

Surface Reconstruction Based on the Modified Gauss Formula

WENJIA LU, ZUOQIANG SHI, JIAN SUN, and BIN WANG, Tsinghua University, China

In this article, we introduce a surface reconstruction method that has excellent performance despite nonuniformly distributed, noisy, and sparse data. We reconstruct the surface by estimating an implicit function and then obtain a triangle mesh by extracting an iso-surface. Our implicit function takes advantage of both the indicator function and the signed distance function. The implicit function is dominated by the indicator function at the regions away from the surface and is approximated (up to scaling) by the signed distance function near the surface. On one hand, the implicit function is well defined over the entire space for the extracted iso-surface to remain near the underlying true surface. On the other hand, a smooth iso-surface can be extracted using the marching cubes algorithm with simple linear interpolations due to the properties of the signed distance function. Moreover, our implicit function can be estimated directly from an explicit integral formula without solving any linear system. An approach called *disk integration* is also incorporated to improve the accuracy of the implicit function. Our method can be parallelized with small overhead and shows compelling performance in a GPU version by implementing this direct and simple approach. We apply our method to synthetic and real-world scanned data to demonstrate the accuracy, noise resilience, and efficiency of this method. The performance of the proposed method is also compared with several state-of-the-art methods.

CCS Concepts: • Theory of computation → Computational geometry; • Computing methodologies → Mesh models; Point-based models;

Additional Key Words and Phrases: Surface reconstruction, point cloud, Gauss lemma, disk integration

ACM Reference format:

Wenjia Lu, Zuoqiang Shi, Jian Sun, and Bin Wang. 2018. Surface Reconstruction Based on the Modified Gauss Formula. *ACM Trans. Graph.* 38, 1, Article 2 (December 2018), 18 pages.

<https://doi.org/10.1145/3233984>

1 INTRODUCTION

Surface reconstruction is a classic problem that has been studied for more than 3 decades. Although many elegant methods are available, we focus on implicit methods. The indicator function is a commonly used choice of implicit function (Kazhdan et al. 2006; Kazhdan and Hoppe 2013). However, the indicator function is discontinuous on the surface. A smoothing filter, such as in Poisson

This project was partially funded by National Science Foundation of China (61772301, 11371220, 11671005).

Authors' addresses: W. Lu, J. Sun, and B. Wang (corresponding author), Room 10-403, East Main Building, Tsinghua University, Beijing, 100084, China; emails: luwenjiaguangyao@163.com, sunjian0813@gmail.com, wangbins@tsinghua.edu.cn; Z. Shi, 259, Jin Chun Yuan West Bldg, Tsinghua University, Beijing, 100084, China; email: zqshi@tsinghua.edu.cn.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from Permissions@acm.org.

© 2018 Association for Computing Machinery.

0730-0301/2018/12-ART2 \$15.00

<https://doi.org/10.1145/3233984>

reconstruction (Kazhdan et al. 2006; Kazhdan and Hoppe 2013), should be applied to the indicator function to obtain a good surface normals approximation; otherwise, the recovered surface may not be smooth, as shown in Figure 2. However, smoothing filter methods frequently experience difficulty controlling approximation error and thus tend to overfit or oversmooth the data. Another widely used function in surface reconstruction is the signed distance function (Curless and Levoy 1996; Hoppe et al. 1992). Unlike the indicator function, a signed distance function is smooth near the surface—this makes extracting a smooth watertight surface from its zero level-set easier compared with other methods. However, the signed distance function is difficult to compute in the area away from the surface. Moreover, the resulting methods are sensitive to noise and may generate spurious surface sheets. The construction of a hybrid function is a typical approach solving these problems. This way, the indicator function dominates the regions away from the surface and the signed distance function controls the near-surface parts.

In this article, an explicit integral formula is proposed for constructing such a function based on the Gauss lemma in the potential theory (e.g., Wendland (2009)). The Gauss lemma provides an integral formula for the indicator function. In this formula, the Gauss lemma is modified to provide an approximation for the signed distance function near the surface while keeping the indicator function away from the surface intact. Our implicit function can be directly estimated from this integral formula without solving any linear system, such as in Poisson reconstruction or minimizing an energy function such as in Calakli and Taubin (2011).

In our integral formula, the integrand is near singular at the sample points and global over the entire computational domain, which introduces some difficulties in accurately and efficiently evaluating the integral. To overcome the singularity of the integrand, we introduce a method called *disk integration* to compute the integral near the singularity. However, the globality of the integral formula causes the algorithm to be relatively slow. To address this issue, the fast multipole method (FMM) (Greengard and Rokhlin 1987) is used to speed up the computation. By integrating all these pieces, explicit integral formula, disk integration, and FMM, an accurate, efficient, and robust surface reconstruction method is obtained.

In summary, our contributions are as follows:

- A hybrid implicit function combining an off-surface indicator function and a near-surface signed distance function is proposed. This has many beneficial properties, such as resilience to poor-quality data, less spurious surface sheets away from the input samples, and an easy way to recover the iso-surface. Moreover, the proposed method achieves high accuracy by avoiding the smoothing operation.
- An explicit integral formula based on Gauss lemma is applied to estimate the implicit function, which has a natural parallel implementation and an almost negligible overhead.

