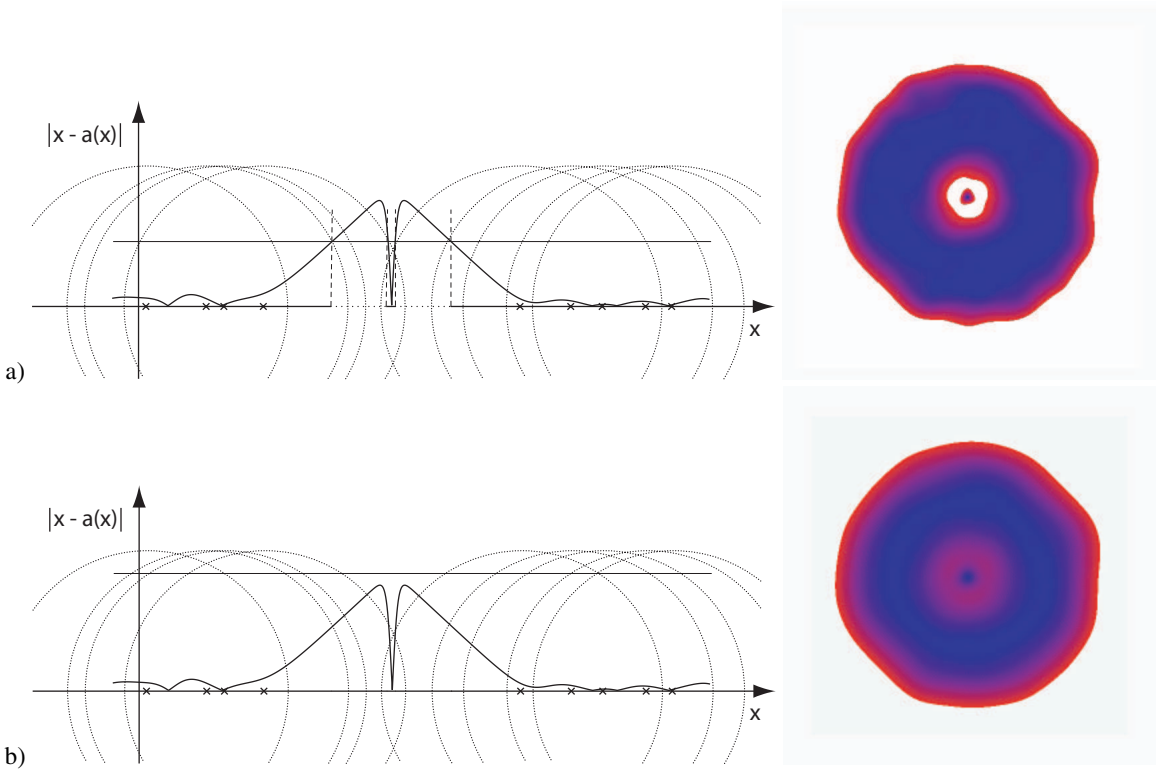

Figure 6. The enclosing ball structure just covers the hole in the point set. If the off-center value is too small (a), an island appears, however, we can estimate a sufficient condition for practical cases (b) so that no islands appear. Note that this entails that surface is closed wherever it is contained in the enclosing sphere structure.

computing the mean distance. The radius r_B of the enclosing sphere structure is set to $1.5h$. The limit for the off-center value is $\epsilon_c = 0.75r_B$ in all examples. Note that we have not yet encountered the problem of islands for these values in practice.

The Stanford Bunny model is known to have several holes on its bottom side. When we render the original point set only the two large holes appear, while the two longish holes are closed. We have up-sampled the original data set using the methods described in [3] so that we can visualize all four holes. The result is depicted in Figure 7. The color coded image reveals that $c(\mathbf{x})$ is large in areas of small sampling density but also in areas of high curvature.

To demonstrate the that the surface could as well be non-orientable we have rendered a Möbius strip and a Klein bottle represented by points (see Figure 8). The point set for the Möbius strip consists of 2.6k points, and the Klein bottle of 33k points. We have cut out a hole in the Klein bottle so that the surface is free of self-intersections. We have used a semi-transparent (non-refracting) material to show also the inner part of the shapes.

8 Discussion & Conclusions

We have presented a framework for approximating point-sampled surfaces with boundaries. This framework builds on the existing tools for intersecting surfaces represented by unstructured point sets [1].

