Surface Reconstruction of Noisy and Defective Data Sets

Hui Xie Kevin T. McDonnell Hong Qin [*]
Computer Science Department, State University of New York at Stony Brook, NY 11794-4400

Abstract

We present a novel surface reconstruction algorithm that can recover high-quality surfaces from noisy and defective data sets without any normal or orientation information. A set of new techniques are introduced to afford extra noise tolerability, robust orientation alignment, reliable outlier removal, and satisfactory feature recovery. In our algorithm, sample points are first organized by an octree. The points are then clustered into a set of monolithically singly-oriented groups. The inside/outside orientation of each group is determined through a robust voting algorithm. We locally fit an implicit quadric surface in each octree cell. The locally fitted implicit surfaces are then blended to produce a signed distance field using the modified Shepard's method. We develop sophisticated iterative fitting algorithms to afford improved noise tolerance both in topology recognition and geometry accuracy. Furthermore, this iterative fitting algorithm, coupled with a local model selection scheme, provides a reliable sharp feature recovery mechanism even in the presence of bad input.

CR Categories: I.3.5 [Computer Graphics]: Computational Geometry—Geometric Algorithms and Object Representations

Keywords: Computer Graphics, Surface Reconstruction, Surface Representation, MPU implicits, Modified Shepard's Method.

1 Introduction

Surface reconstruction of a scanned real-world object is a critical step of the *reverse engineering* pipeline. The surface of an object is first digitized into a set of sample points. Once these points have been collected, a surface reconstruction technique is invoked to build a surface representation that is faithful to the collected points. Many surface digitizing devices, such as laser range image scanners, can produce rather dense and accurate data samples. Some optical devices can also provide inside/outside information or even accurate surface normals. Most conventional surface reconstruction techniques are highly (strongly) device-oriented, in the sense that they rely on specific information such as surface normals or viewpoint (e.g. volume carving), or that they pose excessive demand on data accuracy or density. General devices, according to the digitizing techniques used, often produce defective data samples that are subject to noise, local absence of data (e.g., due to occlusion), outliers, etc. A set of desirable properties for a general-purpose surface reconstruction technique includes:

- *Time and space efficiency*,
- *Robustness for defective data sets*,
- *No restriction on topological type*,
- *Ability to fill holes and adapt to varying sampling density*,
- *(Abrupt) noise and outlier removal*,
- *Sharp feature preservation*.

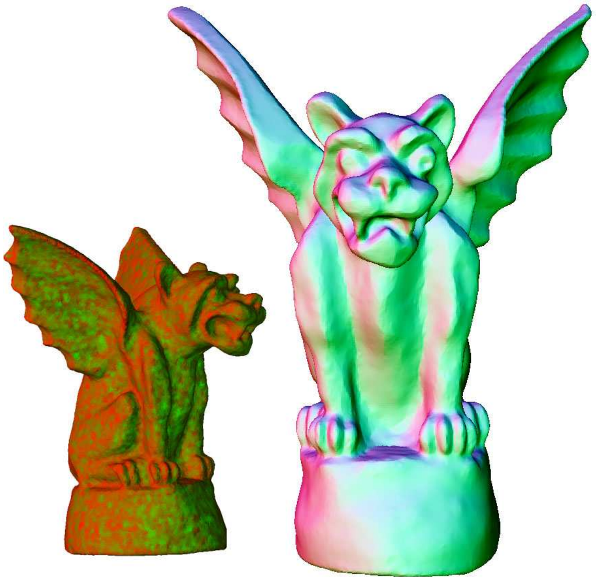


Figure 1: Our noise-tolerant surface reconstruction system. The noise rate is 0.7% of the object size. The side view of the gargoyle shows the fitting error map. Red color indicates larger fitting error.

In this paper, we propose a novel surface reconstruction technique which poses very little restriction on the data set. Our system takes as input a set of (noisy) sample points without any normal or orientation information and outputs a piecewise smooth surface. Sharp edges and corners can be well identified and recovered. The main ideas behind our technique are that (1) we never use local unreliable information to determine global properties, and (2) we iteratively detect false local results and recompute the local fitting when such bad input is detected.

We employ the recently introduced Multi-level Partition of Unity implicits (MPU implicits) by Ohtake et al. [2003a] as the underlying surface representation. In general, implicit surfaces can provide better topological flexibility and automatic hole filling than can piecewise parametric models. MPU implicits can also offer the desirable properties of local support and local feature representation. Xie et al. [2003] also developed a similar implicit surface representation scheme based on the modified Shepard's method for noisy data fitting. Ohtake et al's surface reconstruction algorithm builds an MPU implicit surface from an unorganized point cloud with prescribed normals. Their system exhibits extraordinary capability in sharp feature preservation, hole filling, and topological adaptability. However, their method requires accurate data sets with precise normals associated with each point. We note that the normal information plays a key role in determining the topology of the surface. Our goal is to augment the MPU implicit reconstruction system to general data sets while preserving all its desirable properties such

[*] email:{xhui|ktm|qin}@cs.sunysb.edu

as sharp feature recovery.

We term such noisy and incomplete data sets with possible outliers and abrupt bursts of high noise as “noisy” and “defective” data sets. In our system, the sample points are arranged over an octree. We first cluster the points into monolithically oriented groups through an active contour method. These groups will be used to determine the orientation of each locally fitted implicit surface. Instead of using Hoppe et al.’s [1992] orientation propagation method, the inside/outside test is conducted through a voting algorithm. This algorithm utilizes not neighboring cells but global information to decide the orientation of each implicit surface patch; thus, it provides improved robustness. We also propose an iterative surface fitting algorithm to correct erroneous fitting results that exhibit high error or undecidable orientations. Later fitting iterations use the normal vectors produced by previous iterations for feature detection and model selection. In a nutshell, our system exploits a series of mechanisms to deal with noisy and corrupted data sets with extra robustness and efficiency.

2 Related Work

Our work seamlessly integrates ideas from several research areas in the literature of computer modelling and surface reconstruction. In this section, we discuss related work in areas of implicit surfaces (signed distance fields), deformable models (active contour methods), and spatial partition-based surface reconstruction algorithms.

