

# Piecewise $C^1$ Continuous Surface Reconstruction of Noisy Point Clouds via Local Implicit Quadric Regression

Hui Xie

Jianning Wang

Jing Hua

Hong Qin

Arie Kaufman \*

Center for Visual Computing (CVC) and Department of Computer Science  
Stony Brook University, Stony Brook, NY 11794-4400, U.S.A.

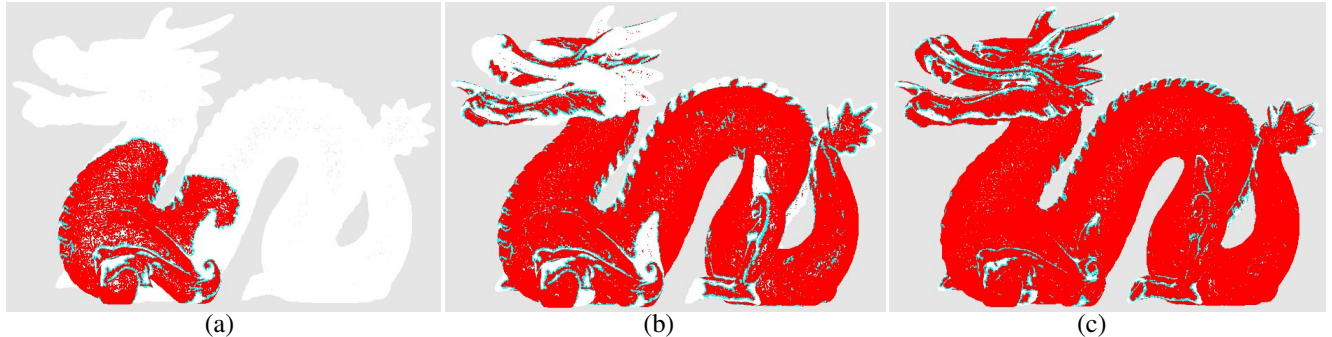


Figure 1: Prioritized incremental surface fitting. Red stands for the already fitted surface areas, green for the growing front and white for the untouched regions. The already fitted surface grows along an easier way, that is, along surface regions with smaller curvature and less noise.

## Abstract

This paper addresses the problem of surface reconstruction of highly noisy point clouds. The surfaces to be reconstructed are assumed to be 2-manifolds of piecewise  $C^1$  continuity, with isolated small irregular regions of high curvature, sophisticated local topology or abrupt burst of noise. At each sample point, a quadric field is locally fitted via a modified moving least squares method. These locally fitted quadric fields are then blended together to produce a pseudo-signed distance field using Shepard’s method. We introduce a prioritized front growing scheme in the process of local quadrics fitting. Flatter surface areas tend to grow faster. The already fitted regions will subsequently guide the fitting of those irregular regions in their neighborhood.

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

**Keywords:** Computer Graphics, Surface Reconstruction, Point Cloud, Surface Representation, Solid Modeling, Moving Least Squares, Shepard’s Method.

## 1 Introduction

\*email:{xhui | jianning | jinghua | qin | ari}@cs.sunysb.edu

The problem of reconstructing a surface from a large set of scattered sample points arises in a variety of applications in the fields of reverse engineering, computer graphics, computer vision, medical image segmentation, etc. Generally speaking, surface reconstruction from scattered samples is an ill-posed problem, i.e., there is no unique solution. Constraints on both the original surface and the sample points are required to guarantee a faithful and unique solution. In the last decade, a number of surface reconstruction techniques have been proposed. Despite of the versatility of these large amount of algorithms at present, however, most existing algorithms make certain strong assumptions on the original surface and its sample points. For example, many algorithms necessitate a very dense sampling to be able to capture drastic topological or geometric change in a small region; some approaches need additional knowledge such as surface normal or interior/exterior information; some algorithms are not tolerant of noise and corrupted data.

The goal of this paper is to develop a robust surface reconstruction algorithm that can handle noisy data sets and tolerate critical situations, such as abrupt bursts of high noise and topological ambiguity where two objects come too close together. These problems arise frequently for 3D data acquisition techniques such as those in stereo photogrammetry.

## 2 Problem Statement and Contributions

Before proceeding with our algorithm, an explanation may be desirable of our major concerns in our algorithm design, that is, the problems we intend to solve.

A correct reconstruction of a surface is possible only if it is “properly” sampled, i.e., there are enough sample points in areas of high curvature or great topological complexity. One representative criteria of an adequate sampling rate was suggested by Amenta [2001], the *r-sampling* criteria. Generally, higher sampling rates are required where two parts of the surface come close or the local curvature is high. Noise can also inflict trouble for topological recognition.

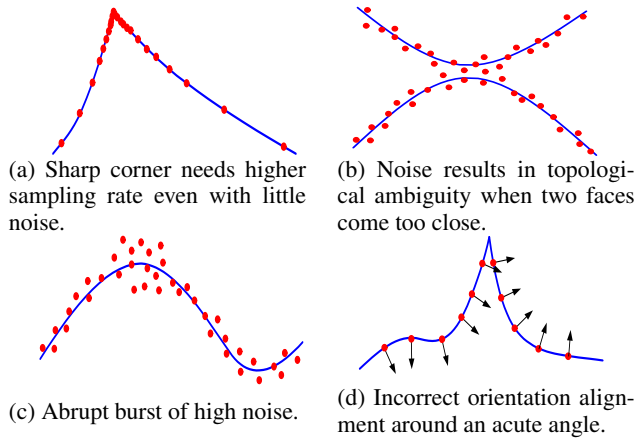


Figure 2: Reconstruction difficulties due to sharp edges and noise.

In Fig. 2(a), we observe that on both sides of the acute sharp corner are surfaces of low curvature, but a sampling rate of infinity is needed for a correct reconstruction of this sharp feature in most algorithms. In Fig. 2(b), the topology of the actual surface is obscured where the standard deviation of noise is larger than the local feature size. In Fig. 2(c), the noise rate in a small region is abruptly high. In Fig. 2(d), many front propagation-based normal alignment algorithms can not correctly identify the orientation abnormality near an acute sharp edge.

Most existing algorithms can only partially solve these problems under certain strong assumptions on either the sampling rate, the smoothness of the underlying surface, or the noise power. But visually, our human eyes are able to capture the shape of a noisy and under-sampled model with ease. This kind of ability can be attributed to several assumptions on the underlying surface in our brain: (1) The major part of a surface is absent of abrupt curvature change and burst of noise; (2) Abrupt changes of curvature or topology do not cluster in a small region. (3) The underlying surface is a 2-manifold.