The surface definition given in Section 3 restricts the normal computation to directions of smallest co-variance and the local approximation to a constant. We like to note that our definition of boundaries is not limited to this choice. In fact, any normal computation that is smooth in the input and any local approximation technique could be combined with our definition of boundaries. However, we find the combination as discussed here particularly elegant, as the computation of bounded surfaces requires no extra work.

We have used a constant weight function for the surface.

Proceedings of the Shape Modeling International 2004 (SMI'04)
0-7695-2075-8/04 $20.00 © 2004 **IEEE**

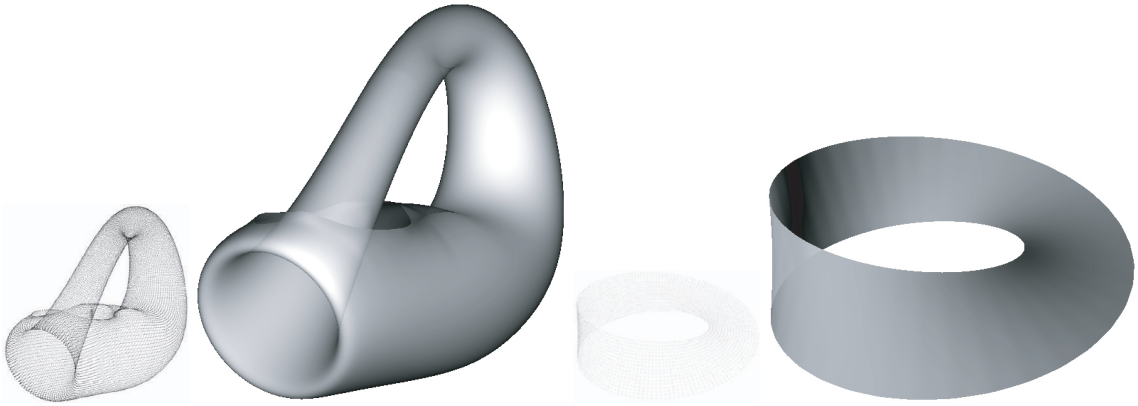

Figure 8. Renderings of non-orientable, bounded surfaces represented by unstructured point sets. Note that we have added a boundary to the Klein bottle where it would have self-intersected.

It might be advantageous to adapt the weight function to the local sampling density. The weight function (or, the feature size h) influences the topological type of the reconstruction. Note that this includes possible holes. Moreover, the complexity of a boundary curve is naturally limited by the chosen feature size, which in practice depends on the sampling density. Thus, the explicit definition of complex boundary curves for moderately complex surfaces also requires an adaptive feature size.

Our framework applies so far only to manifold surfaces. Especially in the example of the Klein bottle, we have cut a hole to avoid the self-intersection of the surface. It is clear, that the case of self-intersection cannot be handled fully automatically, because it is inherently ambiguous. We have hope, however, that with additional information (e.g normal directions in the points) or user intervention also self-intersecting surfaces could be handled faithfully with our approach.

9 Acknowledgments

We thank Nina Amenta and Leo Guibas for helpful comments. The bunny model is courtesy of the Stanford Computer Graphics Laboratory. The insightful comments of the anonymous reviewers are also gratefully acknowledged.

References

[1] A. Adamson and M. Alexa. Approximating and intersecting surfaces from points. In L. Kobbelt, P. Schröder, and H. Hoppe, editors, *Proceedings of the Eurographics Symposium on Geometry Processing*, pages 245–254, June 2003.

[2] A. Adamson and M. Alexa. Ray tracing point set surfaces. In M.-S. Kim, editor, *Proceedings of Shape Modeling International 2003*, pages 272–279. IEEE Computer Society, May 2003.

[3] M. Alexa, J. Behr, D. Cohen-Or, S. Fleishman, D. Levin, and C. T. Silva. Point set surfaces. In *IEEE Visualization 2001*, pages 21–28, October 2001. ISBN 0-7803-7200-x.

[4] N. Amenta, M. Bern, and M. Kamvysselis. A new voronoi-based surface reconstruction algorithm. *Proceedings of SIGGRAPH 98*, pages 415–422, July 1998. ISBN 0-89791-999-8. Held in Orlando, Florida.

[5] N. Amenta, S. Choi, and R. Kolluri. The power crust, unions of balls, and the medial axis transform. *CGTA: Computational Geometry: Theory and Applications*, 19(2–3):127–153, 2001.

[6] M. Attene and M. Spagnuolo. Automatic surface reconstruction from point sets in space. *Computer Graphics Forum*, 19(3), August 2000. ISSN 1067-7055.

