Estimating the In/Out Function of a Surface Represented by Points

Vinícius Mello
IMPA - Instituto de Matemática Pura e Aplicada
vinicius@visgraf.impa.br

Luiz Velho
IMPA - Instituto de Matemática Pura e Aplicada
lvelho@impa.br

Gabriel Taubin
IBM T.J. Watson Research Center
taubin@us.ibm.com

ABSTRACT

We present a method to estimate the in/out function of a closed surface represented by an unorganized set of data points. From the in/out function, we compute an approximation of the signed distance function to a surface $\mathcal{M}$ whose sampling are given by this set of points. The procedure correctly detects the interior and the exterior of $\mathcal{M}$, even if $\mathcal{M}$ is multiply connected. The representation of the signed distance function combines the advantages of two previously known schemes, "Implicit Simplicial Models" and "Adaptively Sampled Distance Fields", incorporating new features deriving from the concept of a Binary Multitriangulation.

Categories and Subject Descriptors

I.3.5 [**Computing Methodologies**]: Computer Graphics—*Computational Geometry and Object Modeling*

General Terms

Binary Multitriangulation, Implicit Simplicial Models

Keywords

Stellar Operators, Topology Estimation, Implicit Surfaces

1. INTRODUCTION

The reconstruction of a surface from unorganized sets of data points has became a classical problem in computer graphics. It can be informally posed as follows:

PROBLEM 1. *Given a sufficiently well sampled set of points $S = \{x_i\}$ on a surface $\mathcal{M}$, find a surface $\mathcal{M}'$ that closely approximates $\mathcal{M}$.*

This problem can be greatly simplified if additional information, like the sense of the normal vector in each point, are given. In this paper, we attack problem 1 in its full generality, without further assumptions.

SM'03, June 16–20, 2003, Seattle, Washington, USA.

With no pretension of formulating a complete taxonomy, we can classify the different methods to solve problem 1 into two main lines: the computational geometry approach and the implicit surface approach. The computational geometer uses all his machinery of Voronoi diagrams, Delaunay triangulations, alpha shapes and so on, in order to guarantee sufficient conditions for reconstruction and optimal computational complexity bounds. In spite of their theoretical relevance, the related algorithms tend to be of complex implementation and of low effectivity in real problems, since we cannot always verify that sufficient conditions exist. In a sense, this approach is too combinatorial.

The implicit surface approach seeks to represent $\mathcal{M}'$ as the zero set of a function f such that

$$f(x_i) = 0.$$

This formulation leads naturally to the analytical techniques of interpolation theory and, among many other methods, the use of *Radial Basis Functions* (RBF's) as interpolant elements seems to be the most successful [4]. However, because of the unbounded and continuous nature of the RBF's, sophisticated numerical methods are required to compute and to evaluate f (see [3]). That is, we could say that this approach is too continuous.

In [15], Taubin and Ronfard proposed a half way, introducing the concept of *Implicit Simplicial Models*, where the surface $\mathcal{M}'$ still is represented as the zero set of a continuous function f, but now f is defined over a simplicial complex K. Thus, the simpler combinatorial nature of the domain of f can be exploited to recover, for example, the topology of $\mathcal{M}'$. Nevertheless, they implemented only an algorithm to reconstruct planar curves.

In this paper, we extend that algorithm to surface reconstruction and improve it putting more structure on the simplicial complex K, by applying the concept of *Binary Multitriangulations* [5]. In this sense, our approach resembles that taken by Frisken et al in [8], where the concept of *Adaptively Sampled Distance Fields* was proposed, but there are essential differences between then, as we will point out in section 3.2.

Shortly, our algorithm takes as input a set of points sampled over a closed surface [1], without normals, and outputs a simplicial decomposition of a bounding box enclosing the points, and, for each vertex of that decomposition, a sign +/- telling if the vertex is outside or inside of the surface. From that output, its easy to compute an approximation

[1]For a definition of closed surface, see [6].

of the signed distance function of the point cloud. Finally, a marching tetrahedra algorithm can be used to extract a polygonal mesh that approximates the surface.