Based on these assumptions, we have designed a prioritized surface growing algorithm. In this prioritized front propagation algorithm, the system first fits the “good” parts of the surface, then use the already built parts to guide the fitting of “bad” parts of the surface. We also use implicit quadrics instead of parametric surfaces as basic models in the local fitting. The use of implicit surfaces is mainly for better sharp feature representation and easier blending of locally fitted surface patches via Shepard’s method. The combination of these two techniques would allow for a reliable surface reconstruction from noisy data points with surface reconstruction hazards mentioned above. In addition, we propose a robust orientation alignment algorithm and several techniques for efficient evaluation of surface properties such as variation of curvature.

### 3 Previous Work

Generally, the existing approaches to constructing a surface model from a set of scattered points fall into two categories. One is the optimal triangulation of a discrete set of spatial points. The other is function reconstruction from a set of samples, or, data fitting. In essence, the problem of surface reconstruction has two aspects: topology recovery and geometry recovery. Triangulation-based algorithms mainly focus on faithful recovery of a flawless topology from relatively accurate but possibly incomplete data sets. On the contrary, data fitting based algorithms concentrate much more on regression analysis of inaccurate data samples, and try to recon-

struct the underlying function even in existence of random noise.

#### ■ Piecewise-linear surface reconstruction

Algorithms falling into this category include the  $\alpha$ -shape [Edelsbrunner and Mücke 1994], Power Crust [Amenta et al. 2001] and the ball pivoting algorithms [Bernardini et al. 1999]. Some algorithms from this group can provide provable topological correctness, such as the Power Crust. Although some variants are proposed, most of these algorithms assume the availability of a noise-free data set.

#### ■ Surface reconstruction through the building of a signed-distance field

The application of implicit models in the field of surface reconstruction are primarily due to two reasons: (1) In contrast to piecewise surface patches, an implicit surface involves no specific topology-sensitive parametric domain and thus avoids any complicated procedures such as surface trimming and patching; (2) An implicit function defines a simulated signed distance field in the 3D space. A weighted sum of two distance fields intuitively averages the two underlying surfaces. This intrinsic averaging property allows for both multiple data sets registration and sample denoising, which in many algorithms rely on a preprocessing or postprocessing stage.

Hoppe, DeRose, et al. [1994] first introduced the signed distance field as a basic tool for reconstructing surfaces from a set of range images. Curless and Lovoy [1996] soon proposed an algorithm based on constructing a distance field on volumetric grid. Bajaj, Bernardini and Xu [1995] combined the piecewise algebraic surface with  $\alpha$ -shape. Carr, et al. [2001] explored the use of radial basis functions for uncomplete data set. Zhao, Osher, et al. [2000] proposed a level set method driven by dynamic PDEs. All these methods have displayed the merits of using implicit surfaces, that is, noise tolerable, topology flexible, and easy to convert to other representations.

#### ■ Moving least squares and Shepard’s method

The moving least squares method and Shepard’s method can date back to McLain and many other’s work [McLain 1974; Shepard 1968; Renka 1988; Lancaster and Salkauskas 1981]. In the meantime, in the computational statistics community, the local regression method (similar to the moving least squares) was extensively studied for noisy data smoothing. Many issues on the choice of the local fitting functions, the selection of the bandwidth (the support of the local weight function) are discussed in depth. We refer readers to [Cleveland and Loader 1995] for details. The early moving least squares method was dedicated to function fitting, where a parametric domain is innate. Levin [1998] proposed the *moving least squares surfaces* by introducing local reference frames. Based on Levin’s approach, Lee [2000] implemented a curve reconstruction method which was capable of automatically selecting local bandwidth. The recent *point set surfaces* [Alexa et al. 2001] are also based on moving least squares fitting. In this paper, we further extend the local function family to implicit quadrics. This extension to local quadric fitting allows more accurate discovery of the local topology and geometry under high noise. Ohtake, et al. [2003] also developed a hierarchical surface fitting system based on Shepard’s blending to handle accurate data set with prescribed normals.

### 4 Moving Least Squares and Shepard’s Method

Both the moving least squares method of Levin’s type and Shepard’s method are employed in our paper. Strictly speaking, this paper only borrows from moving least squares the technique of local quadratic function fitting by introducing a local frame. The “moving” in moving least squares has been replaced by Shepard’s methods.

## 4.1 Moving Least Squares Method

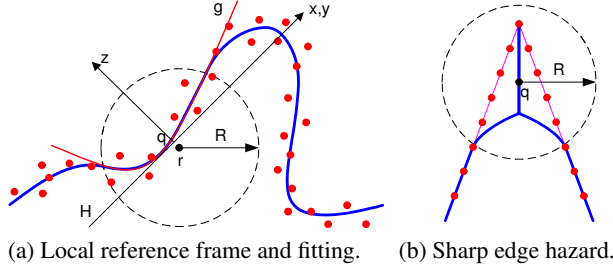


Figure 3: Standard moving least squares fitting.

The standard moving least squares method is illustrated in Fig. 3. Let points  $\{p_i \in \mathbb{R}^3, i = 1, \dots, N\}$  be sampled from a surface  $S$  with possible measurement noise. For an arbitrary test point  $r \in \mathbb{R}^3$  (not necessarily from  $\{p_i\}$ ) near the surface, we first find a local reference domain (plane)  $H$  for  $r$ . The local plane  $H = \{x | \langle n, x \rangle - D = 0, x \in \mathbb{R}^3, n \in \mathbb{R}^3, \|n\| = 1\}$  is defined so as to minimize

$$\sum_{i=1}^N (\langle p_i, n \rangle - D)^2 \theta(\|p_i - q\|), \quad (1)$$

where  $\theta$  is a nonnegative, smooth, radial, monotonously decreasing function, usually with finite support  $R$  as shown in Fig. 3(a),  $q$  is the perpendicular foot of  $r$  on plane  $H$  and  $\langle \cdot, \cdot \rangle$  denotes dot product. When  $r$  is close enough to the surface, we can just use  $q = r$  to lead the minimization of (1) to a linear system. The reference domain for  $r$  is subsequently used to compute a local bivariate polynomial approximation  $g(x, y)$  in a neighborhood of  $r$ . This process is carried out through minimizing a weighted least squares error

$$\sum_{i=1}^N (g(x_i, y_i) - z_i)^2 \theta(\|p_i - q\|), \quad (2)$$

where  $(x_i, y_i, z_i)$  are the coordinates of  $p_i$  in the local coordinate system induced by  $H$ , see Fig. 3(a). Levin defines  $S_p(r) = q + g(0, 0)n$ , i.e., the intersection of axis  $z$  with the local fitting polynomial  $g$ . We refer readers to [Alexa et al. 2001] for details. Levin’s marvelous result on this projection function  $S_p(r)$  is that under certain conditions, all the fixed points, that is,  $\{r \in \mathbb{R}^3 : S_p(r) = r\}$ , define a 2-manifold [Levin 1999]. Practically, these conditions may not always be well-satisfied. An acute angle may generate a non-manifold surface as shown in Fig. 3(b) (blue curves). The reference planes for all  $r$ ’s near the apex go across the axis of the cone. Taking  $q = r$  may make this situation better, but a much thinner pinnacle is inevitable. Our new solutions to this problem will be discussed in the following sections.