[7] J. Bloomenthal, C. Bajaj, J. Blinn, M. P. Cani-Gascuel, A. Rockwood, B. Wyvill, and G. Wyvill. *Introduction to Implicit Surfaces*. Morgan Kaufmann, 1997.

[8] J.-D. Boissonnat. Geometric structues for three-dimensional shape representation. *ACM Transactions on Graphics*, 3(4):266–286, October 1984.

[9] J. C. Carr, R. K. Beatson, J. B. Cherrie, T. J. Mitchell, W. R. Fright, B. C. McCallum, and T. R. Evans. Reconstruction and representation of 3d objects with radial basis functions. In *Proceedings of ACM SIGGRAPH 2001*, pages 67–76, August 2001.

[10] B. Chen and M. X. Nguyen. Pop: a hybrid point and polygon rendering system for large data. In *IEEE Visualization 2001*, pages 45–52, October 2001. ISBN 0-7803-7200-x.

[11] J. D. Cohen, D. G. Aliaga, and W. Zhang. Hybrid simplification: combining multi-resolution polygon and point rendering. In *IEEE Visualization 2001*, pages 37–44, October 2001. ISBN 0-7803-7200-x.

Proceedings of the Shape Modeling International 2004 (SMI'04)
0-7695-2075-8/04 $20.00 © 2004 **IEEE**

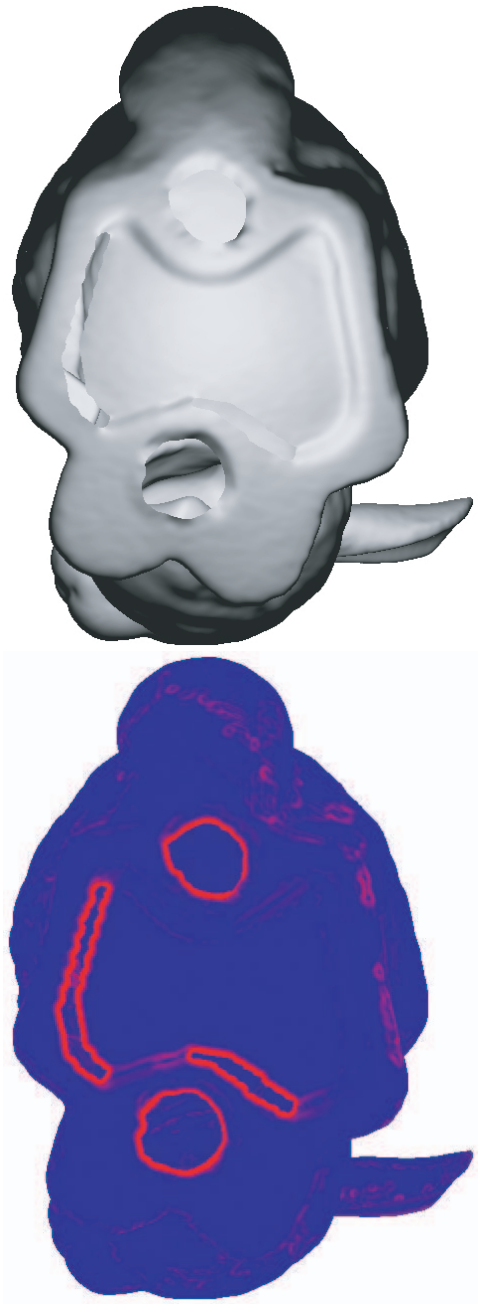

Figure 7. Renderings of the Stanford bunny model showing the holes in the bottom. The first image shows a rendering resulting from ray casting including shadow rays. The second image is color coded depicting the off-center value.

[12] B. Curless and M. Levoy. A volumetric method for building complex models from range images. *Proceedings of SIGGRAPH 96*, pages 303–312, August 1996. ISBN 0-201-94800-1. Held in New Orleans, Louisiana.

[13] H. Edelsbrunner and E. P. Mücke. Three-dimensional alpha shapes. *ACM Transactions on Graphics*, 13(1):43–72, January 1994. ISSN 0730-0301.

[14] S. Fleishman, D. Cohen-Or, M. Alexa, and C. Silva. Progressive point set surfaces. *ACM Transactions on Computer Graphics*, 22(4), 2003.

[15] M. Gopiy, S. Krishnan, and C. T. Silva. Surface reconstruction based on lower dimensional localized delaunay triangulation. *Computer Graphics Forum*, 19(3), August 2000. ISSN 1067-7055.