The paper is organized as follows. In Section 2, we discuss some of the previous work to solve problem 1, emphasizing the techniques that are closely related to ours. In Section 3, we present the basic concepts that are necessary to the comprehension of the structuring and reconstruction algorithm introduced in Section 4. Finally, in Section 5, we summarize the results of our method and suggest some possible extensions to it.

2. RELATED WORK

One of the most cited paper that is related to problem 1 is the work of Hoppe [9]. In this paper, local approximations of the tangent plane of a point cloud are constructed using principal component analysis (PCA). Then, an optimization algorithm is used to analyze the *Riemannian Graph*, (i.e. a graph that encodes geometric proximity of the tangent plane centers), and determines a consistent orientation of the tangent planes. From a family of oriented tangent planes, a piecewise linear approximation of the signed distance function is obtained. The contact point between this work and ours is the use of PCA to estimate the tangent plane to the underlying surface defined by a point cloud. On the other hand, our approach is hierarchical and adaptive: we build a sequence of partial solutions progressively more refined and fitted to the data, and the final mesh is obtained using a marching tetrahedra algorithm over a adaptive simplicial mesh, rather than a marching cubes over a fixed resolution cubical mesh.

In [1], the focus is the generation of implicit representations of a $\mathcal{M}'$, that are smooth and not just given by a piecewise linear approximation. In order to achieve this goal, the authors employ computational geometry techniques to build the, so called, simplicial hull (a tetrahedral mesh) around the surface $\mathcal{M}$, and then a C^1 interpolatory surface is constructed using cubic A-patches. That is, $\mathcal{M}'$ is represented as a piecewise cubic implicit surface. One problem of this approach is that the sufficient conditions to guarantee that each A-patch is a single sheeted surface are very intricate [2]. We overcome this difficulty by requiring a diffeomorphism condition on each tetrahedron (see definition 1), avoiding *a priori* self-intersections in the implicit surface.

As we have mentioned before, RBF's are very powerful approach to solve problem 1. However, an important point, frequently forgotten (and one of the key features of fast multipole methods, the tool needed to efficiently evaluate the RBF's), is the use of a hierarchical subdivision of space to cluster the sample points [3]. In this paper, the process of hierarchical subdivision is used to estimate the topology and adaptively approximate the geometry, simultaneously.

It is exactly in relation to the use of hierarchical subdivision techniques, that resides the main difference between our paper and the work of Taubin and Ronfard [15] (besides, of course, the extension from curve to surface reconstruction). In their work, a triangle could be subdivided in two different ways and the space refinement process didn't follow any pre-established order. We, on the other hand, through the concept of a Binary Multitriangulation, enforce the police that each tetrahedron is always divided into two, along its longest edge, and conversely, an edge can only be subdivided if it is the longest edge of all tetrahedron incident on it. Because the triangulation obtained by the above process possesses excellent adaptation qualities, we were able to eliminate the relaxation step of the algorithm described in [15], which simplifies considerably our implementation.

3. BACKGROUND

In this section we review some concepts that are necessary for the understanding of the structuring and reconstruction algorithm described in Section 4.

3.1 Implicit Simplicial Models

Implicit Simplicial Models (ISM's) were first introduced in [15]. Here we present a generalized definition. The precise definitions of topological concepts like simplex, simplicial complex and combinatorial manifold can be found in [5]. The following definitions are needed to grasp definition 1. Let $\Pi^n = \{(w_0, \ldots, w_n) \in R^{n+1} | w_0 + \cdots + w_n = 1\}$. The *standard n-simplex* Δ^n is defined by $\Delta^n = \{(w_0, \ldots, w_n) \in \Pi^n | w_i \geq 0\}$. By $v_i\Delta^n$ we denote the *i-th vertex* of Δ^n, that is, if $v_i\Delta^n = (w_0, \ldots, w_n)$, then $w_i = 1$. For each n-simplex $\sigma =< p_0, \ldots, p_n >$ in the Euclidean space R^n, we can associate a map $W_\sigma : \Pi^n \to R^n$ given by