Compared to explicit models, implicit models can very efficaciously deal with objects of arbitrary topology, blend surface primitives, tolerate noise, and fill holes automatically. The traditional approaches are based mostly on Blinn’s idea of blending globally or locally fitted implicit primitives [1982]. Following this line, Muraki [1991] uses a linear combination of Gaussian blobs as the surface fitting model. Carr et al. [2001], Turk and O’Brien [1999] use globally supported radial basis functions(RBFs) to fit data points by solving a large dense linear system. Morse et al. [2001] and Ohtake et al. [2003b] reduce the computational cost to a sparse linear system by using locally supported RBFs. Another approach in this direction defines the distance field as the distance to a locally fitted surface. Hoppe, DeRose, et al. [1994] define the signed distance field as the distance to a locally fitted tangent plane. Curless and Levoy [1996] build the signed distance function on a volumetric grid using a set of depth images. Bajaj, Bernardini et al. [1995] combine piecewise algebraic surfaces with α-shapes. Although algebraic patches are more flexible for local shape fitting, they need a cumbersome procedure to maintain the single-sheeted nature and the continuity between adjacent implicit patches. Recently, Ohtake et al. [2003a] proposed Multi-level Partition of Unity implicits (MPU implicits). This method can be seen as the combination of algebraic patches and radial basis functions. Xie et al. [2003] also proposed a similar method to build the distance field using the modified Shepard’s blending of implicit patches. One common requirement for the local implicit fitting methods is to determine the orientation of each patch. A frequently used technique proposed by Hoppe et al.[1994] is by orientation propagation along a minimum spanning tree. This method can suffer in the presence of sparse and incomplete data sets, noise, and sharp corners [Xie et al. 2003]. Recently, Mello et al. [2003] proposed an orientation alignment algorithm by minimizing a cost function. However, this method may be costly due to the globally defined cost function and the simulated annealing solver. Many algorithms determine the orientation by using some additional information such as normals or view direction in data sets acquired from some particular devices.

Another category of surface reconstruction algorithms widely exploited in computer vision are deformable models or active contour methods. Kass, Witkin and Terzopoulos[1987] posed the image segmentation problem as an energy minimization of a deforming surface. DeCarlo and Metaxas[1995], McInerney and Terzopoulos[1995] proposed a technique that allows changes in topology. Recent extensions of the deformable models to implicit surfaces (also known as the level-set method) have been developed that employ simpler data structures and support a wider range of topologies [Sethian 1999; Osher and Fedkiw 2002; Whitaker 1998; Zhao et al. 2001]. Deformable models are very robust in dealing with noisy data sets. Their main drawback is their low performance due to their reliance on partial differential equations.

The third group of approaches are through Delaunay triangulation[Boissonnat 1984; Edelsbrunner and Mcke 1994; Amenta et al. 2001]. The surface is a subset of the faces of the Delaunay triangulation upon the data points. Nina Amenta, et al.[2001] present the Power Crust algorithm, which employs the medial axis transformation.

In essence, explicit deformable models and level-set methods can be formulated as a volume sweeping method. The volumetric regions swept by the active contour are distinguished from the rest of the volume. Delaunay triangulation-based algorithms follow a space partition and selection manner. In this paper, we design a pseudo-medial axis transformation algorithm using a simple deformable model. This novel algorithm features improved robustness and efficiency for inside/outside determination.

3 The Modified Shepard’s Blending Method

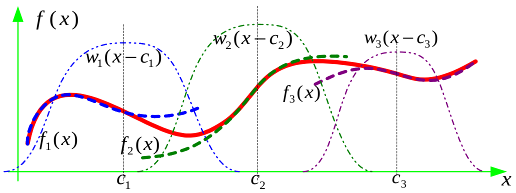


Figure 2: The modified Shepard’s blending of a set of locally defined functions. The resulting function (red solid curve) is constructed from three local functions (thick dashed curves) with their associated weight functions (dashed double-dotted curves).

The modified Shepard’s blending method was originally designed for building a global function from a set of locally defined functions. The resulting global function resembles the behavior of each locally defined function within their respective domains of influence. In Fig. 2, three real functions $f_1(x)$, $f_2(x)$ and $f_3(x)$ are defined locally in the vicinities of c_1, c_2 and c_3 respectively. The domain of influence of each local function is designated by assigning to it a nonnegative weight function $w_i(x)$. Accordingly, the local functions $f_i(x)$ can be considered basis functions. The result blended function by the modified Shepard’s method can be formulated as:

$$f(x) = \frac{\sum_{i=1}^{n} f_i(x) w_i(x - c_i)}{\sum_{i=1}^{n} w_i(x - c_i)}, \tag{1}$$

where $n = 3$ for this specific example. Usually, a weight function takes the form of a nonnegative real-valued radial basis function $w(x) = \phi(r), r = \|x - c_i\|$. Typical choices of the radial basis functions include the inverse distance functions $\phi(r) = (\frac{Max(R-r,0)}{R\,r})^2$ [Franke and Nielsen 1980], the Gaussian distribution function $\phi(r) = exp(-c\ r^2)$ and the inverse multiquadric $\phi(r) = (r^2 + c^2)^{-1/2}$.

To achieve computational efficiency, the weight functions $\phi_i(r)$ are often compactly defined, that is, there exist a set of R_i, $r > R_i \Rightarrow \phi_i(r) = 0$. Interpolation at $f_i(c_i)$ can be achieved by letting $\lim_{r \to 0} \phi_i(r) \to +\infty$. We use a B-spline-based form as Ohtake et al. [2003a] suggested.

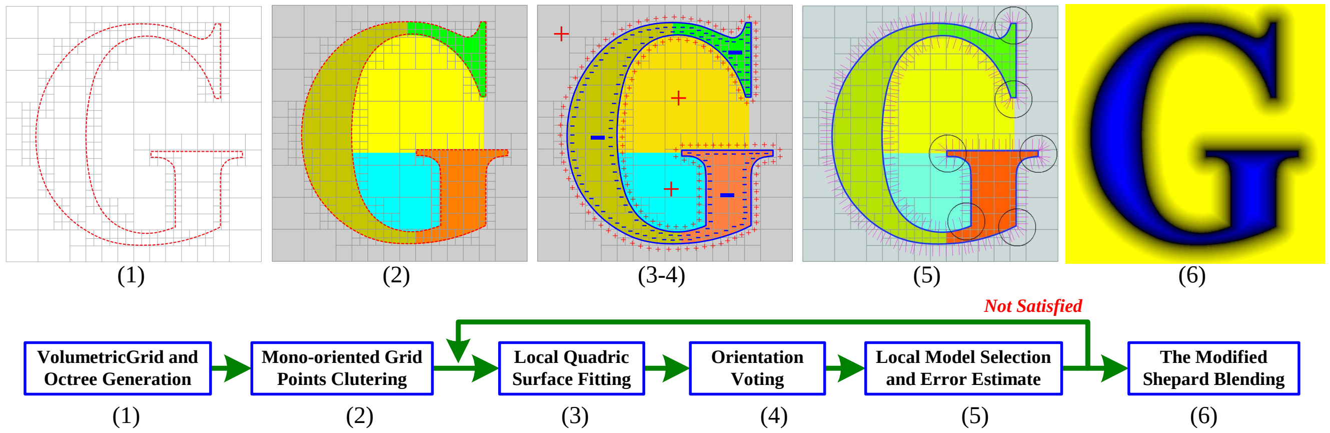

(1)
(2)
(3-4)
(5)
(6)
Not Satisfied
VolumetricGrid and Octree Generation
(1)
Mono-oriented Grid Points Clutering
(2)
Local Quadric Surface Fitting
(3)
Orientation Voting
(4)
Local Model Selection and Error Estimate
(5)
The Modified Shepard Blending
(6)

Figure 3: The algorithmic flow of our surface reconstruction system.