## 4.2 Shepard’s Method

With the moving least squares method, every surface point evaluation will invoke two stages of minimization, one for the calculation of a reference plane, another for a local polynomial fitting. This can be prohibitively expensive for practical applications. One way to reduce the amount of computation is by just evaluating a set of representative test points and then figuring out the position of in-between points by locally blending its neighboring representative points.

We first take as an example the case for approximating a function  $\mathbb{R}^2 \rightarrow \mathbb{R}$ . In Fig. 4, the approximated function values  $\hat{f}(c_i)$  at a set of representative points  $c_i$  are known (either were given or evaluated at a surface locally fitted to a set of neighboring samples), and

we want to figure out the function value at  $p$  by a linear combination of the values  $\hat{f}(c_i)$  at all  $p$ ’s neighboring representative points. We associate with each representative point  $c_i$  a smooth, positive and monotonously decreasing function  $w_i(\|p - c_i\|)$ , which is the weight function associated with  $c_i$ . Note that it is not necessary for the weight functions associated with each representative point to be the same. Now we define the value  $f(p)$  to be (see Fig. 4):

$$f(p) = \frac{\sum_{i=1}^n \hat{f}(c_i) w_i(\|p - c_i\|)}{\sum_{i=1}^n w_i(\|p - c_i\|)}. \quad (3)$$

To obtain computational efficiency, the weight functions  $w_i(\|p - c_i\|)$  are mostly compactly defined, that is, there exists a set of  $R_i$ ,  $w_i(\|p - c_i\|) = 0$ , if  $\|p - c_i\| > R_i$ . Interpolation can be achieved by letting  $w_i(\|p - c_i\|) \rightarrow +\infty$ , when  $\|p - c_i\| \rightarrow 0$ . We observe that  $\hat{f}(c_i)$  in (3) are a set of constants. Blending function values at a set of representative points  $c_i$  may result in an unappealing flat spot at  $c_i$ , and a steep interim in the middle of a pair of representative points [Shepard 1968; Franke and Nielsen 1980]. Normally,  $\hat{f}(c_i)$  is obtained by evaluating at  $c_i$  a locally fitted function  $\hat{f}_i(p)$  of a set of nearby sample points. Instead of blending sample function values, we can blend local representative functions. We replacing the term  $\hat{f}(c_i)$  by  $\hat{f}_i(p)$ , which is the local approximation of  $f(p)$  around  $c_i$ . This leads to the modified Shepard’s method,

$$f(p) = \frac{\sum_{i=1}^n \hat{f}_i(p) w_i(\|p - c_i\|)}{\sum_{i=1}^n w_i(\|p - c_i\|)}. \quad (4)$$

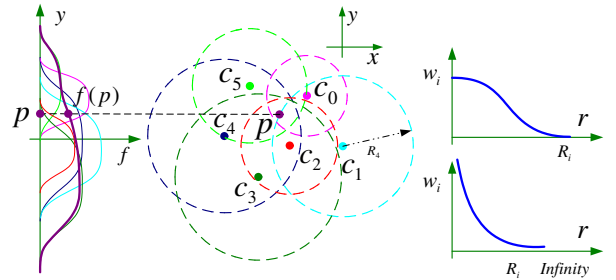


Figure 4: Shepard’s local blending method. In the middle are a set of representative functions with their centers  $c_i$  and local support  $R_i$  marked. To the left are the projections of each weighted representative functions. To the right we show two classical weight functions.

The result function  $f(p)$  would be of  $C^\infty$  continuity when  $w_i(\|p - p_i\|)$  are  $C^\infty$  continuous. 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 accompany this approach. In comparison to other blending methods such as splines, NURBS (we can see they share similar formulations), the modified Shepard’s method imposes no limitations on the local connectivity and number of local blending kernels, and thus can be easily fitted to a large variety of applications.

All the above addresses the blending of a real-valued function. To blend 3D surface patches, it would be rather challenging. One possible approach is by following the spirit of Levin’s local reference frame. But this may lead to many subtle questions such as how to blend local frames, how to deal with multi-sheeted surface patches resulting from coordinates transformation, etc. In this paper, we utilize the intrinsic surface “averaging” behavior when summing up several signed distance fields. In our algorithm, each surface is represented implicitly as the zero-level set of a trivariate real-valued function  $f(x, y, z)$ . Thus, surface blending through Shepard’s method is a direct application of this method to the case of 3D functions.

## 5 Incremental Local Fitting and Denoising

In this section, we introduce our prioritized incremental local regression algorithm. Along with the local model selection to be discussed in the next section, this algorithm can provide an appealing smoothing effect with great robustness.

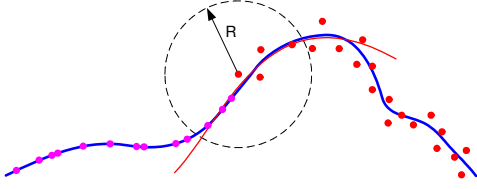


Figure 5: The already fitted points are subsequently used to guide the local fitting of its neighboring points by being assigned with a heavier weight.

The basic idea behind this algorithm is that the majority of real world models are comprised of well-sampled surface regions of low curvature and of low noise (hereinafter, “regular regions”) sparsely interspersed with small regions of sharp edges, peaks, bulges, and regions subject to high measurement noise (hereinafter, “irregular regions”), and a natural extension of these regular regions can present a very good prediction to the surface behavior in irregular areas.

In our new incremental surface construction system, multiple active seeds iteratively traverse “regular regions” of the samples before “irregular regions”. In Fig. 5, a seed is propagating from left to right. The current active point is shifted to a more smooth-looking position via the local least squares fitting just mentioned. The already fitted points are subsequently used to guide the local fitting of their neighboring points by being assigned with heavier weights, see Fig. 5.