$$W_\sigma(w_0, \ldots, w_n) = w_0 p_0 + \ldots + w_n p_n,$$

called *barycentric parametrization*. This map has an inverse W_σ^{-1} known as *barycentric coordinates* with respect to σ. Note that W_σ takes Δ^n in σ and that $W_\sigma(v_i\Delta^n) = p_i$. We can label the faces of Δ^n with a function $\chi : \Delta^n \to \{0,1\}^{(n+1)}$ defined by

$$\chi(w_0, \ldots, w_n) = (b_0, \ldots, b_n),$$

where

$$b_i = \begin{cases} 0, & \text{se } w_i = 0 \\ 1, & \text{se } w_i \neq 0\,. \end{cases}$$

Finally, we say that a map $X : \Delta^n \to \Delta^n$ is face preserving if $\chi \circ X = \chi$. We can now enunciate the following definition:

DEFINITION 1. *An* Implicit Simplicial Model O *consists of*

1. *A combinatorial n-manifold* $M \subset R^n$ *triangulated by the simplicial complex* K*;*
2. *for each n-simplex* $\sigma \in K$*, a face preserver diffeomorphisms* $X_\sigma : \Delta^n \to \Delta^n$*, satisfying*

$$\sigma_1 \cap \sigma_2 = \sigma \Rightarrow W_{\sigma_1} X_{\sigma_1} W_{\sigma_1}^{-1}|_\sigma = W_{\sigma_2} X_{\sigma_2} W_{\sigma_2}^{-1}|_\sigma$$

3. *a function* f *from the vertices of* K *to the set* $R - \{0\}$*.*

Item 2 is essentially a reparametrization of each n-simplex compatible with the neighborhood relations. From an ISM, we can define a function

$$F(p) = \sum_{i=0}^{n} f(W_\sigma(v_i\Delta^n)) x_\sigma^i(W_\sigma^{-1}(p)), \qquad (1)$$

where σ is a n-simplex that contains p and $X_\sigma = (x_\sigma^0, \ldots, x_\sigma^n)$. The equation $F(p) = 0$ defines a hypersurface which topology can be recovered from the topology of K and from the signal of function f, and which its geometry depends on the reparametrizations X_σ.

At first sight,it may seem complicated to find diffeomorphisms satisfying item 2, specially if we consider that such

diffeomorphisms must depend on a few parameters. A very simple and natural choice is set $X_\sigma = I$, where I is the identity function. In fact, this is the choice adopted in this paper. Because our main interest is to estimate the correct topology of the surface, the piecewise linear reconstruction which we obtain is satisfactory.

However, we could not find any references in the literature to this kind of "generalized barycentric coordinates". Because we realize the importance of this tool for our future work, we are investigating a manageable way to generate such diffeomorphisms and we expect to publish the results of such research in a forthcoming paper.

3.2 Adaptively Sampled Distance Fields

According to its authors, "ADFs consists of adaptively sampled distance values organized in a spatial data structure together with a method for reconstructing the underlying distance field from the sampled values." [8]. In spite of this very general definition, in practice the spatial data structures used for ADF's are quadtrees (octrees), together with bilinear (trilinear) interpolation for reconstruction.

On the other hand, it is well known that such combinations of spatial decomposition and interpolation scheme suffer from discontinuity problems (i.e., the infamous "T-junctions").

It is also well known that this problem can be minimized avoiding that neighbor cells of the quadtree (octree) differ by more than one level. This has the additional advantage of providing a more gradual variation of the interpolated data.

In this paper, we use a simplicial structure for spatial decomposition. Therefore, we do not have T-junction problems, and also we can exploit the above-mentioned graduallity property. The concepts related to this structure is described next.

3.3 Binary Multitriangulations

Here we summarize the main results introduced in [5]. We will show that the interplay between combinatorial and geometrical structures of the subdivision process leads naturally to the concept of binary multitriangulations, or BMT's.