The resulting function $f(x)$ will be of C^n continuity if both w_i and f_i are C^n continuous. Blending the implicit functions naturally blends the implicit surfaces generated from these implicit functions. We refer readers to [Shepard 1968; Franke and Nielsen 1980; Renka 1988] for a background of the origin of this method, and some nice properties accompanying it. In comparison to other blending methods such as splines or NURBS, the modified Shepard's method poses no requirements on the local connectivity and no limitations to the valence (number) of nearby basis functions, and thus can be easily adapted to a variety of applications.

4 Algorithm Overview

The algorithmic flow of our novel surface reconstruction algorithm is illustrated in Fig. 3. (1) We first divide all sample points by a volumetric grid and insert them into an octree. All samples are inserted into the deepest level of the tree. Only empty cells can be leaf nodes (except for nonempty cells at the deepest level). The depth of the octree is determined comprehensively by the noise rate and the feature size of the modelled surface. Adaptively subdividing the octree in the fitting stage is possible but not preferable because Step 2 is not currently a locally adaptive algorithm. For level-of-detail fitting, the fitting is allowed to occur at non-leaf cells. (2) At each level, the grid points are clustered into a set of so-called "mono-oriented" groups using an active contour method. These groups are "mono-oriented" in the sense that all the grid points in the same group have the same orientation, ether inside or outside, but which side is not determined at this point. (3) We locally fit in each cell an implicit quadric surface to its nearby sample points. (4) The mono-oriented groups at the opposite side of each local implicit surface are expected to have opposite orientations. Each nonempty cell casts a vote for the orientation of its neighboring groups. The orientation of the grid point groups will in turn be used to decide the orientation of each locally fitted surface patch. Thus, the orientation can be aligned globally but efficiently using as much information as possible to avoid erroneous results. (5) We calculate the normal of each sample point as well as the fitting error and normal variation in each cell. Cells with normal changes or fitting error bigger than a threshold will undergo an additional fitting process (detected sharp edges indicated by circles). (6) The last step of our algorithm is to blend all the locally fitted implicit surfaces to produce a (pseudo-)signed distance field using the modified Shepard's method. In the figure, yellow indicates positive regions and blue, negative regions. The resulting signed distance field can be visualized by ray tracing, volume rendering techniques or be transformed to a polygon mesh using the Marching Cubes algorithm.

5 Mono-oriented Region Partition

In order to make our surface reconstruction algorithm as general as possible, we assume as input an unorganized point set without normal or orientation information. For data sets equipped with normal or interior/exterior direction, the step of the algorithm described in this section can also be necessary for noisy and untrustworthy normals — which are often the case.

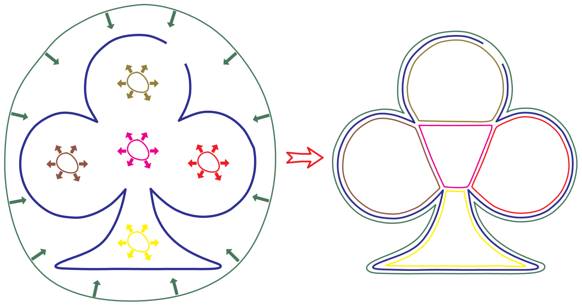


Figure 4: Mono-oriented region partition. Within each cave and recess of the surface as well as the outer bounding space an active contour is launched. The active contours can not penetrate the hole (on top of the club) when growing.

To blend signed distance fields correctly with the modified Shepard's method, the normal direction of each implicit surface patch must be correctly aligned; that is, the gradient of all implicit functions must point outward with respect to the surface. The traditional normal alignment method exploits a Minimum Spanning Tree (MST)-based orientation diffusion approach [Hoppe et al. 1994]. A basic assumption in their method is that the normals n_i, n_j of a pair of sufficiently close surface point p_i, p_j should point toward nearly the same direction, that is $\overrightarrow{n_i} \cdot \overrightarrow{n_j} \approx +1$. But Hoppe et al.'s method cannot easily accommodate sharp edges even if the propagation follows an optimal route along the MST. Consider a regular tetrahedron with accurate normals pointing outward at each surface point. The dot product of two normals across an edge is always negative. Hence, for this example, normal orientation can never be correctly aligned beyond a facet. Another problem with Hoppe et al.'s method is that, in the presence of noise, the locally estimated normals may not always be trustworthy. Moreover, normal alignment of distant islands in an incomplete data set may even be impossible.

Essentially, the effect of orientation alignment of a manifold surface is equivalent to dividing the volume space into two groups of regions: the inside regions and the outside regions. A commonly used approach to find the inside or outside of a surface is the active contour method. To avoid leakage through holes, most deformable models resort to the minimization of some complicated and expensive strain energies. In this paper, we follow a different idea to avoid leaking. We launch two active contours growing at both sides of the hole. If every point on these two active contours travels at the same speed and keeps the same distance to the surface, these two active contours will finally collide at the center of the hole. This leads to the following space partition algorithm, as shown in Fig. 4. We begin by launching an active contour within each cave and recess of the surface as well as in the outer bounding space. These active contours grow or shrink toward the surface and try their best to keep the same distance to the surface. At the end of this process the whole surface is sandwiched between active contours.

The subspace swept by a specific active contour lies as a whole inside or outside of the surface. We name these subspaces "mono-oriented regions." After the orientation of each mono-oriented region is decided, each patch of the surface can be easily oriented.

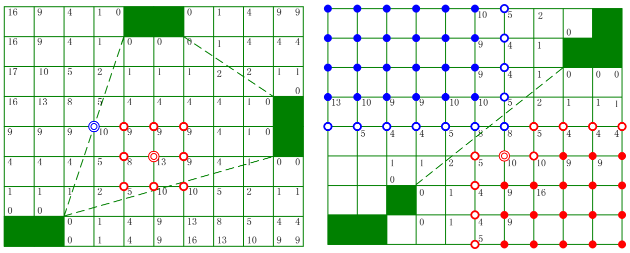

(a)
(b)

Figure 5: The mono-oriented region growing method. (a) The discrete unsigned distance field (actually, squared distances are marked in the figures) and initialization of active contour seeds. (b) Two active contours grow down the discrete distance field and collide in the middle of a hole.