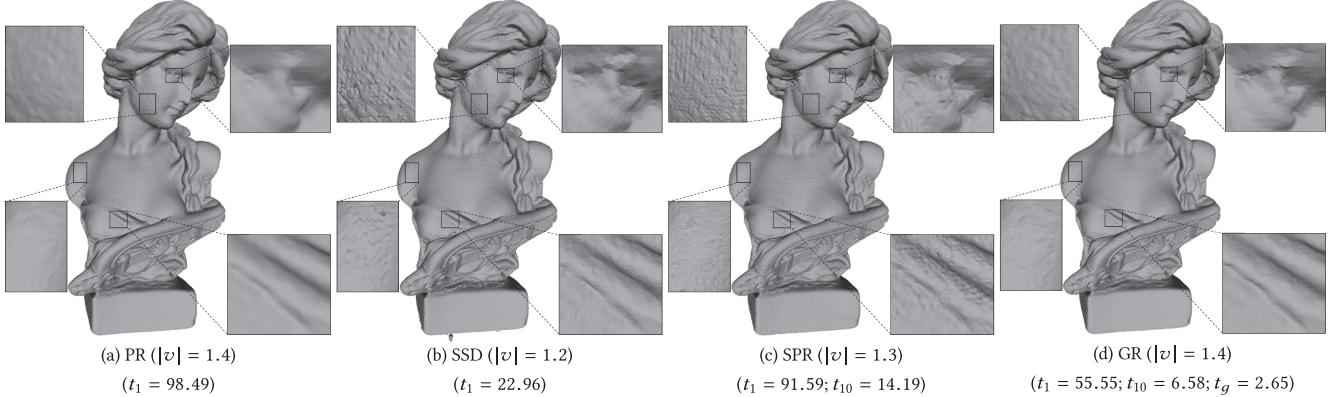


Fig. 1. Reconstructions of the Lady model by Poisson Reconstruction (PR) (Kazhdan et al. 2006), Smoothed Signed Distance Reconstruction (SSD) (Calakli and Taubin 2011), Screened Poisson Reconstruction (SPR) (Kazhdan and Hoppe 2013), and our method (GR). The Lady model is a real-world scanned data with 0.5 million samples. $|v|$ denotes the number of vertices in millions of the reconstructed mesh, and t_1 , t_{10} , and t_g denote the running time in seconds of the reconstructions with single thread, 10 threads, and GPU, respectively.

- A novel approach called *disk integration* is proposed to address the singularity issue and improve the accuracy of reconstruction.
- A concise and efficient version of FMM is proposed to improve efficiency.

Figure 1 compares our Gauss reconstruction with several state-of-the-art methods when applied to the real-world scanned Lady model data. All reconstructions are computed by using an octree with a maximum depth of 10. From Figure 1, we can see that our method generates a high-quality reconstruction of the Lady model, preserving the details, and avoiding overfitting of the input samples. The parallel implementation of our method practically achieved a perfect linear speed-up, whereas with the GPU implementation, our method significantly outperformed several other methods in terms of speed.

2 RELATED WORKS

2.1 Surface Reconstruction

Surface reconstruction from point cloud has attracted much attention in theory and practice in the past 30 years. Many related algorithms have been proposed. We provide a brief review of the algorithms relevant to our work. There are two main categories: combinatorial algorithms and implicit algorithms.

Combinatorial methods utilize part of the input sample points as vertices and reconstruct output meshes by determining the connectivity of the input samples. Many of these methods are based on the Voronoi diagram or its dual Delaunay triangulation, including Crust (Amenta et al. 1998), Power Crust (Amenta et al. 2001), Cocone (Amenta et al. 2002), Robust Cocone (Dey and Goswami 2004), Wrap (Edelsbrunner 2003), and flow complex (Giesen and John 2008). These methods have important theoretical properties. However, they are usually sensitive to noise and may produce jagged surfaces. In Kolluri et al. (2004), a spectral-based approach is proposed to smooth the surface. In Xiong et al. (2014), a learning approach is proposed to treat geometry and connectivity

reconstruction as one joint optimization problem to improve reconstruction quality.

Implicit methods estimate implicit functions from the input samples and extract iso-surfaces to generate triangle meshes. Poisson reconstruction and its variant (Kazhdan et al. 2006; Kazhdan and Hoppe 2013) are the most relevant methods to our work, which estimate the indicator functions of unknown models. In Muraki (1991) and Walder et al. (2005), Radial Basis Functions (RBFs) are used as bases for defining implicit functions, where coefficients of bases are determined by fitting the input data. As RBFs are global in the above-mentioned works, applying FMM is a common way to improve the efficiency (Carr et al. 2001). The signed distance function is a natural choice as an implicit function for surface reconstruction, where the implicit function can be estimated either locally as distances to tangent planes of nearby samples (Curless and Levoy 1996; Hoppe et al. 1992) or globally by minimizing the fitting error (Calakli and Taubin 2011). In Amenta and Kil (2004a), Dey and Sun (2005), Levin (2004), and Shen et al. (2004), moving least squares (MLS) is used to define implicit surfaces, which are extremal sets of a certain energy. MLS is associated with a nice projection operator that can be used for surface smoothing. Compared with our method, the implicit function in traditional MLS is often only meaningful near the surface, and thus the reconstruction of MLS may generate spurious surface sheets away from the surface. For this reason, Amenta and Kil (2004b) attempted to widen the domain without producing spurious sheets. Furthermore, in Amenta and Kil (2004b) and Özti̇reli et al. (2009), methods based on MLS are extended to recover sharp features. Finally, Fuhrmann and Gosele (2014) defined the implicit function as the sum of compactly supported basis functions. The implicit function performed well on large, redundant, and potentially noisy datasets by leveraging the extra scale information provided with the input. The surfaces reconstructed by implicit methods frequently do not interpolate the input samples. Therefore, they are smoother than those reconstructed by combinatorial methods.