Stellar subdivision is the basic operation used to construct a BMT. A review of Stellar theory and its fundamental concepts can be found in [12]. For our purposes here, it is sufficient to state a few facts. We will denote by $M' = (\varepsilon, \nu)M$ the operation of bisecting the edge ε of a simplicial mesh M, inserting the vertex ν to obtain the new mesh M' (note that this is a particular case of a stellar operation). A remarkable fact, proved by Newman (references and a modern proof can be found in [12]), is that if M and M' are two meshes which represent the same polyhedron, then we can transform M into M' through a sequence of stellar operations, such as the one described above.

Subdivision operations on edges have one important additional advantage. Whenever a stellar subdivision happens on an edge ε, all simplices containing ε are splitted in two. Accordingly, a sequence of stellar subdivision induces a binary tree structure in the simplices. And binary trees often leads to simpler algorithms.

In order to define the binary multitriangulation concept (BMT), we need some auxiliary definitions. We follow closely the definitions in [7].

A *partially ordered set* (poset) $(C, <)$ is a set C with an antisymmetric and transitive relation $<$ defined on its elements. Given $c, c' \in C$, notation $c \prec c'$ means $c < c'$ and there in no $c'' \in C$ such that $c < c'' < c'$. An element $c \in C$, such that for all $c' \in C$, $c \leq c'$, is called a *minimal* element in C. If there is a unique minimal element $c \in C$, then c is called the *minimum* of C. Analogously are defined *maximal* and *maximum* elements.

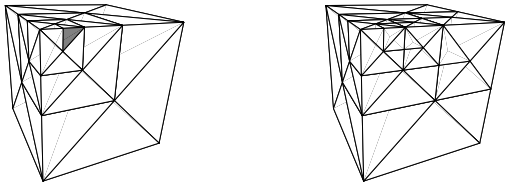
DEFINITION 2. *A* binary multitriangulation *is a poset* $(\mathcal{T}, <)$*, where* $\mathcal{T}$ *is a finite set of abstract* 3*-manifolds (named* triangulations*) and the order* $<$ *satisfies:*

1. $M' \prec M$ *if, and only if,* $M' = (\varepsilon, \nu)M$*, for some edge* $\varepsilon \in M$*.*
2. *There is maximum and minimum abstract* 3*-manifolds in* $\mathcal{T}$*, called* base triangulation *and* full triangulation*, respectively*

Property 2 says, in fact, that a BMT is a *lattice*. Other fact which follows from the definition is that every two triangulations in $\mathcal{T}$ are stellar equivalent. As usual, a BMT can be thought as a directed acyclic graph (DAG), with one drain and one source, whose arrows are labeled with stellar subdivisions on edges. From an algorithimic perspective, the key idea is to use the above mentioned binary tree structure in the simplices to encode the DAG.

We have implemented these ideas in a C++ library using generic programming techniques. This give us great flexibility in the specialization of the general concepts already defined, without a performance overhead. One of the most useful of such specializations is the concept of a *rectangular triangulation*. In this case, the base triangulation is taken to be a cube subdivided into six tetrahedron. Also, the relation $\prec$ possesses the additional restriction that the edge ε has to be the longest [2] edge of all tetrahedra which contain ε.

One of the functions implemented in the library, allow a subdivision of an arbitrary tetrahedron of the rectangular triangulation, guaranteeing that the above mentioned restriction is satisfied, that is, automatically performing extra subdivisions in neighbor tetrahedra which contain longer edges (see Fig. 1). This operation is essential in the adaptation algorithm described below.



(a) (b)

Figure 1: The splitting of the marked tetrahedron automatically causes a sequence of subdivisions in the BMT.

[2]It is possible to give a completely combinatorial interpretation to the term "longest" employed above, but we will not go into the details here.

4. THE ALGORITHM

In this section we describe an algorithm to solve problem 1. The only assumption is that the data points belong, or are close to, a closed surface, and are roughly uniformly distributed along the surface. The topology is unknown in advance and estimated by the algorithm. The algorithm output is a ISM that represents an approximation of the signed distance function, from which we can to easily extract a triangular mesh. Program 1 shows the global structure of the algorithm.