At the start of the algorithm, we have no idea which regions are inside the surface and which are outside, let alone where to put the active contour seeds in each cave and recess. The idea of using medial axis transformation [Amenta et al. 2001] for surface reconstruction indicates that for any spatial region visually bounded by a set of surface samples, there exists at least a space point inside the surface such that all its neighboring space points are not further to the set of samples. That means this point is a local maximum of the unsigned distance field associated with this set of surface samples. We call such a point a (local) peak point (of the unsigned distance field). Hence, it is sufficient to launch an active contour seed at each peak point besides the one in the bounding space.

In practice, we discretize the above continuous approach onto a volumetric grid. We first build a discrete, unsigned distance field on the grid points. The distance value of a grid point is defined as its distance to the nearest nonempty cell (see Fig. 5 (a)). Every peak point on the grid initiates an active contour (the double-circled red point in Fig. 5(a)). Due to discretization, the double-circled blue point is falsely identified as an active contour seed. Since this seed will not be able to grow (try the algorithm in Fig. 7), it poses no harm. The true seed (the red double-circled point) can grow to its neighboring points and lead to a new mono-oriented region. If several peak points are consecutively connected, they share a single mono-oriented group ID.

All points on the active contours are sorted with a heap by their distances to the surface. The point on top of the heap (the furthest point, as double-circled in Fig. 5 (b)) grows first. Hence, all parts of the active contour grow with nearly the same distance to the surface. The grid points swept by the same active contour are clustered into a mono-oriented group. Fig. 5 (a) and Fig. 7 show the details of the algorithm.

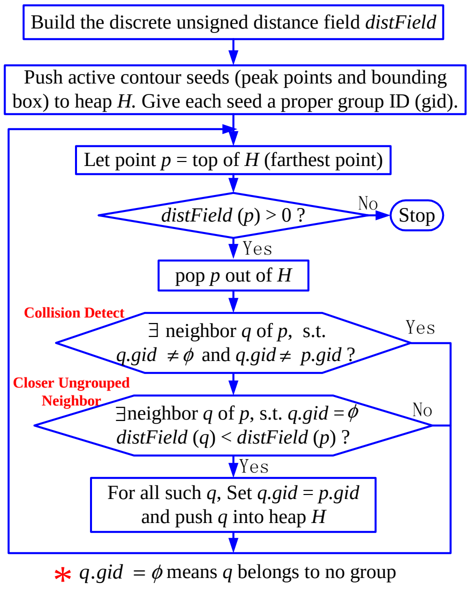

Build the discrete unsigned distance field *distField*
Push active contour seeds (peak points and bounding box) to heap *H*. Give each seed a proper group ID (gid).
Let point *p* = top of *H* (farthest point)
distField (*p*) > 0 ?
No
Stop
Yes
pop *p* out of *H*
Collision Detect
∃ neighbor *q* of *p*, s.t. *q.gid* ≠ φ and *q.gid* ≠ *p.gid* ?
Yes
Closer Ungrouped Neighbor
∃neighbor *q* of *p*, s.t. *q.gid* = φ *distField* (*q*) < *distField* (*p*) ?
No
Yes
For all such *q*, Set *q.gid* = *p.gid* and push *q* into heap *H*
* *q.gid* = φ means *q* belongs to no group

Figure 7: The mono-oriented group growing algorithm. Heap H is sorted in descending order by the distance to the nearest nonempty cell.

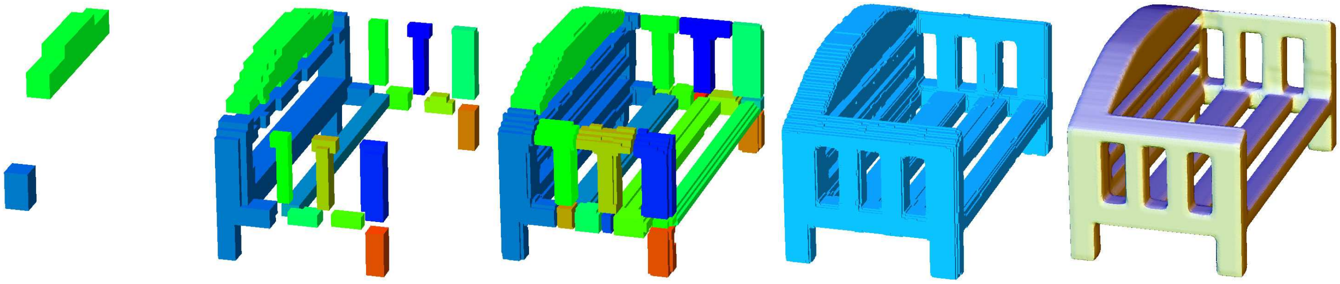
The above discussion leads to an $O(n^3)$ algorithm, where n is the diameter of the volumetric grid. We can improve the performance by introducing a hierarchical version of this algorithm. As shown in Fig. 8, we first perform the mono-oriented region partition algorithm at a coarse level, than we transform the active contour to a finer level and redo the region growing method on the finer level. Typically, it is a loop of the following four steps.

(1) Find new active contour seeds;
(2) Grow the mono-oriented regions;
(3) Transform the active contour to the next level of grid;
(4) Grow the mono-oriented regions;

The purpose of step (4) is to decrease the space available for new seeds, thus reducing the number of mono-oriented groups. We note that the active contours transformed to the next level shrink back a little. This is done intentionally in order to avoid penetration of the blue contour through holes. Left unchecked, such penetrations become larger in finer levels of the octree. As the algorithm descends the octree hierarchy, more and more topological and geometrical details are exposed. Big cells break into smaller ones, cavities come out and thus new peak points appear. At each level of the algorithm we add new peak points to the lists of the active contour.

In Fig. 6, we show the hierarchical growing process of the mono-oriented regions. The first three figures show the mono-oriented regions initiated by peak points inside the surface at level 5, 6 and 8 respectively. Figure (d) shows the boundary of the mono-oriented group initiated by the bounding box. (e) is the shaded model reconstructed with our system.

Since only the state of the grid points near the surface needs to be stored, we can use a hash table to store the status of all grid points. This allows us to obtain a roughly $O(n^2 \log n)$ algorithm assuming the Hausdorff dimension of the sample points is 2. The space complexity is also reduced to $O(n^2)$. Another even more significant aspect to the speedup of our algorithm is that, in our hierarchical approach, the distance field evaluation is conducted only between nearby grid points and cells (except near big holes). Efficient distance calculation of points far from the surface is never a trivial task.