As stated in Berger et al. (2017), the surface normal plays an important role in surface reconstruction. However, surface normals

are difficult to obtain when certain information is not present. Therefore, surface reconstruction based on an unoriented point cloud has also drawn attention. Chen et al. (2013) computed the higher-order local approximations of the nonoriented input gradients based on an MLS formulation. In Alliez et al. (2007), a Voronoi-PCA estimation is performed, which resulted in a tensor field encoding normal information. The implicit function is then computed to recover the surface. To process the unoriented data in our approach, we estimate the normals as a pre-process using the standard method of combining PCA with a consistent operation. Experiments show that our method is resilient to point normals. The accurate estimation of normals is not required.

2.2 Implicit Function Evaluation

Evaluating the implicit function varies from one method to another. Many of the methods are based on solving the Laplace equation. For example, Kazhdan et al. (2006) and Kazhdan and Hoppe (2013) turned it into a linear system discretized using the B-spline basis and solved the sparse linear system to obtain a solution with explicitly defined Dirichlet/Neumann boundary constraints. The boundary element method (BEM), as applied in our method, is another tool for evaluating the solution to the Laplace equation. This is also widely used in different areas, such as mesh segmentation (Jacobson et al. 2013). One advantage of the BEM-based solution is that no boundary conditions are required to be explicitly imposed.

Inspired by the BEM, Jacobson et al. (2013) defined a function by generalizing the winding number to guide a graph cut segmentation of a constrained Delaunay tessellation (CDT). Benefiting from its good properties, this method can segment the inside and outside parts of the input model perfectly. In addition, this method is robust to many poor boundary conditions, such as self-intersection, nonmanifold pieces, and open boundaries. Although our method also applies the BEM to solve the Laplace equation, the two methods have different applications. Jacobson et al. (2013) built tetrahedral meshes for inside spaces bounded by polygonal soups. In contrast, our method reconstructed surfaces from points with normals. Moreover, the differences between the two methods are listed in the following.

- **Kernel Function.** In Jacobson et al. (2013), the original BEM kernel was used without modification and the segmented mesh was used by applying graph cut. However, in our method, the fundamental solution (Equation (4)) is modified to obtain a good approximation of surface normals. Thus, the resultant mesh is generated by a linear interpolation near the surface.
- **Speed-up Method.** A simple calculation method is unsuitable because the BEM is a global method. In Jacobson et al. (2013), a hierarchical method is applied to improve the efficiency, which led to an exact evaluation of the function value. However, the input scale for our method is larger, and this method does not work efficiently in our condition. Moreover, we only need to evaluate the accurate value near the surface for interpolation. This leads to FMM only being applied for high efficiency with acceptable errors.

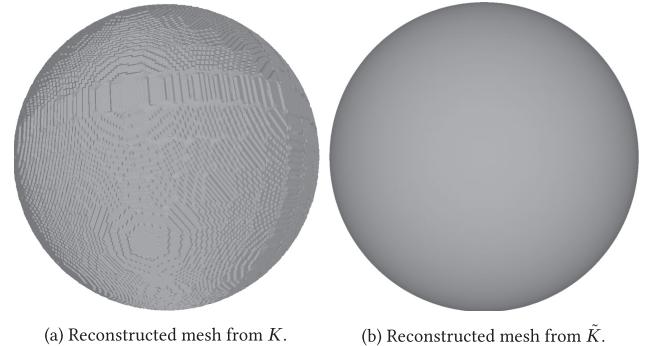


Fig. 2. Reconstructed sphere by different kernels.

- **Near-Singularity Issue.** In performing integration, the kernel function of our method may suffer from the near-singularity problem, which did not occur in Jacobson et al. (2013). To address this issue and make the function evaluation more accurate near the surface, the *disk integration* method is proposed.

2.3 Iso-surface Extraction

For the iso-surface extraction, marching cubes (Lorensen and Cline 1987) and its adaptation to octree (Wilhelms and Van Gelder 1992) are commonly used methods. Many efficient variants or extensions have been proposed. Primal MC methods, such as the method proposed by Kazhdan et al. (2007) extracted a watertight mesh by means of edge trees where the positions of the iso-value-crossings are defined. Schaefer and Warren (2004) extracted the iso-surface, including sharp features, by aligning dual grid vertices with implicit function features. Delaunay refinement-based methods (Boissonnat and Oudot 2005) produced good quality triangle meshes. However, they were less efficient and difficult to parallelize.