Program 1 Main loop of the algorithm

```
while(!to_refine.empty()) {
  simplex s=to_refine.pop();
  local_fitting(s);
  if(!stop_test(s)) subdivide(s);
  estimate_topology();
}
estimate_distance();
```

Initially, we find a cube [3] which enclose the points. There are many ways to choose that cube, but we had adopted a cube which center is the centroid of the points and which orientation is obtained via PCA analysis. We set the base mesh to the cube subdivided in six tetrahedra. These six tetrahedra are inserted into the list `to_refine` and the algorithm enters in a loop for refinement and topology estimation. In this loop, we remove a tetrahedron σ of the `to_refine` list , compute the local fitting and apply the stop test. This test includes the verification of maximum subdivision level, minimum number of samples in σ and a data adjustment in a coplanarity sense, which we will explain in detail later. In case σ does not pass the test, it is subdivided and the new tetrahedra generated by the subdivision are inserted into `to_refine`. After each subdivision, one passe of the topology estimation algorithm is performed. The loop continues while `to_refine` is not empty. Finally, the algorithm ends with the distance estimation step.

Now we describe in detail the main building blocks of the surface reconstruction algorithm: local fitting, topology estimation and distance estimation.

4.1 Local Fitting

The local fitting step consists in the standard application of principal component analysis. We calculate the barycenter p_σ and the covariance matrix C_σ of the N_σ points contained in σ. The smallest eigenvalue λ^0_σ of C_σ measures the least square error of the sample points. We keep stored in each tetrahedron the "coplanarity error" $\epsilon_\sigma = \frac{\lambda^0_\sigma}{\lambda^0_\sigma+\lambda^1_\sigma+\lambda^2_\sigma}$, where λ^i_σ are the eigenvalues of C_σ, together with the plane Π_σ which passes trough p_σ and is perpendicular to the eigenvector associated to λ^0_σ. This information will be necessary during the topology estimation step. The idea is that Π_σ represents a plane which is secant to the portion of the surface $\mathcal{M}$ contained in σ which works as a local approximation of $\mathcal{M}$, with approximation accuracy given by ϵ_σ.

[3]In fact, to achieve a better adjustment to the data points, a scaling can be applied to the cube, obtaining in this way, a parallelepiped. However, to simplify the exposition of the algorithm, we will refer to this bounding cell as "the cube".

It may happen during the subdivision that some tetrahedra possess an insufficient number of samples, causing the matrix C_σ to be singular. In that case, that is, when the number of samples per tetrahedron is less than a user defined parameter, usually between 5 and 10, we set ϵ_σ and Π_σ with the same values of the parent tetrahedron.

4.2 Topology Estimation

The goal of the topology estimation step is to determine a sign $S_\nu \in \{+1, -1\}$ for each vertex ν of the triangulation, thus indicating if ν is inside or outside the volume enclosed by the surface $\mathcal{M}$. Note that we can prescribe a sign for each vertex ν of a tetrahedron σ based on the plane Π_σ,

$$\begin{cases} S^\sigma_\nu = +1, & \text{if } \nu \in \Pi^+_\sigma \\ S^\sigma_\nu = -1, & \text{if } \nu \in \Pi^-_\sigma. \end{cases}$$

However, nothing guarantees that these signs are globally consistent in relation to the other tetrahedra which also contain ν. It is like each tetrahedron could have its own opinion about which sign should be given for its vertices. Of course, it is expected that the opinion of a tetrahedron that possesses a small mean square error should be more trustworthy than the opinion of a tetrahedron with a large error. It is also clear thst, if two trustworthy tetrahedra which share the same edge ε both agree that there is a sign change, i.e., that the surface $\mathcal{M}$ intersects ε, then, this is a strong indication that the vertices of ε should have opposite signs. Taking these facts into account, we conclude that the minimization of the function $D(S)$ below, is equivalent to minimize the disagreement of the tetrahedra with respect to the sign prescribed to its vertices:

$$D(S) = \sum_{\varepsilon \in E} H_\varepsilon \; S_{\nu_0} S_{\nu_1} \tag{2}$$

with the sum ranging over all the edges $\varepsilon = \{\nu_0, \nu_1\}$ and where the edge weight H_ε is given by