[16] M. Gross. Are points the better graphics primitives? *Computer Graphics Forum*, 20(3), 2001. ISSN 1067-7055.

[17] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle. Surface reconstruction from unorganized points. *Computer Graphics (Proceedings of SIGGRAPH 92)*, 26(2):71–78, July 1992. ISBN 0-201-51585-7. Held in Chicago, Illinois.

[18] A. Kalaiah and A. Varshney. Differential point rendering. In *Rendering Techniques 2001: 12th Eurographics Workshop on Rendering*, pages 139–150. Eurographics, June 2001. ISBN 3-211-83709-4.

[19] N. Kojekine, I. Hagiwara, and Savchenko. Software tools using CSRBFs for processing scattered data. *Computers & Graphics*, 27(2), 2003. To appear.

[20] D. Levin. Mesh independent surface interpolation. In G. Brunett, B. Hamann, and H. M editors, *Geometric Modeling for Scientific Visualization*.

[21] M. Levoy, K. Pulli, B. Curless, S. Rusinkiewicz, D. Koller, L. Pereira, M. Ginzton, S. Anderson, J. Davis, J. Ginsberg, J. Shade, and D. Fulk. The digital michelangelo project: 3d scanning of large statues. *Proceedings of SIGGRAPH 2000*, pages 131–144, July 2000. ISBN 1-58113-208-5.

[22] B. S. Morse, T. S. Yoo, P. Rheingans, D. T. Chen, and K. R. Subramanian. Interpolating implicit surfaces from scattered surface data using compactly supported radial basis functions. In *Shape Modeling International 2001*, pages 89–98, Genova, Italy, May 2001.

[23] P. Neugebauer. Geometrical cloning of 3D objects via simultaneous registration of multiple range images. In *Proceedings of the 1997 International Conference on Shape Modeling and Application (SMI-97)*, pages 130–139, Los Alamitos, CA, Mar. 3–6 1997. IEEE Computer Society.

[24] Y. Ohtake, A. Belyaev, M. Alexa, G. Turk, and H.-P. Seidel. Multi-level partition of unity implicits. *ACM Transactions on Computer Graphics (SIGGRAPH 2003 Proceedings)*, 22(3):463–470, 2003.

[25] Y. Ohtake, A. G. Belyaev, and H.-P. Seidel. A multi-scale approach to 3D scattered data interpolation with compactly supported basis functions. In *Shape Modeling International 2003*, Seoul, Korea, May 2003. Accepted.

[26] H. Pfister, M. Zwicker, J. van Baar, and M. Gross. Surfels: Surface elements as rendering primitives. *Proceedings of SIGGRAPH 2000*, pages 335–342, July 2000. ISBN 1-58113-208-5.

Proceedings of the Shape Modeling International 2004 (SMI'04)
0-7695-2075-8/04 $20.00 © 2004 **IEEE**

[27] S. Rusinkiewicz, O. Hall-Holt, and M. Levoy. Real-time 3d model acquisition. *ACM Transactions on Graphics*, 21(3):438–446, July 2002. ISSN 0730-0301 (Proceedings of ACM SIGGRAPH 2002).

[28] S. Rusinkiewicz and M. Levoy. Qsplat: A multiresolution point rendering system for large meshes. *Proceedings of SIGGRAPH 2000*, pages 343–352, July 2000. ISBN 1-58113-208-5.

[29] V. V. Savchenko, A. A. Pasko, O. G. Okunev, and T. L. Kunii. Function representation of solids reconstructed from scattered surface points and contours. *Computer Graphics Forum*, 14(4):181–188, 1995.

[30] G. Schaufler and H. W. Jensen. Ray tracing point sampled geometry. *Rendering Techniques 2000: 11th Eurographics Workshop on Rendering*, pages 319–328, June 2000. ISBN 3-211-83535-0.

[31] G. Turk and J. O'Brien. Modelling with implicit surfaces that interpolate. *ACM Transactions on Graphics*, 21(4):855–873, October 2002.

[32] M. Zwicker, M. Pauly, O. Knoll, and M. Gross. Pointshop 3d: An interactive system for point-based surface editing. *ACM Transactions on Graphics*, 21(3):322–329, July 2002. ISSN 0730-0301 (Proceedings of ACM SIGGRAPH 2002).

[33] M. Zwicker, H. Pfister, J. van Baar, and M. Gross. Ewa splatting. *IEEE Transactions on Visualization and Computer Graphics*, 8(3):223–238, July - September 2002. ISSN 1077-2626.