The untraversed neighboring points of the current seed are then activated, each associated with a weight based on the irregularity of the current seed. This irregularity metric comes as a combination of the local normal/gradient variation and the local fitting error  $I = \alpha_1 \kappa + \alpha_2 \varepsilon$ . The next current point would go to the active point with the lowest irregularity.

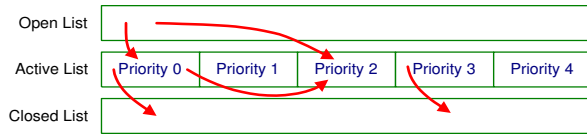


Figure 6: The data structure for the front propagation algorithm.

As in most region growing methods, our algorithm maintains three doubly linked lists of data points, the open list, the active list and the closed list. We further partition our active list into several prioritized segments. The sample points can only be moved downward and rightward (from higher to lower priority). The prioritized segmentation of the active list avoids a rigorous sorting of the points in the active list based on their irregularity.

This prioritized front propagation method as outlined in Fig. 7 always follows the easiest way of proceeding. As a result, the interspersed “irregular” areas will be first besieged before they are attacked. The accumulation of more fitted points in the local region around the “irregular” points can provide clues on what the local shape looks like.

Line 14–16 in Fig. 7 provides a seed jumping mechanism. When some part of the surface is confined by a barrier of irregular regions, this mechanism allows a new seed to be “airdropped” into that area.

```

1. OPEN  $\leftarrow$  all sample points
2. ACTIVE  $\leftarrow \emptyset$ ; CLOSED  $\leftarrow \emptyset$ 
3. WHILE ACTIVE  $\neq \emptyset$  OR OPEN  $\neq \emptyset$ 
4.   IF ACTIVE =  $\emptyset$ , ACTIVE  $\xleftarrow{pri}$  P  $\in$  OPEN of smaller Irr.
5.   CUR = first point in ACTIVE
6.   local fitting around CUR
7.   new Irr.(CUR) = Curvature(CUR) + FitError(CUR)
8.   calculate new Priority(CUR) based on new Irr.(CUR)
9.   IF new Priority(CUR) > old Priority(CUR)
10.    ACTIVE  $\xleftarrow{pri}$  CUR; GOTO 19 // back into active list
11.  create successors of CUR
12.  FOR each successor S
13.    Priority(S) = Priority(CUR); ACTIVE  $\xleftarrow{pri}$  S
14.    IF ACTIVE|priority  $\leq$  current priority =  $\emptyset$ 
15.      find T  $\in$  OPEN, such that Prior.(T)  $\leq$  Prior.(CUR)
16.      ACTIVE  $\xleftarrow{pri}$  T, if there is such a T
17.  END FOR 12.
18.  CLOSED  $\leftarrow$  CUR
19. END WHILE 3.

```

Figure 7: Prioritized front propagation algorithm.  $\xleftarrow{pri}$  means inserting into the end of a segment of the active list based on priority.

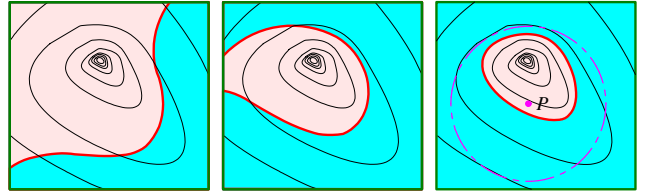


Figure 8: “Irregular” areas will first be besieged before they are attacked. The accumulation of already-fitted points around the “irregular” regions provides clues on how the local shape looks like.

## 6 Local Bandwidth and Model Selection

An active research topic in the field of local regression is on how to select a suitable smoothing window size (bandwidth) and on how to choose an optimal degree of the local polynomial (model) [Fan et al. 1996]. In Fig. 9(a), a local linear regression with different bandwidth is used for fitting a set of data points. We can see that the fitting curve with smaller bandwidth is much closer to the point set (we say it is of low bias), but suffers greatly from the undulation of noise (we say it is of high variance), larger window size is more likely to produce a smooth curve, but fails to grasp the sharp feature (peak). The same situation goes to local polynomial degree selection. In Fig. 9(b), polynomials with higher degrees are capable of capturing local features, but the shape may fluctuate undesirably.

Many sophisticated algorithms have been proposed to tradeoff bias with variance by introducing a certain metric. We refer readers to [Cleveland and Loader 1995] for a survey. One particular work in the computer graphics community was done by Lee [2000] to offer adaptive local bandwidth control in the process of curve smoothing.

### 6.1 Local bandwidth control

In this paper, we adopt a simple local bandwidth selection technique in which surface regions with larger irregularity tend to have larger fitting bandwidth since it is desirable to accommodate more already-fitted points. An initial feature size  $h_0$  is computed with an algorithm similar to the one used in Lee’s curve smoothing system. Each priority is associated with an empirical real number  $c_i$ . The local fitting bandwidth  $h$  is then computed as  $c_i h_0$ . This method

works well when the sampling density does not change drastically. More sophisticated bandwidth control algorithms borrowed from the statistics community will be further accommodated.

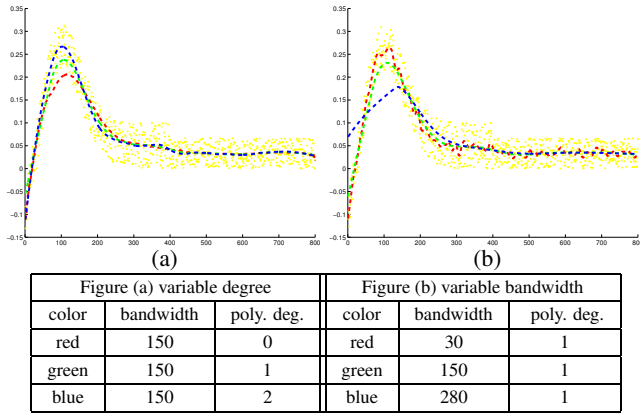


Figure 9: Selection of local polynomial degree and local fitting bandwidth.

## 6.2 Local model selection

Because the implicit quadrics are capable of representing multi-sheeted surfaces, extraneous parts may appear when using quadrics to fit simple shapes. For example, in Fig.10(a), fitting a set of planar point with implicit quadrics results in a shape close to two intersecting planes. Similar to high degree polynomials, implicit surface fitting is very sensitive to noise. In this section, we follow the spirit of local model selection, that is, selecting local fitting models by means of some error metric, to address these problems.