(a) Inside surface, Level 5 (b) Inside surface, Level 6 (c) Inside surface, Level 8 (d) Outside surface, Level 8 (e) Shaded model

Figure 6: Mono-oriented groups growing method. The first three figures show mono-oriented regions inside the surface, while (d) is the active contour initiated by the outer bounding box.

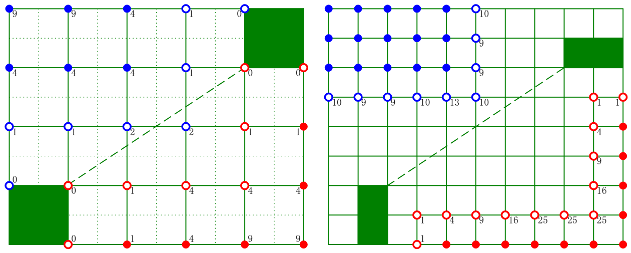


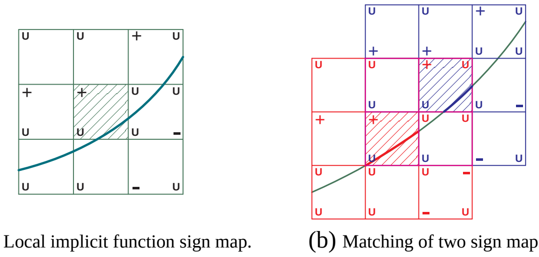
Figure 8: Hierarchical volumetric mono-oriented region partition. The result of the coarse level is transformed to a finer level.

6 Orientation Determination

By this stage of the surface reconstruction process we have partitioned the volumetric grid into mono-oriented regions. However, the orientation of each region (insideness/outsideness) is yet to be determined. For each locally fitted implicit surface patch, its neighboring grid points are divided into two oppositely oriented groups. We can make a local sign map of the implicit function at those grid points within a specific range of distance to the surface. A typical choice of the range is 1 to 3 times the grid size. In Fig. 9 (a), as an example, we mark the sign on those grid points with '+' or '-' and all others undetermined('U'). In Fig. 3 (steps 3-4), we show the orientation determination of the mono-oriented regions by voting. Each local implicit sign map matches to its nearby oriented mono-oriented groups and gives an orientation vote. The outermost mono-oriented region is obviously outside. Then, the orientations of inner regions are determined to maximize the number of matches to their boundary local sign maps. This voting process needs to be conducted several times till the innermost regions be properly oriented. After the orientations of all mono-oriented groups being determined, they will be subsequently exploited to determined the orientation of each local implicit patch using local sign maps. Very often not all surface patches can be aligned with their nearby mono-oriented regions (this may due to the existence of too small a cavity to hold a mono-oriented region). The orientation alignment algorithm of adjacent surface patches can then be applied. Two adjacent maps are called a positive match if there is no conflict of signs at their common grid points; that is, no + and - meet at the same grid point. A negative match requires that one sign must be flipped in order to give a positive match. In Fig. 9(b), the two adjacent implicit surfaces have a positive match.

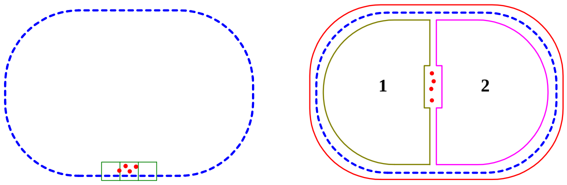
7 Outlier Detection

Outliers, which do not truly belong to the surface, may exist in defective data sets. There are two types of outliers. The first type lies not too far away from the valid samples (Fig. 10(a)). The octree cells containing these outliers have sufficient valid samples nearby when doing local surface fitting. Thus, the outliers can be detected and removed by measuring the distance to the fitted surfaces. Another type of outlier clusters with others at a distant place from the surface. These outliers can erroneously produce a surface patch if not properly removed. Our mono-oriented groups can help identify this type of outlier. As shown in Fig. 10(b), two mono-oriented groups envelop the outliers in the center. But the orientations of groups 1 and 2 are both marked inside after the voting step because we assume outliers are rare. Hence, the local implicit surface fitting these outliers can not be oriented, and outliers are identified.



(a) Local implicit function sign map. (b) Matching of two sign maps.

Figure 9: The local orientation map associated with the center cell and the orientation alignment of two adjacent implicit surfaces. 'U' stands for uncertain.

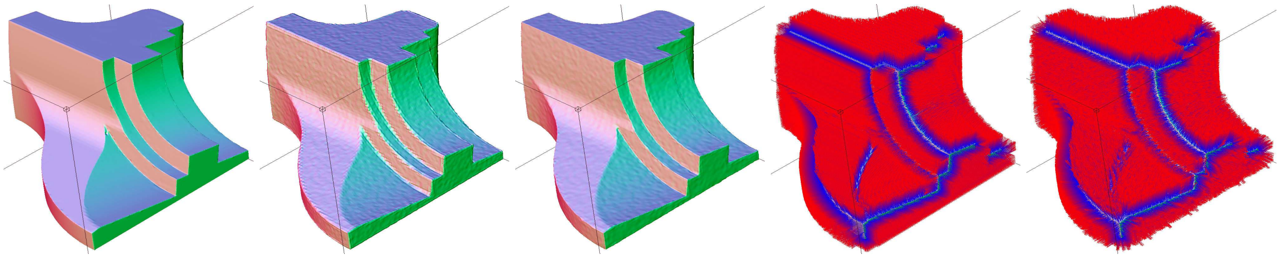


(a) Near surface outliers. (b) Distant outliers.

Figure 10: Two types of outlier removal methods.

8 (Abrupt) Noise Removal

A critical aspect of our surface reconstruction algorithm is a new iterative noise removal method we describe in this section. In the first iteration of the surface fitting process, we fit an implicit quadric surface to the sample points in the cell and all its neighboring cells with a weight function of compact support R (Fig. 12(a)). Then, we project all the sample points in this cell to the fitted implicit surface and term them the "projected points" (Fig. 12(b)). Normal vectors associated with each projected point are also computed for feature detection. In the subsequent fitting iterations, we use



(a) Reconstructed model from noise-free data set. (b) Recovered surface after one iteration of noise removal. (c) After five iterations. (d) Reconstructed normals for (a). (e) Reconstructed normals for (c).