3 ALGORITHM

Given a set of points \mathcal{P} and the corresponding normal field \vec{N} , sampling the boundary $\partial\Sigma$ of an unknown region $\Sigma \subset \mathbb{R}^3$, that is, $p \in \mathcal{P}$ lies on or near the surface, and $\vec{N}(p)$ approximates the surface normal near the position p . Our goal is to reconstruct a triangle mesh approximating the boundary $\partial\Sigma$.

3.1 Reconstruction Function

In this section, an explicit integral formula is introduced to estimate the implicit reconstruction function.

First, we recall that the indicator function χ of the region Σ has an explicit integral formula, which is given in the Gauss lemma in potential theory (Wendland 2009).

LEMMA 3.1 (GAUSS LEMMA). *Let Σ be an open region in \mathbb{R}^3 , and $\bar{\Sigma}$ denotes the closure of Σ . Consider the following double layer potential: for any $x \in \mathbb{R}^3, y \in \partial\Sigma$*

$$\chi(x) = \int_{\partial\Sigma} \frac{\partial G}{\partial \vec{N}(y)}(x, y) d\tau(y), \quad (1)$$

where $\vec{N}(y)$ denotes the outward normal of $\partial\Sigma$ at y , $d\tau(y)$ is the surface area form of $\partial\Sigma$ at y , and G is the fundamental solution of the

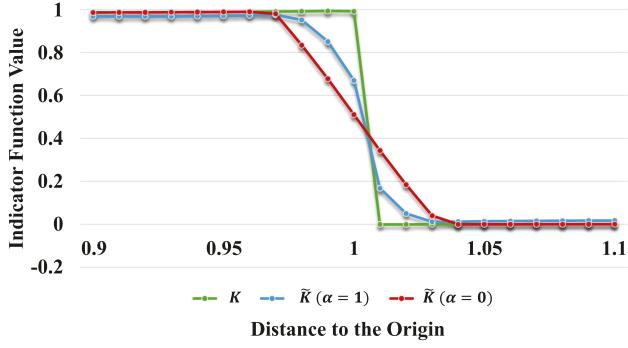


Fig. 3. The illustration of the indicator function around the north pole restricted to the diameter passing the north pole.

Laplace equation in \mathbb{R}^3 , which can be stated explicitly as:

$$G(x, y) = -\frac{1}{4\pi \|x - y\|}. \quad (2)$$

Then, $\chi(x)$ is the indicator function of Σ , i.e.

$$\chi(x) = \begin{cases} 0 & x \in \mathbb{R}^3 \setminus \bar{\Sigma} \\ 1/2 & x \in \partial\bar{\Sigma} \\ 1 & x \in \Sigma \end{cases} \quad (3)$$

Note that

$$\frac{\partial G}{\partial \vec{N}(y)}(x, y) = -\frac{1}{4\pi} \frac{(x - y) \cdot \vec{N}(y)}{\|x - y\|^3},$$

which we call the *kernel function*, and denote by $K(x, y)$.

The integral formula (1) exhibits many good properties. However, two issues require addressing:

- Given the indicator function χ , the resultant triangle mesh by iso-surfacing χ , denoted by M , resides in a small tubular neighborhood of the surface $\partial\Sigma$. Namely, the Hausdorff distance between M and $\partial\Sigma$ is small. However, given that the function χ is discontinuous at $\partial\Sigma$, the normal of a triangle in M may not approximate the normals of $\partial\Sigma$ at points close to the triangle, as shown in Figures 2(a) and 3.
- The *kernel function* $K(x, y)$ becomes singular as x approaches y . To accurately evaluate the indicator function χ at points close to the surface $\partial\Sigma$, a very dense sampling of the surface is required, which is impractical.

To address the two issues, our approach is to modify the indicator *kernel function* K . We associate points in \mathbb{R}^3 with widths $w \in \mathbb{R}$ and modify the *kernel function* $K(x, y)$ for any $y \in \partial\Sigma$ as:

$$\tilde{K}(x, y) = \begin{cases} K(x, y), & \|x - y\| \geq w(x), \\ -\alpha \frac{(x - y) \cdot \vec{N}(y)}{4\pi \cdot w^3(x)} & \|x - y\| < w(x), \end{cases} \quad (4)$$

where $\alpha \in [0, 1]$ is a parameter. Note that $\tilde{K}(x, y)$ becomes discontinuous for $\alpha \neq 1$.

The reconstruction function can be stated as:

$$\tilde{\chi}(x) = \int_{\partial\Sigma} \tilde{K}(x, y) d\tau(y). \quad (5)$$

Note that $\tilde{K}(x, y)$ remains the same as $K(x, y)$ for any $y \in \partial\Sigma$ with $\|x - y\| \geq w(x)$, and hence $\tilde{\chi}(x) = \chi(x)$ for any x with

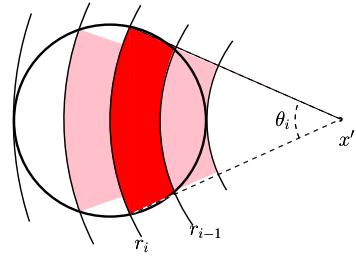


Fig. 4. Illustration of integral domain (shaded region) of disk integration.