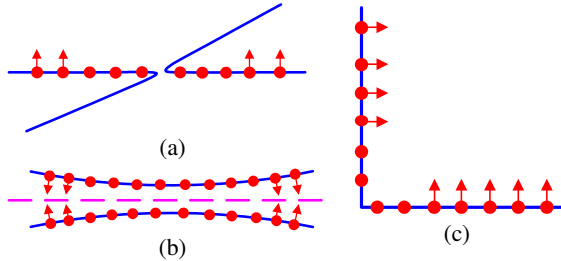


Figure 10: Local implicit quadrics fitting and model selection.

We first partition all types of quadric surfaces into two groups. Elliptic paraboloids, hyperbolic paraboloids, parabolic cylinders and planes (pairs of planes are not included) fall into the first group, and all other quadrics fall into the other group. Levin’s local least squares fitting method only uses models from the first group. Our new local fitting method will allow both groups of quadrics be utilized for local surface fitting.

Fitting with quadrics from the first group can follow Levin’s local least squares fitting as mentioned in section 4.1. A transformation of the function  $z - g(x, y) = 0$  in its local coordinate system back to the real world coordinate system is subsequently carried out. Now we get an locally fitted implicit quadric surface  $f_1(x, y, z) = 0$  from the first group.

Fitting with quadrics from the second group is by directly solving a linear system:

$$ax_i^2 + by_i^2 + cz_i^2 + dx_iy_i + ey_iz_i + fz_ix_i + gx_i + hy_i + iz_i + j = 0.$$

where  $i$  goes through all points  $p_i = (x_i, y_i, z_i)^\top$ . A non-trivial solution can be obtained by singular value decomposition(SVD).

Now, we have two locally fitted models  $f_1(x, y, z) = 0$  and  $f_2(x, y, z) = 0$  in hand. Our next step is to score the fidelity of these two models to the original data points. Basically, three metrics are used.

- *The local gradient variation*: Sharp features and multiple sheets in a local area will result in larger local gradient variation. This metric can be used to select models for cases like in Fig. 10(a). Obviously, a single-sheeted plane will be favored by this metric.
- *The fitting error*: Mostly, this is defined as the average squared distance of the sample points to the model. This metric can be applied to situations as in Fig. 10(c).
- *The normal orientation consistency*: Normal orientation is the most reliable metric for local model selection. In Fig 10(a), if two seeds propagate from both sides carrying normals as illustrated. The possibility of a multi-sheeted fitting as shown will certainly be ruled out. In Fig 10(b), with the normal information shown, the purple horizontal line will be ruled out.

These three metrics have ascending importance on the determination of local fitting model. A continuous real-valued fidelity metric can be designed based on the above analysis. Similar to the mixed degree local polynomial regression in [Cleveland and Loader 1995], a mixture of these two models with proper weights derived from this metric can be formulated.

## 7 Orientation Alignment

Traditionally, the normal orientation of each point is aligned by traversing a minimum spanning tree. But this approach may inflict difficulties when seeds go through areas of high curvature with high noise. For the case in Fig. 11(a), obviously, the most often used criteria by aligning neighbor normals into acute angles will lead to the normals in black. Even local curve fitting may not help, since the fitting curve for a point  $p$  and its neighbor may vary much due to fitting sensitivity in this kind of critical area (see green curve).

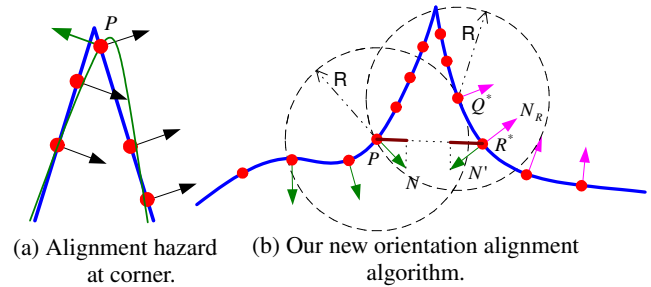


Figure 11: Normal orientation alignment.

In our system, our prioritized propagation rules guarantee a seed must not penetrate an acute edge. The problem becomes how to align orientation of patches where multiple seeds meet. As measurement precision, image registration accuracy and such all suffer from sharp corners and areas of high curvature, our top concern in orientation alignment algorithm is robustness. Under certain assumptions on feature size, we are able to design a robust algorithm.

Let us explain this algorithm with an example. In Fig. 11(b), a seed has reached this region from the right hand side and hibernates near  $Q^*$ . Another seed comes near and reaches  $P$ . At this moment, all points with arrows(normals) are in CLOSED list. Line 1–6 in Fig. 12 finds a nearest point  $Q^* \in \text{CLOSED}$  in its neighbor or radius  $R$  and on a different surface patch. We believe that for “opposite” points on the same surface, their normal would annihilate each other along the line linking them, i.e.  $(\text{Normal}(P_1) + \text{Normal}(P_2)) \perp (P_2 - P_1)$ . A local “opposite” point for  $P$  is searched for through Line 8–15. In the figure,  $N'$  is the reflection of  $N$  along line  $PR^*$ . This algorithm favors those points

R whose normals agree with  $N'(R)$  or  $-N'(R)$ . After the best  $R^*$  is found, the two patches are merged into one single patch. All point normals in one of the two patches will be flipped if  $N'(R^*)$  agree with  $-N(R^*)$ .

```

for the current seed P DO // P has been put in CLOSED
1.  $Nb_P \leftarrow$  all neighbor points in P's vicinity of radius  $R$ 
2.  $Q^* = \emptyset, \text{Dist}Q^* = \infty$ 
3. FOR each  $Q \in Nb_P$  DO // find a nearest "alien"
4.   IF  $Q \in \text{CLOSED}$  AND  $Q$  on a different patch from P
5.     IF  $\text{Dist}(P,Q) < \text{Dist}Q^*, Q^*=Q$  and  $\text{Dist}Q^* = \text{Dist}(P,Q)$ 
6.   END FOR 2.
7. IF  $Q^* = \emptyset$  GOTO 19.
8.  $Nb_Q \leftarrow$  all neighbor points in  $Q^*$ 's vicinity of radius  $R$ 
9.  $R^* = \emptyset, \text{Angle}R^* = 0$ 
10. FOR each  $R \in Nb_Q$  DO // find the best opposite point in  $Nb_Q$ 
11.   IF  $R \in \text{CLOSED}$  AND  $R$  on the same patch as  $Q^*$ 
12.      $N' = \text{Reflect}(\text{Normal of } P), N_R = \text{Normal of } R^*$ 
13.     IF  $|\langle N', N_R \rangle| \geq |\text{Angle}R^*|$ 
14.        $\text{Angle}R^* = \langle N', N_R \rangle$  and  $R^* = R$ 
15.   END FOR 10
16. IF  $|\text{Angle}R^*| < \sqrt{3}/2$  GOTO 19
17. IF  $\text{Angle}R^* < 0$  flip all normals in the patch containing  $R^*$ 
18. merge the two patches contain P and  $R^*$ 
19. end procedure

```

Figure 12: Normal orientation alignment algorithm.