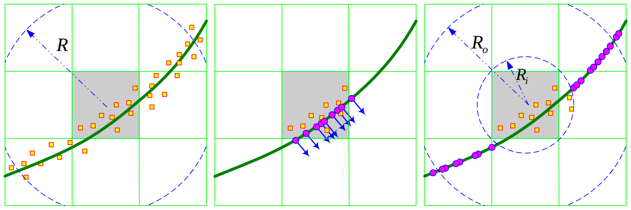
Figure 11: Feature recovery of noisy point cloud without prescribed normals.

the original sampling points within the sphere of radius R_i, while use the projected points in a spherical shell with a larger support R_o (Fig. 12(c)). The projected points should be assigned heavier weights. A typical selection is:

$$W_{orig}(p) = \begin{cases} \phi(\|p-c\|, R_o) & \text{if } \|p-c\| \le R_i \\ 0 & \text{if } \|p-c\| > R_i \end{cases}, \qquad (2)$$

$$W_{proj}(p) = \begin{cases} s * \phi(\|p-c\|, R_o) & \text{if } R_i < \|p-c\| < R_o \\ 0 & \text{otherwise} \end{cases}, \qquad (3)$$

where $\phi(\cdot, R_o)$ is a radial basis function of compact support R_o, c is the center of the inner cell and s is a constant which we name the stress factor. Bigger R_o result in relatively smoother fitting. A typical R_i can be one half of R_o. The selection of s depends on the type of radial basis function ϕ. Too big a s may result in non-convergency or bumpy results. We use $s = 2$ and the spline weight function $\phi(x,R) = b(3x/2R)$ in our implementation, where b is a B-spline basis function. The optimal choice of s and its role in convergency is still an open issue. The original samples in the spherical shell between radii R_i and R_o can also be used for the fitting. In this case, $W_{orig}(p) = \phi(\|p-c\|, R_o)$ for all $\|p-c\| < R_o$. This often gives better convergency.

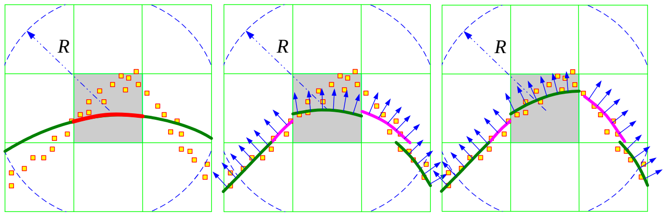


(a) Iteration 1. Fitting an implicit quadric patch to the original samples. (b) Iteration 1. Projecting samples in the inner cell to the local surface patch. (c) Iteration 2. Fitting to the projected points as well as the original samples.

Figure 12: Iterative noise removal using projected points.

9 Feature Detection and Local Model Selection

In order to recover sharp edges and corners, we propose a new feature detection scheme for noisy data sets following the spirit of Ohtake et al.'s algorithm [Ohtake et al. 2003a]. Generally, we choose one of the three candidate local fitting models according to the local normal distribution. For flatter surface areas, we use bivariate parametric quadrics defined in the local tangent plane; for areas where there are two sheets of surfaces, we use general quadrics; and for sharp edges and sharp corners, we exploit piecewise quadric surface models. The choice is made by evaluating the maximum span angle of nearby point normals. Reliable and accurate normal evaluation are required to detect and recover features reliably in noisy data sets.



(a) Fitting sharp features with bivariate parametric quadrics often smooths out features (only one patch for the inner cell shown). (b) Fitting sharp features with bivariate parametric quadrics after the first iteration (all patches in non-empty cells shown). (c) After several iterations of fitting with error tuned weight functions, a feature is now detected based on normal distribution.

Figure 13: Iterative feature recovery using projected points.

In the first iteration, the system uses bivariate parametric quadrics as the fitting model for all cells. This results in larger fitting error in cells with sharp features (see Fig. 13(a)). Also, the normals evaluated from the first iteration are very noisy. Thus, for the second iteration, we may falsely detect or miss features using Kobbelt et al.'s criteria [2001]. In this paper, we propose an iterative feature detection algorithm that is naturally integrated with our iterative noise removal method. The basic idea is to assign various weight stress factors to the projected points in neighboring cells based on their fitting error or estimated curvature. For the example in Fig. 13, the fitting of a sharp edge using a parametric bivariate quadric results in flattened surfaces. The feature detection fails to identify the feature. However, for the fittings in the next iteration, the points near the feature will have less impact on the implicit fitting in its neighboring cells. Thus, the overall distribution of normals will gradually form a feature (Fig. 13(c)). After this, a feature is detected with Kobbelt et al.'s criteria. The fitting model for this cell in the the next iteration will be replaced by piecewise quadrics as discussed by Ohtake et al[2003a]. To fit a piecewise quadric model, we first group the projected points by their normal directions similar to Ohtake et al.'s[2003a] approach. For the original sample points in the inner cell (whose normal vectors are not used), we group them by measuring their distances to the locally fitted plane of each group of previously projected points. (That is, we locally fit each group of projected points to a plane that interpolates the averaged center of the projected points and that has as its normal vector the averaged normal of the group of projected points.) The weight function typically takes the form: $w = \phi(r,R)\phi(e,\varepsilon)\phi(v,\eta)$, where, r is the distance to the center, R is the local support, e is the average fitting error of the cell containing that point, ε is a user-specified expected error, u is local normal change (maximum span angle), and η is a user specified expected normal change, we usually take the value of 30 degrees.

Fig.11 presents some results of applying our feature recovery algorithm to noisy data set. Fig.11 (a) is the reconstructed surface from noise-free data without normals. Fig.11(b) and (c) are the results of applying our algorithms 1 and 5 iterations respectively, to data set with 5% of noise. Fig.11(d) and (e) are the normal maps of (a) and (c) respectively.

10 Experimental Results

Due to the space limitations, we only show several representative experimental results for the four main functionalities of our algorithm: automatic hole filling, abrupt noise removal, dealing with nonuniform samples in absence of normal information, and feature recovery from noisy data sets. In Fig.14, a surface with complex topology is reconstructed. Fig.14 (a) shows two of the mono-oriented regions where the inner core is successfully determined as an outside mono-oriented region. In Fig.15, the noise in the data set with complex topology is iteratively removed. Fig.16 shows a close-up view of the fandisk model from Fig.11. We can see that the sharp features are well captured and recovered. Fig.16(a) clearly shows the effect of the feature detection threshold. Fig.16(c) and Fig.16(d) are models recovered from the fandisk data set with noise added. Fig.17 is an example of sharp feature detection and recovery of noise-free data sets without normals. Our algorithm can also process non-uniformly sampled data sets as shown in Fig.19. Using different levels of the octree, we can obtain multiresolution surface recovery. At a finer level of the octree, details can be well recovered as shown by the dinosaur skin in Fig.18.