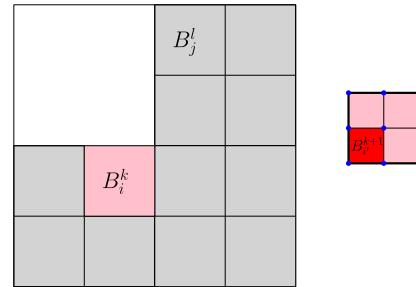


Fig. 5. The cubes of O at depth k may not cover the entire domain due to the adaptivity of O . The red cube B_i^{k+1} is a subcube of the pink cube B_i^k . The blue dots in B_i^k form set of grid vertices V_i^k in B_i^k .

$|d(x)| \geq w(x)$, where $d(x)$ denotes the signed distance from x to $\partial\Sigma$.

At a point x with $d(x) < w(x)$, $B(x, r)$ is the ball in \mathbb{R}^3 centered at x and of radius r . Notice that $w(x)$ is constantly a small number, whose choice is described in Section 3.4. This condition indicates $B(x, w(x)) \cap \partial\Sigma$ is a small piece of $\partial\Sigma$. Under the assumption that the surface Σ is smooth, $B(x, w(x)) \cap \partial\Sigma$ can be well approximated by a disk.

This approximation provides

$$\tilde{\chi}(x) = \frac{1}{2} + \left(\frac{1}{2} - \frac{\alpha}{4}\right) \frac{d(x)}{w(x)} + \frac{\alpha d^3(x)}{4w^3(x)} + O(|w(x)|). \quad (6)$$

The calculation of $\tilde{\chi}(x)$ can be found in the appendix.

When $\alpha = 1$: as Figure 3 shows, when $d(x)$ is small, that is, point x is close to the surface, $\tilde{\chi}(x)$ is dominated by a linear function of the signed distance $d(x)$. When $\alpha = 0$: as Figure 3 shows, $\tilde{\chi}(x)$ is strictly a signed distance function near the surface, which is desirable for extracting iso-surface (Calakli and Taubin 2011). In this case, only the points far away from x are considered so that it may make holes when extracting the iso-surface. However, we empirically set $\alpha = 0$ because most of the examples are made from dense inputs with complex shapes and noise and the resulting surface is extracted as a relatively coarse mesh.

The integral formula (5) is straightforward and simple. Note that the estimation of the implicit function $\tilde{\chi}$ at different points is independent, which leads to a natural parallel algorithm.

3.2 Disk Integration

Although the kernel function $\tilde{K}(x, y)$ is not singular, near-singularity problems may still exist when x, y are very close due

ALGORITHM 1: FMM($B_i^{d'}, B_j^{d''}, f: \mathcal{V} \rightarrow \mathbb{R}$)

```

if  $\|\bar{p}_i^{d'} - \bar{v}_j^{d''}\| \geq c\alpha_k$  then
    evaluate  $e \approx C(\bar{v}_j^{d''}, \bar{p}_i^{d'})$ ;
     $f(v) = f(v) + e$  for any  $v \in \mathcal{V}_j^{d''}$ ;
else
    if both  $B_i^{d'}$  and  $B_j^{d''}$  are leaves then
        forall the  $p \in \mathcal{P}_i^{d'}$  and  $v \in \mathcal{V}_j^{d''}$  do
            evaluate  $e \approx C(v, p)$ ;
             $f(v) = f(v) + e$ ;
        end
    else
        if Neither  $B_i^{d'}$  nor  $B_j^{d''}$  is a leaf then
            forall the  $B_{i'}^{d'+1} \subset B_i^{d'}$  and  $B_{j'}^{d''+1} \subset B_j^{d''}$  do
                FMM( $B_{i'}^{d'+1}, B_{j'}^{d''+1}, f$ );
            end
        else
            if  $B_i^{d'}$  is a leaf and  $B_j^{d''}$  is not a leaf then
                forall the  $B_{j'}^{d''+1} \subset B_j^{d''}$  do
                    FMM( $B_i^{d'}, B_{j'}^{d''+1}, f$ );
                end
            else
                forall the  $B_{i'}^{d'+1} \subset B_i^{d'}$  do
                    FMM( $B_{i'}^{d'+1}, B_j^{d''}, f$ );
                end
            end
        end
    end

```

to the specification of the small-width coefficient. What's more, we are interested in estimating function values for the points near surface $\partial\Sigma$, which means it is unavoidable to estimate $\tilde{K}(x, y)$ for a point x very close to the sample point $y \in \partial\Sigma$. To address this singularity issue and improve the accuracy, an approach referred to as *disk integration* is proposed. Recall that the input data \mathcal{P} samples the surface $\partial\Sigma$. Assume that each sample point $y \in \mathcal{P}$ represents a neighboring region on $\partial\Sigma$, denoted as $R(y)$, such that the set $\{R(y)\}_{y \in \mathcal{P}}$ decomposes the surface $\partial\Sigma$. $R(y)$ can be assumed to be the Voronoi region of y on $\partial\Sigma$. Then, $\tilde{\chi}(x) = \sum_{y \in \mathcal{P}} C(x, y)$, where

$$C(x, y) = \int_{R(y)} \tilde{K}(x, y) d\tau(y). \quad (7)$$

Note that $R(y)$ is unknown and a disk perpendicular to $\vec{N}(y)$ is used to approximate $R(y)$. The radius of this disk is estimated as the average distance of point y to its n -nearest samples in \mathcal{P} . In this article, $n = 10$ is fixed for all samples. We denote this disk $D(y)$, its radius $D(y).r$, and take the area of $D(y)$ to be the surface area $y.A$.