$$H_\varepsilon = - \sum_{\sigma : \varepsilon = \{\nu_0, \nu_1\} \in \sigma} S^\sigma_{\nu_0} S^\sigma_{\nu_1} \left(\frac{\epsilon_{\max} - \epsilon_\sigma}{\epsilon_{\max} - \epsilon_{\min}} \right)$$

with the sum extended over all tetrahedra that contains ε as an edge. Above, $\epsilon_{\max}$ means the maximum ϵ_σ and $\epsilon_{\min}$ is similarly defined. Of course, we also have to care of empty tetrahedra. We set $S^\sigma_\nu = 1$ and $\epsilon_\sigma = \epsilon_{\text{empty}}$ whenever σ does not contain points inside. The ϵ_{empty} parameter relates to the global uncertainty of the data. Setting $\epsilon_{\text{empty}} = 0$ is equivalent to knowing the point locations with absolute accuracy.

The minimization of the quadratic expression (2) in $\{+1, -1\}$ is exactly the Ising model, for which simulated annealing schemes are well documented [11]. Our implementation of the simulated annealing algorithm took advantage of the fact that our data structure allows very efficient queries about the neighborhood of each vertex.

4.3 Distance Estimation

The distance estimation step is currently very simple. Initially, we collect the vertices of all tetrahedra which contain data points in a set V. Next, for each ν in V we compute the smallest distance $d_{\min}$ between ν and the data points contained in each tetrahedra incident in ν. Since our BMT spatial data structure provides all these incidence relation-

ships, the process is quite fast. Then, we set the distance value f in ν as $S_\nu d^2_{\min}$.

For our immediate objective, that is, the extraction of a polygonal mesh that approximates the surface, it is sufficient to know the value of f in the set V. Depending on the application, it may be necessary to extend f to all other vertices of the BMT. In this case, we could use a propagation method, such as the fast marching method [10, 13].

4.4 Details

In order to facilitate the exposition, we have omitted some details that we discuss now. Since it is somewhat expensive to run the topology estimation step each time a tetrahedron is subdivided, we decided to execute this step only after a (user-defined) number of subdivisions. This increases significantly the performance of the program, on the other hand it degrades slightly the quality of the topology estimation. The solution to this problem is to run the topology estimation one more time when the loop terminates, with the simulated annealing parameters adjusted for a more gradual decrease in temperature. In summary, the coarse topology, found in the refinement loop is used as the initial condition for a final iteration of topology estimation.

The `to_refine` list is, in fact, a priority queue ordered by the ϵ_σ value. Thus the tetrahedra with greater error are divided first, ensuring the correctness of the hierarchical optimization process.

When the point cloud is not sufficiently dense, or when the user chooses a small number of subdivision levels, it may happen that the distance computed by the algorithm vary excessively between neighbor vertices – causing the extracted polygonal mesh to be "noisy". In this case, we can apply a smoothing algorithm such as the one described in [14].

4.5 Results

In this section we show some results of using the algorithm in synthetic and scanned data sets.

The first example is the Stanford bunny model. Figure 2(a) shows the set of cells V which are classified as containing the surface. Figure 2(b) exhibits the polygonization obtained from the zero level set of the distance function in V.

The second example is a synthetic sampling of knotted shape. Figure 3 shows the cells which contain the surface. Note that, despite of the complicated embedding of this surface, the topology estimation algorithm was able to recover the correct shape.

The third example is a scanned dragon model, from the Stanford repository. The original data set contained more than 400 K points. Figure 4 shows a phong shaded image of the extracted polygonal mesh.

The computational performance of our algorithm can be analyzed through the timings shown in table 1. The execution was performed on a 1.5GHz Pentium 4 with 512 MB RAM. The data indicates that the topology estimation is the most expensive operation in the process. We believe that we can increase the performance by fine tuning the simulated annealing parameters.

	samples	ref.	dist.	top.	mesh	smooth.
bunny	35947	25.7	0.8	48.31	0.13	2.37
knot	100000	37.26	1.17	49.6	0.2	3.34
dragon	437645	9:18.34	5.77	1:13.92	1.13	16.40

Table 1: Timing results in seconds. The meaning of the columns: ref is the time spent in the main loop; dist, the time in the distance estimation step; top, the topological refinement described in section 4.4; mesh, the mesh extraction time; smooth, the time spent in the mesh smoothing.