11 Conclusion and Future Work

We have presented a novel, noise-tolerable surface reconstruction method with partition of unity implicits. Compared to other surface reconstruction methods that handle noisy point clouds, our method possesses both efficiency and robustness. Our system can accept data sets subject to noise, bursts of noise, missing data, and varying sampling density without normal or orientation information. The basic idea behind this work is (1) to avoid using local unreliable information to determine global properties (e.g. orientation); (2) to detect false results iteratively and to redo the local fitting as necessary. The techniques we develop for space partition, iterative fitting, feature detection, and normal alignment can be adapted to a large variety of applications in computer graphics, vision, medical image segmentation, etc. Our method significantly improves the previous MPU implicit method to accommodate general data sets. We designed a robust hierarchical volumetric partition algorithm to distinguish the interior and exterior of the modelled surface and employed a divide-and-conquer approach to surface fitting. Each locally fitted surface patch is blended to produce a smooth (pseudo-)signed distance field using the modified Shepard's method. It avoids the expensive process of global implicit function fitting, while preserving the desirable nature of the implicit approaches – namely, automatic hole filling. Our iterative local fitting algorithm also features topology recognition and geometry recovery in the presence of noise.

Some future work includes: (1) design a locally adaptable mono-oriented region growing algorithm. This algorithm makes possible the locally adaptive level-of-details fitting; (2) automatic level-of-detail checking and recovery from noisy data; (3) automatic determination of the optimal support radius of the weight function as well as other user specified parameters—the expected error and the expected normal change.

References

Amenta, N., Choi, S., and Kolluri, R. K. 2001. The power crust. *Proceedings of the Sixth ACM Symposium on Solid Modeling and Applications*, 249–266.

Bajaj, C. L., Bernardini, F., and Xu, G. 1995. Automatic reconstruction of surfaces and scalar fields from 3D scans. *Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive Techniques*, 109–118.

Blinn, J. F. 1982. A generalization of algebraic surface drawing. *ACM Transactions on Graphics 1*, 3 (July), 235–256.

Boissonnat, J.-D. 1984. Geometric structures for three-dimensional shape representation. *ACM Trans. Graph. 3*, 4, 266–286.

Carr, J. C., Beatson, R. K., Cherrie, J. B., Mitchell, T. J., Fright, W. R., McCallum, B. C., and Evans, T. R. 2001. Reconstruction and representation of 3d objects with radial basis functions. *Computer Graphics Proceedings, ACM SIGGRAPH Annual Conference Series* (August), 67–76.

Curless, B., and Levoy, M. 1996. A volumetric method for building complex models from range images. *Computer Graphics Proceedings, ACM SIGGRAPH Annual Conference Series*, 303–312.

DeCarlo, D., and Metaxas, D. 1995. Adaptive shape evolution using blending. *In Proc. Fifth International Conference on Computer Vision, Boston*, 834–839.

Edelsbrunner, H., and Mcke, E. P. 1994. Three-dimensional alpha shapes. *ACM Transactions on Graphics (TOG) 13*, 1, 43–72.

Franke, R., and Nielsen, G. 1980. Smooth interpolation for large sets of scattered data. *Int. Journal for Numerical Methods in Engineering 15*, 1691–1704.

Greg, T., and O'Brien, J. 1999. Variational implicit surfaces. *Tech Report GIT-GVU-99-15, Georgia Institute of Technology, May 1999, 9 pages.*.

Hoppe, H., DeRose, T., Duchamp, T., McDonald, J., and Stuetzle, W. 1992. Surface reconstruction from unorganized points. *Computer Graphics 26*, 2, 71–77.

Hoppe, H., DeRose, T., Duchamp, T., Halstead, M., Jin, H., McDonald, J., Schweitzer, J., and Stuetzle, W. 1994. Piecewise smooth surface reconstruction. *Computer Graphics Proceedings, ACM SIGGRAPH Annual Conference Series* (July), 295–302.

Kass, M., Witkin, A., and Terzopoulos, D. 1987. Snakes: active contour models. *International Journal of Computer Vision 1*, 4, 321–331.

Kobbelt, L. P., Botsch, M., Schwanecke, U., and Seidel, H.-P. 2001. Feature-sensitive surface extraction from volume data. In *SIGGRAPH 2001, Computer Graphics Proceedings*, ACM Press / ACM SIGGRAPH, E. Fiume, Ed., 57–66.

McInerney, T., and Terzopoulos, D. 1995. Topologically adaptable snakes. *ICCV*, 840–845.

Mello, V., Velho, L., and Taubin, G. 2003. Estimating the in/out function of a surface represented by points. *Symposium on Solid Modeling and Applications 2003*, 108–114.

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

Muraki, S. 1991. Volumetric shape description of range data using Blobby Model. *Computer Graphics 25*, 4, 227–235.

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

Ohtake, Y., Belyaev, A. G., and Seidel, H.-P. 2003. A multi-scale approach to 3d scattered data interpolation with compactly supported basis function. *Shape Modeling International 2003, Seoul, Korea* (May), 153–164.

Osher, S., and Fedkiw, R. 2002. *The Level Set Method and Dynamic Implicit Surfaces*. Springer-Verlag.

Renka, R. J. 1988. Multivariate interpolation of large sets of scattered data. *ACM Transactions on Mathematical Software (TOMS) 14*, 2, 139–148.

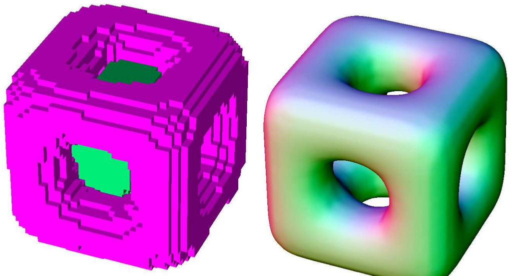
Sethian, J. A. 1999. *Level Set Methods and Fast Marching Methods : Evolving Interfaces in Computational Geometry, Fluid Mechanics, Computer Vision, and Materials Science*. Cambridge University Press.

Shepard, D. 1968. A two-dimensional interpolation function for irregularly-spaced data. *Proceedings of the 1968 23rd ACM national conference*, 517–524.

Whitaker, R. 1998. A level-set approach to 3d reconstruction from range data. *Int'l J. Computer Vision 29*, 3, 203–231.

Xie, H., Wang, J., Hua, J., Qin, H., and Kaufman, A. 2003. Piecewise C^1 continuous surface reconstruction of noisy point clouds via local implicit quadric regression. *14th IEEE Visualization Conference* (October), 91–98.

Zhao, H.-K., Osher, S., and Fedkiw, R. 2001. Fast surface reconstruction using the level set method. *1st IEEE Workshop on Variational and Level Set Methods, 8th ICCV, Vancouver*, 194–202.



(a) Two of the mono-oriented regions.