## 8 Numerical Techniques

In our work, many geometric properties on the local fitting quadrics such as local curvature, fitting error, etc. are required to be calculated accurately and efficiently. In this section, we introduce some algorithms we developed for local surface property evaluation.

### 8.1 Curvature at a surface point

The curvature of an implicit surface at a surface point  $\mathbf{p}$  can be evaluated efficiently without introducing a local parametrization [Xie and Qin 2002].

Let  $\mathbf{M} = \left\{ \frac{\partial^2 f}{\partial x_i \partial x_j} \right\}_{i,j=0,1,2}$  be the Hessian matrix of  $f$ . Denote  $\mathbf{M}^* = \det(\mathbf{M}) \mathbf{M}^{-1}$ , which is sometimes called adjoint matrix of  $\mathbf{M}$ ,  $\mathbf{H} = \mathbf{M} - \text{tr}(\mathbf{M}) \mathbf{I}$  and  $\mathbf{g} = \nabla f(\mathbf{p})$ , we have,

$$\text{Gaussian curvature: } K(\mathbf{p}) = \|\mathbf{g}\|^{-4} \mathbf{g}^\top \cdot \mathbf{M}^* \cdot \mathbf{g} \quad (5)$$

$$\text{Mean curvature: } H(\mathbf{p}) = \frac{1}{2} \|\mathbf{g}\|^{-3} \mathbf{g}^\top \cdot \mathbf{H} \cdot \mathbf{g} \quad (6)$$

For a quadratic polynomial  $f$ , entries of  $\mathbf{M}$  are just the coefficients of its second degree terms. The total curvature at  $\mathbf{p}$  is defined as  $T = 4H^2 - 2K$ .

### 8.2 Variation of normal and gradient

The normal variation near a point  $\mathbf{p}$  within a neighborhood of radius  $R$  is important for sharp feature detection and local model selection. Using the curvature at this point is not always a good estimation of the local normal change. In Fig. 13(a), the curvature at point  $\mathbf{p}$  vanishes, but a sharp edge of infinite curvature is nearby. Ideally, the local normal variation for a surface point  $\mathbf{p}$  should be defined as the maximum angle spanned by any two normals in its vicinity of a given radius  $R$ , as shown in Fig. 13(a). This definition can also capture the cases when two surfaces come close together as shown in Fig. 13(b). But because normal is not a linear function of the

implicit function  $f$ , evaluation of normal change can be expensive, let alone the calculation of the maximum normal angle spanned in a region.

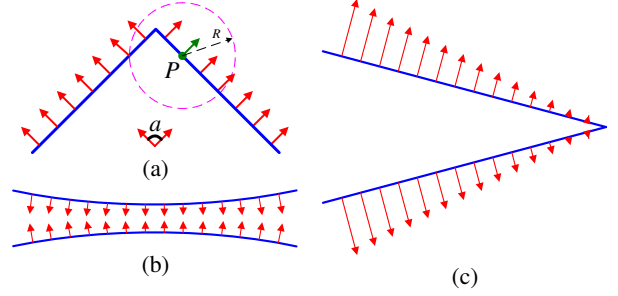


Figure 13: Normal (gradient) variation near a point.

An alternative approach is to measure the local gradient change instead of normal change. Prior to our derivation, we define:

$$\begin{aligned} V_{gx} &= \max(2|a|, |d|, |f|), \\ V_{gy} &= \max(2|b|, |d|, |e|), \\ V_{gz} &= \max(2|c|, |e|, |f|). \end{aligned} \quad (7)$$

$$\text{Var} = \|(V_{gx}, V_{gy}, V_{gz})^\top\| \quad (8)$$

Now consider a quadric surface

$$q = ax^2 + by^2 + cz^2 + dxy + eyz + fzx + gx + hy + iz + j,$$

the gradient at a point  $\mathbf{p} = (x, y, z)^\top$  is,

$$\mathbf{g} = (g, h, i)^\top + (2ax + dy + fz, 2by + dx + ez, 2cz + ey + fx)^\top.$$

The variable part, that is, the second term, can only vary within a cuboid

$$[-RV_{gx}, RV_{gx}] \times [-RV_{gy}, RV_{gy}] \times [-RV_{gz}, RV_{gz}], \quad (9)$$

where,  $R$  is the radius of the neighborhood around  $\mathbf{p}$ . Note that (8) is constant with respect to  $\mathbf{p}$  and invariant under translation of  $f$ <sup>1</sup>. By translating the origin of coordinate system to  $\mathbf{p}$ , we can define the gradient variance in  $\mathbf{p}$ 's vicinity of radius  $R$  as:

$$V = \frac{R \text{Var}}{\|\nabla f(\mathbf{p})\|} \quad (10)$$

## 9 Experiments

In Fig.1 on the front page, we show the feature detecting capability embedded in our prioritized incremental surface construction algorithm. At the final stage of the shape growing, the remaining active points (in green) outline the dragon as its feature lines. In Fig. 14, we show the segmentation power of our incremental surface construction algorithm. In Fig. 15, we show the normal propagation algorithm. We can also see that with implicit quadrics, the normals take abrupt changes at sharp edges as we expected. But quadrics have a drawback that it can not express corners well where more than two faces meet. This remains one of our future work. In Fig. 16, we show the denoising process of our algorithm. Since the noise rate is very high (about one fiftieth of the size of the object), surface details are wiped out. But at the tail, which is a cusp-like shape, our algorithm presents better results than many other point-based denoising methods. Fig. 17 shows our work for accurate models. We see that the Shepard's method works well even for low sampling rate point sets.

<sup>1</sup>A mathematically accurate version of  $\text{Var}$  is defined as the largest eigenvalue of the Hessian matrix of  $f$ , which is invariant under translation and rotation.