We approximate $C(x, y)$ using $\int_{D(y)} \tilde{K}(x, y) dy$. Note that even over the simple domain $D(y)$, the above integration does not have a closed form. Our approach is to approximate $D(y)$ by using l layers of partial annuli (see the shaded regions in Figure 4), and the integration of the kernel function $\tilde{K}(x, y)$ can be calculated analytically over each layer. Let x' be the projection of x on the plane containing $D(y)$. Denote by $C(r)$ the circle centered at x' of radius r , and

ALGORITHM 2: Gauss Surface Reconstruction

```

Input:  $\mathcal{P}$ : samples,  $D$ : maximum depth,  $\beta$ : width coefficient
Output:  $M$ : watertight iso-surface
Estimate  $D(p)$  for each sample  $p \in \mathcal{P}$ ;
Given  $D$ , construct an adaptive octree  $O$ ;
Compute representative samples  $\bar{p}$  for all cubes in  $O$ ;
Compute representative grid vertices  $\bar{v}$  for all cubes in  $O$ ;
Given  $\beta$ , estimate  $w(v)$  for each grid vertex  $v \in \mathcal{V}$ ;
Initialize  $f: \mathcal{V} \rightarrow \mathbb{R}$  to be zero;
Call FMM( $B_1^1, B_1^1, f$ );
Set the iso-value as the median of  $f$ ;
Extract the iso-surface  $M$  using marching cubes over  $O$ ;
Output  $M$ ;

```

$A(r, R)$ as the annulus centered at x' of inner radius r and outer radius R . Let $r_0 = \min_{z \in D(y)} \|x' - z\|$ and $r_l = \max_{z \in D(y)} \|x' - z\|$, and $r_i = r_0 + \frac{i(r_l - r_0)}{l}$, for $0 \leq i \leq l$. Here, r_0 is 0 if x' is in the disk. Let θ_i be the central angle of the arc $C(r_i) \cap D(y)$, and F_i be the fan spanned by the same arc. The partial annulus at the i th layer is $F_i \cap A(r_{i-1}, r_i)$. Set $d = \|x - x'\|$. Then we can approximate $C(x, y)$ as $\sum_{1 \leq i \leq l} c_i$ where

$$\begin{aligned} c_i &= \int_{F_i \cap A(r_{i-1}, r_i)} \tilde{K}(x, y) dy \\ &= -\frac{1}{4\pi} \int_0^{\theta_i} \int_{r_{i-1}}^{r_i} \frac{d}{(d^2 + r^2)^{3/2}} r dr d\theta \\ &= \frac{\theta_i d}{4\pi} \left(\frac{1}{\sqrt{d^2 + r_{i-1}^2}} - \frac{1}{\sqrt{d^2 + r_i^2}} \right) \end{aligned}$$

In the article, the number of layers $l = 20$ is fixed.

Furthermore, notice that if point x is far away from sample y , then integrand $\tilde{K}(x, y)$ over $D(y)$ becomes well approximated by the constant $\tilde{K}(x, y)$, and $C(x, y)$ can simply be evaluated by $\hat{C}(x, y) = \tilde{K}(x, y)y.A$. Set $R(x, y) = \frac{\|x - y\| + D(y).r}{\|x - y\| - D(y).r}$. The larger $R(x, y)$ is, the closer the function $\tilde{K}(x, y)$ over $D(y)$ is to the constant $\tilde{K}(x, y)$. In this article, when $R(x, s) > 2$, we approximate $C(x, y)$ using $\hat{C}(x, y)$.

High accuracy can be achieved in computing the integral by using disk integration. Figures 2 and 3 show the indicator function of unit spheres restricted to a line of points passing through the center, which were estimated by using the above approach from 1,000 random samples. The Hausdorff distance between the reconstructed triangle mesh and the original sphere is lower than 5×10^{-3} . Another advantage of disk integration is that it handles missing data in most cases. If the holes resulting from the missing data are relatively small and can be covered by disks, the integral formula integrates all the disks to automatically provide an accurate watertight surface.

Disk integration is a specific approach to address the singularity problem in our integral formula (5). It is possible to apply disk integration to other reconstruction methods even without singularity problem to improve the approximation accuracy.

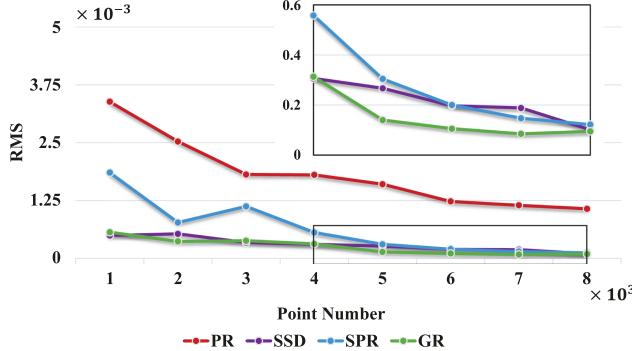


Fig. 6. The average error RMS of the reconstructions by different methods. The sub-figure on top-right is the closeup on the marked box.