(b) The recovered surface.

Figure 14: Reconstructed model featuring complex topology.

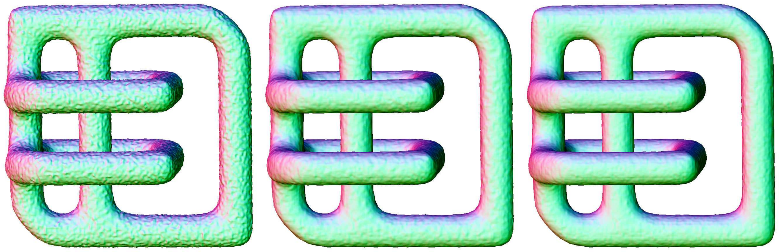


Figure 15: A noisy model smoothed after 1, 2 and 5 iterations, respectively, of the noise removal algorithm.

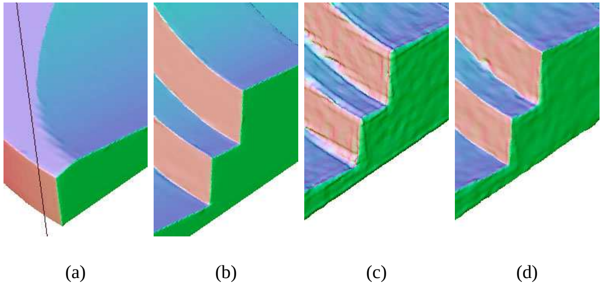


Figure 16: Detail of reconstructed fan disk. (a-b) Original data-set. (c-d) Model recovered after adding 0.5% noise. (c) 1 iteration. (d) 30 iterations.

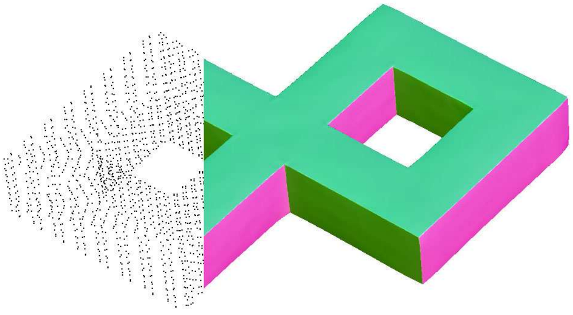


Figure 17: Sharp features can be recovered very accurately.

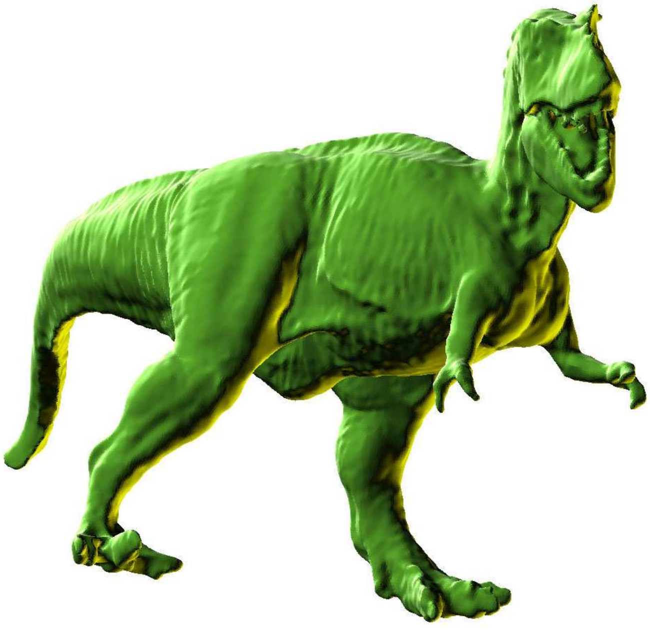


Figure 18: Our algorithm reconstructs details very effectively.

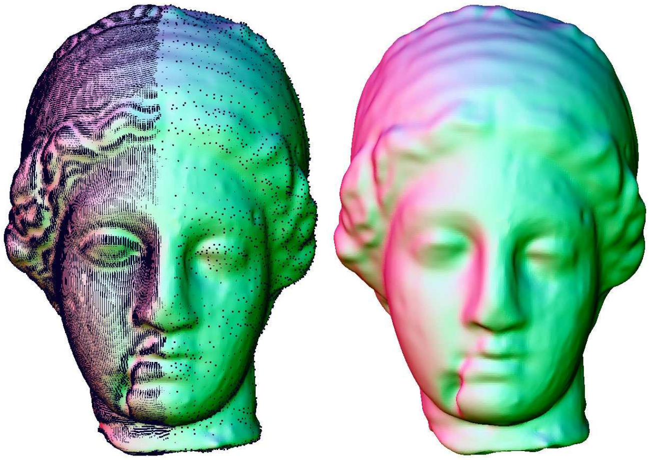


Figure 19: Our algorithm can process non-uniformly-sampled data sets without difficulty.

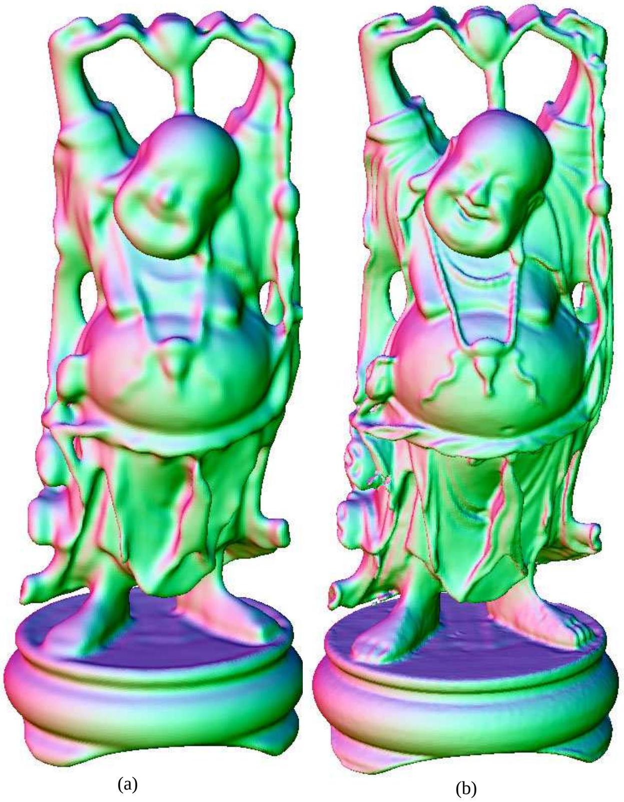


Figure 20: Surface reconstruction at varying levels of detail. (a) Octree level 6. (b) Octree level 8.