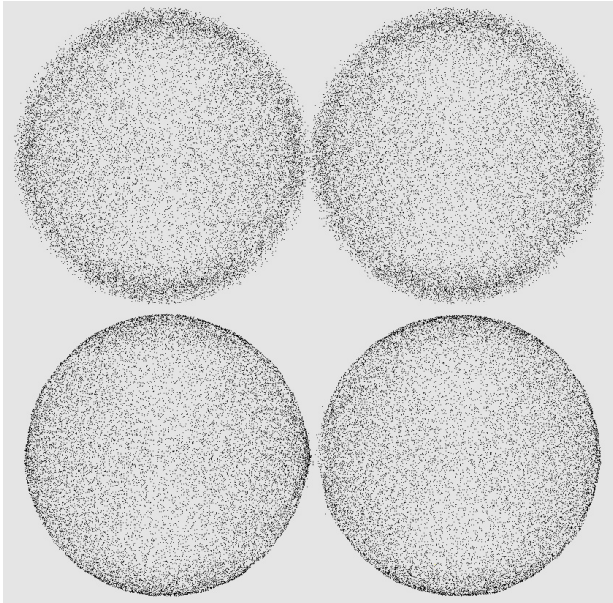


Figure 14: Ambiguous topology determination. In this figure, we show the segmentation power of our incremental surface growing algorithm. The sample points of the top two spheres are mixed together due to noise. The bottom two spheres are thinned and separated into two groups. Due to high sensitivity of the model (two-sheeted quadrics) to local noise, we still see several points between them. This needs a further process to discard outliers.

## 10 Conclusions and Future Work

In this paper, we have presented a new intelligent incremental surface reconstruction system based on local fitting and blending of implicit quadric surfaces. The incorporation of implicit quadrics as basic local fitting models allows us to reconstruct local features more accurately and to blend local fitted surface patches in a clean and efficient manner. To deal with noisy data set and topological ambiguity, we design a prioritized incremental algorithm, in which, flatter areas are fitted prior to areas with features. The already fitted areas can subsequently guide the local fitting of its neighboring points. We also implement a method for automatic selection of local bandwidth and local fitting models. Finally, we implement a robust orientation alignment method as well as a set of algorithms on efficient evaluation of local geometric information. Our system has displayed the capability and flexibility of our novel algorithms. Some potential applications include feature line extraction, segmentation of close objects, etc.

Despite all the merits that our algorithms have displayed, they still suffer from some limitations. The quadric surfaces can not reconstruct the shapes well at sharp corners where more than two faces meet or at places where more than two objects come close. Extension of this method to a more flexible function set will definitely enhance its capability. Our system still has some empirical parameters. Deep study of the regularity metric is also an open issue.

## Acknowledgments

This research is supported in part by the NSF CAREER award CCR-9896123, the NSF grant DMI-9896170, the NSF ITR grant IIS-0082035, the NSF grant IIS-0097646, the NSF grant IIS-0326388, the NSF grant ACI-0328930, the NSF grant CCR-0306438, Alfred P. Sloan Fellowship and Honda Initiation Award.

## References

- ALEXA, M., BEHR, J., COHEN-OR, D., FLEISHMAN, S., LEVIN, D., AND SILVA, C. T. 2001. Point set surfaces. *Proceedings of the IEEE Visualization 2001* (October), 21–28.
- 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.
- BERNARDINI, F., MITTLEMAN, J., RUSHMEIER, H., SILVA, C., AND TAUBIN, G. 1999. The ball-pivoting algorithm for surface reconstruction. *IEEE Transactions on Visualization and Computer Graphics* 5, 4, 349–359.
- 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*, 67–76.
- CLEVELAND, W., AND LOADER, C. 1995. Smoothing by local regression: Principles and methods. Technical Report, AT&T Bell Laboratories, Murray Hill, NY.
- 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.
- EDELSBRUNNER, H., AND MCKE, E. P. 1994. Three-dimensional alpha shapes. *ACM Transactions on Graphics (TOG)* 13, 1, 43–72.
- FAN, J., GIJBELS, I., HU, T.-C., AND HUANG, L.-S. 1996. A study of variable bandwidth selection for local polynomial regression. *Statistica Sinica* 6, 113–127.
- FRANKE, R., AND NIELSEN, G. 1980. Smooth interpolation for large sets of scattered data. *Int. Journal for Numerical Methods in Engineering* 15, 1691–1704.
- 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.
- LANCASTER, P., AND SALKASKAS, K. 1981. Surfaces generated by moving least squares methods. *Math. of Comput.* 37, 155, 141–158.
- LEE, I.-K. 2000. Curve reconstruction from unorganized points. *Computer Aided Geometric Design* 17, 2, 161–177.
- LEVIN, D. 1998. The approximation power of moving least-squares. *Mathematics of Computation* 67, 224, 1517–1531.
- LEVIN, D. 1999. Mesh-independent surface interpolation. *4th International Conference on Curves and Surfaces*, 46–62.
- MCLAIN, D. H. 1974. Drawing contours from arbitrary data points. *The Computer Journal* 17, 4, 318–324.
- OHTAKE, Y., BELYAEV, A., ALEXA, M., TURK, G., AND SEIDEL, H.-P. 2003. Multi-level partition of unity implicits. *Computer Graphics Proceedings, ACM SIGGRAPH Annual Conference Series*. To appear.
- RENKA, R. J. 1988. Multivariate interpolation of large sets of scattered data. *ACM Transactions on Mathematical Software (TOMS)* 14, 2, 139–148.
- SHEPARD, D. 1968. A two-dimensional interpolation function for irregularly-spaced data. *Proceedings of the 1968 23rd ACM national conference*, 517–524.
- XIE, H., AND QIN, H. 2002. A physics-based framework for subdivision surface design with automatic rules control. *10th Pacific Conference on Computer Graphics and Applications*, 304–315.
- ZHAO, H.-K., OSHER, S., MERRIMAN, B., AND KANG, M. 2000. Implicit and nonparametric shape reconstruction from unorganized data using a variational level set method. *Computer Vision and Image Understanding* 80, 3, 295–314.