5. CONCLUSION

The starting point for our approach to solve problem 1 was the minimization of function (2). For this, we employ a stochastic optimization algorithm, the simulated annealing. A topic of further research is the utilization of another combinatorial optimization algorithm. Anyway, the key aspect seems to be the use of the kind of "multi-grid" approach we have taken here.

Our method recovers the topology just knowing the points spatial position. Another research topic is the incorporation of normal vector information to our scheme. This will provide better topology and geometry results. The geometry can also be improved by using diffeomorphisms X_σ of higher degree, not merely piecewise linear.

A careful analysis of the topology estimation step revels that the point location data is used just to compute the centroid and covariance matrix for each tetrahedron. The algorithm works equally well if a black box supplies such information. Therefore, we conclude that our algorithm can be generalized to reconstruct surfaces described by probability distributions. We pretend to explore that point in future work.

Summarizing, the main feature of Implicit Simplicial Models is the intrinsic separation between topology and geometry, combinatorial and continuous data. In this paper, we have focused on topology recovering. In future work, we intend to exploit the hierarchical organization and space adaptivity of Binary Multitriangulations, as well the ISM local geometry fitting power, in order to achieve better geometric reconstructions.

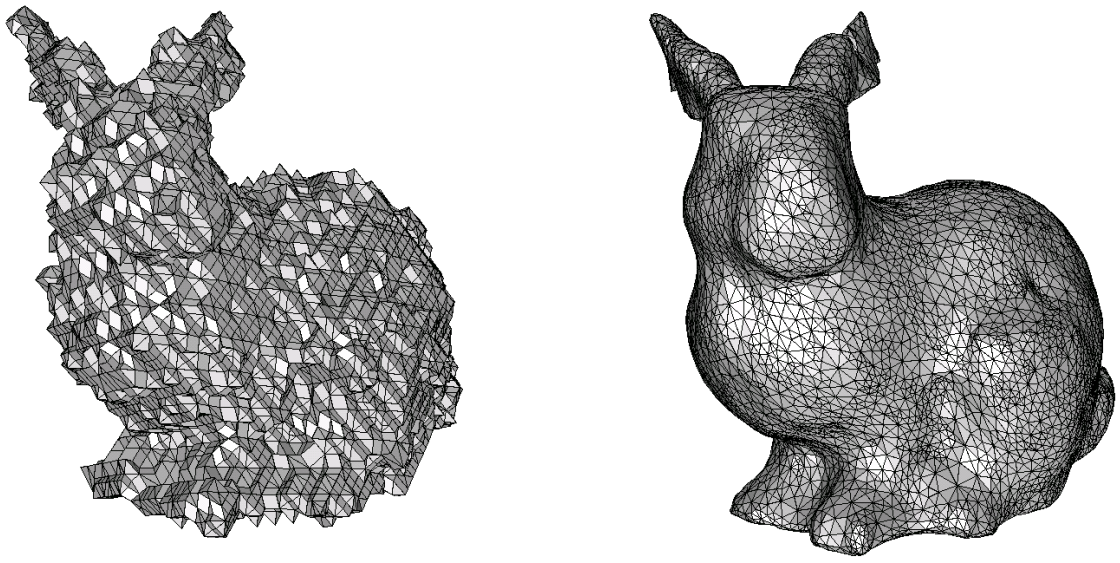
6. REFERENCES

[1] Chandrajit L. Bajaj, Fausto Bernardini, and Guoliang Xu. Automatic reconstruction of surfaces and scalar fields from 3d scans. In *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, pages 109–118. ACM Press, 1995.

[2] Chandrajit L. Bajaj, Jindon Chen, and Guoliang Xu. Modeling with cubic A-patches. *ACM Transactions on Graphics (TOG)*, 14(2):103–133, 1995.