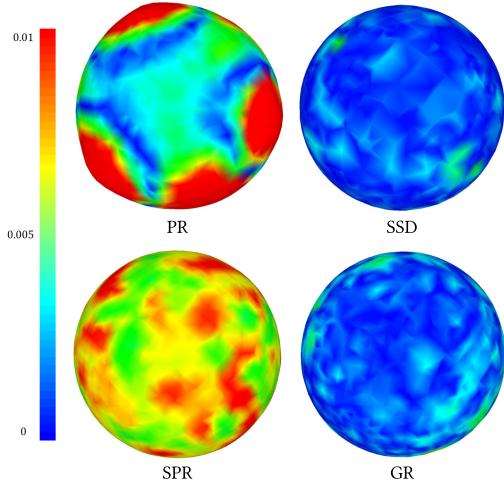


Fig. 7. The reconstructed unit sphere from 1,000 random samples. The color illustrates the RMS (relative to the bounding box diagonal) error distribution: small error in blue, and large error in red.

3.3 Fast Multipole Method

Given that the estimation of $\tilde{\chi}(x)$ by using the integral formula (1) is global, the kernel function $\tilde{K}(x, y)$ should be integrated over the entire surface $\partial\Sigma$ to obtain a correct estimation of $\tilde{\chi}(x)$. In particular, thresholding cannot be performed based on the value of $\tilde{K}(x, y)$ to avoid integrating the region where $\tilde{K}(x, y)$ is small. For the purpose of demonstration, assume Σ is a ball of radius r , and x is the center of the ball. For $y \in \partial\Sigma$, $\tilde{K}(x, y)$ can be made arbitrarily small by selecting a radius r that is sufficiently large. However, $\tilde{\chi}(x)$ remains the constant 1, which is independent of r . Therefore, a naive implementation requires at least $O(m|\mathcal{P}|)$ operations where $|\mathcal{P}|$ denotes the cardinality of set \mathcal{P} to estimate $\tilde{\chi}$ at m different locations. Fortunately, the kernel function $\tilde{K}(x, y)$ over two distant regions can be well approximated by a constant function. This condition enables us to speed up the estimation of χ by using the FMM, which improves the complexity to $O(m + |\mathcal{P}| \log |\mathcal{P}|)$.

In this subsection, we describe an implementation of FMM for the estimation of the reconstruction function $\tilde{\chi}$. An octree is

employed as the multiresolution data structure in FMM, and the same octree is also used for iso-surfacing $\tilde{\chi}$.

Given a set of samples \mathcal{P} and a maximum tree depth D , the minimal octree is constructed such that each sample falls into a leaf node of depth D . For a nonuniform sampling, we follow (Kazhdan et al. 2006) and reduce the depth for the samples in the sparse regions. We denote \mathcal{O} as the resultant octree, and \mathcal{V} as the set of grid vertices of octree \mathcal{O} . Our goal is to evaluate the reconstruction function at \mathcal{V} . Now consider the cubes $\{B_i^d\}_i$ of \mathcal{O} at depth d (see Figure 5). A cube B_i^d may be half open, that is, does not contain the faces with the maximum x , or y , or z coordinate, unless they are on the boundary of the bounding box. See the pink cube in Figure 5. Let $\mathcal{V}_i^d = \mathcal{V} \cap B_i^d$, see the blue dots in B_i^d in Figure 5, and $\mathcal{P}_i^d = \mathcal{P} \cap B_i^d$. Let \bar{v}_i^d be the representative vertex of B_i^d defined by

$$\bar{v}_i^d = \frac{\sum_{v \in \mathcal{V}_i^d} v}{|\mathcal{V}_i^d|},$$

where $|\mathcal{V}_i^d|$ denotes the cardinality of \mathcal{V}_i^d , and \bar{p}_i^d be the representative sample of B_i^d defined by

$$\bar{p}_i^d \cdot p = \frac{\sum_{p \in \mathcal{P}_i^d} p \cdot A \cdot p}{\sum_{p \in \mathcal{P}_i^d} p \cdot A},$$