A Asymptotic relation of distance to point set and off-center value

We are interested in the behavior of $c(\mathbf{x}) = ||\mathbf{a}(\mathbf{x}) - \mathbf{x}||$ relative to $d_{\mathcal{P}}(\mathbf{x})$, mostly for the case when $\mathbf{x}$ is in the tangent space of the surface represented by the points. We first look at the problem in one dimension and adjust it for the case we are interested in later.

In one dimension, we can sort and translate the points so that $p_0 = 0$, $p_1 = -\delta, \delta > 0$, and $p_i \leq -\delta, i > 1$. Then

$$c(x) = \left| \frac{p_0\theta(|p_0 - x|) + \sum_{i>0} p_i\theta(x + p_i)}{\theta(|p_0 - x|) + \sum_{i>0} \theta(x + p_i)} - x \right| \quad (12)$$

and by exploiting the properties of the p_i this can be rewritten as

$$c(x) = \left| \frac{-\sum_{i>0} |p_i|\theta(x + p_i)}{\theta(|x|) + \sum_{i>0} \theta(x + p_i)} - x \right|. \quad (13)$$

Because we are interested in the behavior of $c(x)$ as x moves away from the points we restrict our analysis to $x > 0$, which allows writing

$$c(x) = \left| -\frac{\sum_{i>0} |p_i|\theta(x + p_i)}{\theta(x) + \sum_{i>0} \theta(x + p_i)} - x \right| \quad (14)$$

and using that θ is defined to be strictly positive in Ω this is equivalent to

$$c(x) = \frac{\sum_{i>0} |p_i|\theta(x + p_i)}{\theta(x) + \sum_{i>0} \theta(x + p_i)} + x. \quad (15)$$

For our particular choice of $\mathcal{P}$ we have $d_{\mathcal{P}}(x) = x$, which immediately shows $c(x) > d_{\mathcal{P}}(x)$. We claim further that the asymptotic behavior for $x \to \infty$ is $c(x) \approx d_{\mathcal{P}}(x)$ (note that θ has compact support, so in practice $c(x)$ won't be supported for $x \to \infty$). Since $\sum_{i>1} |p_i|\theta(x + p_i) < m \sum_{i>1} \theta x + p_i$ for an appropriate $m < \infty$, it is sufficient to show that

$$\lim_{x\to\infty} \theta(x) = \theta(x + \delta), \quad (16)$$

which follows directly from the requirement of monotone decreasing behavior for θ (and also holds for $\delta = 0$).

For the practical case, we assume the points to be scattered on a two-dimensional manifold (with boundary). Because h is determined as the average distance between close points, we assume that on the manifold the (geodesic) distance to any of the sample points is less than $h/2$. We establish an orthonormal coordinate system x, y, z at the boundary of the manifold, so that z points in normal direction and x is orthogonal to the boundary. If the manifold was planar than the distance $d_{\mathcal{P}}$ of points on the x-axis was simply bounded by $\sqrt{x^2 + h^2/4}$. Since we assume the manifold cannot represent features smaller than h, the radius of curvature is larger than h. Thus, the maximum deviation in normal direction between two points at a distance $h/2$ is less than $h - \sqrt{h^2 - h^2/4}$. This leads to

$$d_{\mathcal{P}} < \sqrt{x^2 + h^2/4 + \left(h - \sqrt{h^2 - h^2/4}\right)}$$

for points on the x-axis and we conlcude

$$c(\mathbf{x}) > \sqrt{d^2_{\mathcal{P}}(\mathbf{x}) - h^2(2 - \sqrt{3})}. \quad (17)$$

The asymptotic behavior for large distances remains unchanged, because we assume $d_{\mathcal{P}}(\mathbf{x}) >> h$.

B Bounds on the off-center value in small gaps

We consider sets of points on the real line with spacing roughly h containing the points $-1.5h$ and $+1.5h$ but no points in the interval $]-1.5h, 1.5h[$, representing a gap of $2r_B$ (assuming our preferred choice of $r_B = 1.5h$). We have performed a large number of numerical experiments to evaluate the maximum of $c(x)$. Keeping the points at the boundary of the gap and allowing an arbitrary amount of noise for the other points we find that $c(x) < h$ under all circumstances. Note that we could compute the maximum of $c(x)$ analytically for special settings, however, we feel more comfortable with an exhaustive numerical experiment as the special settings might not cover all real-world cases.

Proceedings of the Shape Modeling International 2004 (SMI'04)
0-7695-2075-8/04 $20.00 © 2004 **IEEE**