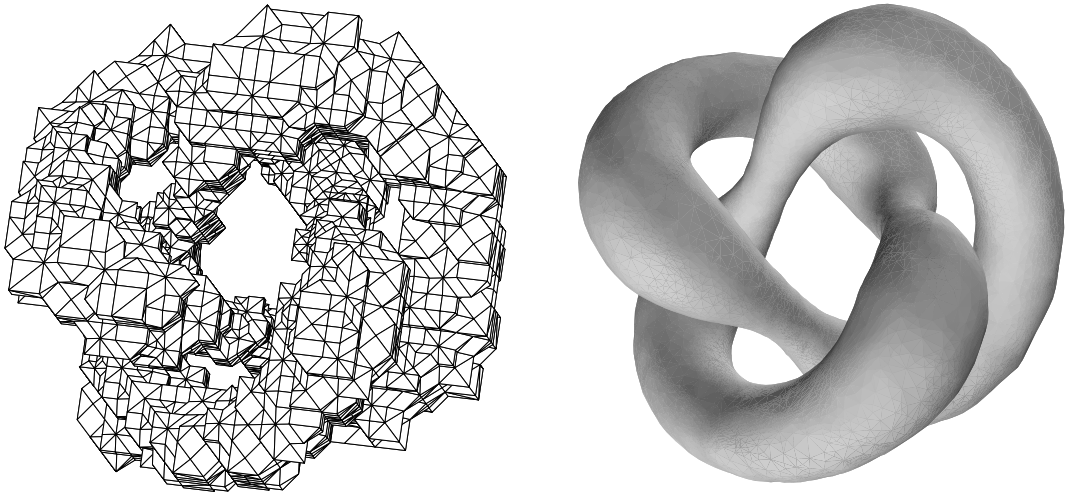
[3] R. K. Beatson and L. Greengard. *A short course on fast multipole methods*, pages 1–37. 1997.

[4] 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. *Proc. Siggraph 2001*, pages 67–76, 2001.



(a) (b)

Figure 2: Bunny.



(a) (b)

Figure 3: knot.

Figure 4: Dragon.

[5] Vinícius Mello, Luiz Velho, Paulo Roma Cavalcanti, and Cláudio Silva. A generic programming approach to multiresolution spatial decompositions. *Proc. VisMath'02*, 2001.

[6] M. P. do Carmo. *Differential Geometry of Curves and Surfaces*. Prentice-Hall, Englewood Cliffs, NJ, 1976.

[7] L. De Floriani, E. Puppo, and P. Magillo. A formal approach to multiresolution modeling. In W. Straßer, R. Klein, and R. Rau, editors, *Theory and Practice of Geometric Modeling*. SpringerVerlag, 1996.

[8] Sarah F. Frisken, Ronald N. Perry, Alyn P. Rockwood, and Thouis R. Jones. Adaptively sampled distance fields: A general representation of shape for computer graphics. *Proc. Siggraph 2000*, pages 249–254, 2000.

[9] Hugues Hoppe, Tony DeRose, Tom Duchamp, John McDonald, and Werner Stuetzle. Surface reconstruction from unorganized points. *Proc. Siggraph 92*, 26(2):71–78, 1992.

[10] R. Kimmel and J. Sethian. Computing geodesic paths on manifolds. *Proceedings of National Academy of Sciences, USA*, 95(15):8431–8435, 1998.

[11] P.V. Laarhoven and E. Aarts. *Simulated annealing: Theory and applications*. Reidel Pub., Dordrecht, Holland, 1987.

[12] W. B. R. Lickorish. Simplicial moves on complexes and manifolds. In *Proceedings of the Kirbyfest*, volume 2, pages 299–320, 1999.

[13] J. Strain. Fast tree-based redistancing for level set computations. *Comput. Phys.*, 152:648–666, 1999.

[14] G. Taubin. Linear anisotropic mesh filters. Technical Report RC-22213, IBM Research, 2001.

[15] Gabriel Taubin and Remi Ronfard. Implicit simplicial models for adaptive curve reconstruction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(3):321–325, 1996.