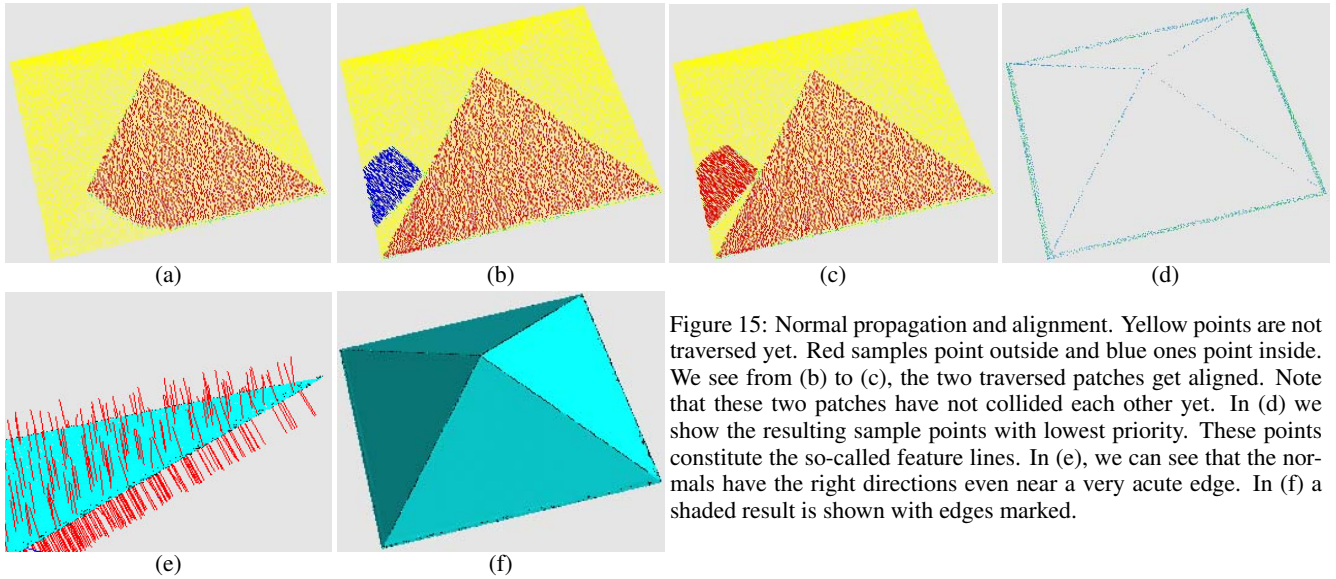


Figure 15: Normal propagation and alignment. Yellow points are not traversed yet. Red samples point outside and blue ones point inside. We see from (b) to (c), the two traversed patches get aligned. Note that these two patches have not collided each other yet. In (d) we show the resulting sample points with lowest priority. These points constitute the so-called feature lines. In (e), we can see that the normals have the right directions even near a very acute edge. In (f) a shaded result is shown with edges marked.

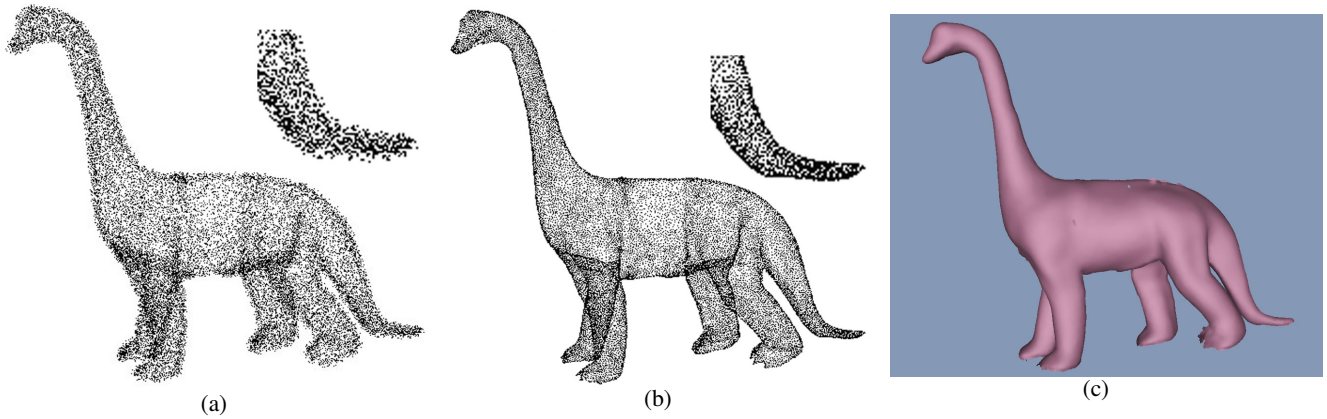


Figure 16: Denoising functionality of our algorithm. In (a) a dinosaur is sampled noisily (note that the dinosaur has a relatively cusped tail with noise rate close to its feature size). In (b) we show the resulting thinned point cloud. In (c), the thinned point cloud is rendered by PointShop (by Zwicker, Pauly et al.).

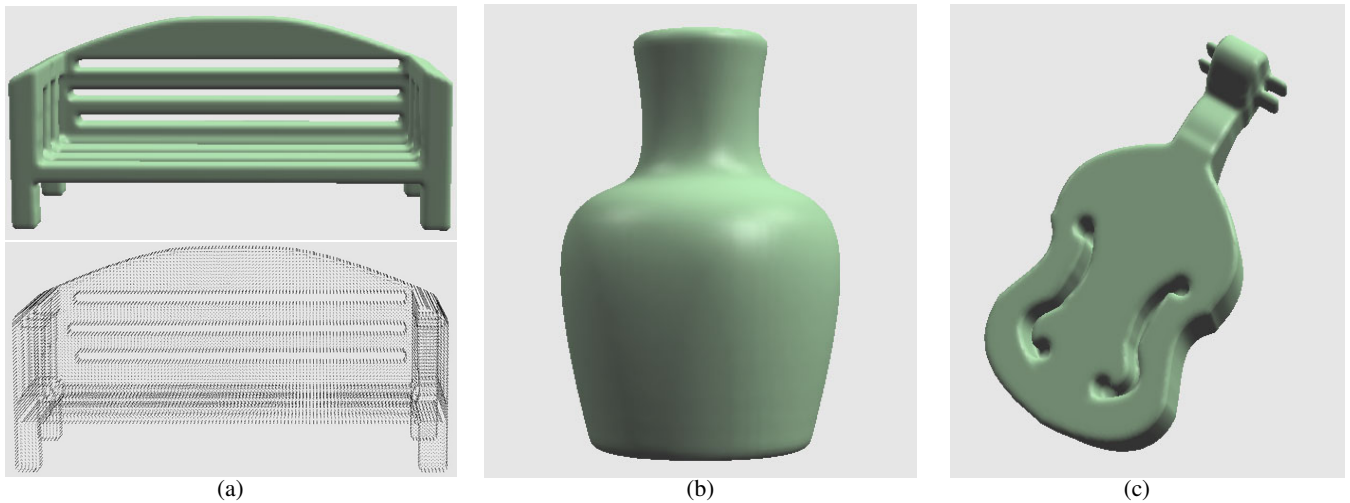


Figure 17: Some models from accurate point clouds. The sampling rates for these models are lower than the previous models. The Shepard's method works well for blending implicit fields generated at each sample point together. All the resulting models are rendered in polygonal meshes extracted by marching-cubes algorithm.