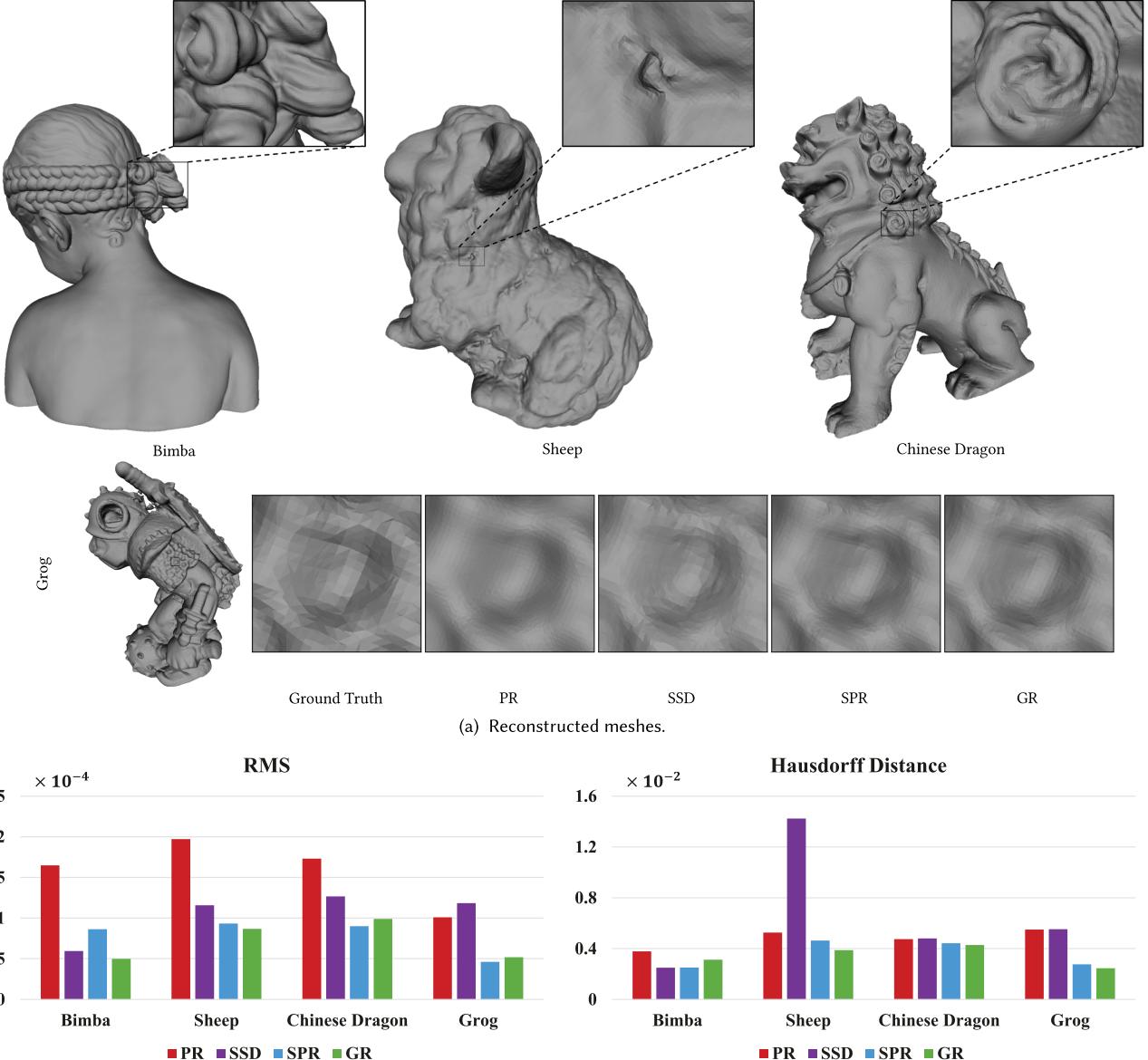
$$\vec{N}(\bar{p}_i^d) = \frac{\sum_{p \in \mathcal{P}_i^d} p \cdot A \cdot \vec{N}(p)}{\sum_{p \in \mathcal{P}_i^d} p \cdot A}, \text{ and}$$

$$\bar{p}_i^d \cdot A = \sum_{p \in \mathcal{P}_i^d} p \cdot A.$$

The disk $D(\bar{p}_i^d)$ is centered at \bar{p}_i^d , perpendicular to $\vec{N}(\bar{p}_i^d)$, and has area $\bar{p}_i^d \cdot A$. Let a_d be the side length of the cubes at depth d . The basic idea of our FMM implementation is as follows: we start with the cube at depth 1. In general, consider two cubes $B_i^{d'}$ and $B_j^{d''}$ at depth d' and depth d'' , respectively. Note that $B_i^{d'}$ and $B_j^{d''}$ may be the same cube. If $\|\bar{p}_i^{d'} - \bar{p}_j^{d''}\| \geq c a_d$, then for any grid vertex $v \in \mathcal{V}_j^{d''}$, we approximate $\sum_{p \in \mathcal{P}_i^{d'}} C(v, p)$ using $C(\bar{v}_j^{d''}, \bar{p}_i^{d'})$. Otherwise, we repeat the above procedure for any pairs of sub-cubes, one in $B_i^{d'}$ and the other in $B_j^{d''}$ until both become leaf nodes. When both are leaf nodes, we estimate $C(v, p)$ for an individual sample $p \in \mathcal{P}_i^{d'}$ and an individual grid vertex $v \in \mathcal{V}_j^{d''}$. Moreover, when the estimation of $C(\bar{v}, \bar{p})$ is invoked for a representative grid vertex \bar{v} and a representative sample \bar{p} , we assume that \bar{v} and \bar{p} are far away from each other and compute $C(\bar{v}, \bar{p})$ using the kernel function \tilde{K} . Therefore, associating a width to a representative grid vertex \bar{v} is not required. In this article, we fix the constant $c = \sqrt{2}$. Algorithm 1 shows our FMM implementation.

3.4 Selection of Width Coefficient and Iso-surface Extraction

We now move to the specification of width $w(x)$. Note that we only need to specify the width for the grid vertices \mathcal{V} . For a grid vertex $v \in \mathcal{V}$, let $v.B$ be the set of the leaf nodes in \mathcal{O} having v as one of their vertices. Set $w(v)$ to be β times the side length of the smallest cube in $v.B$, where β is a constant known as the width



(b) The RMS approximation error and the Hausdorff approximation error for the reconstructions of four point sets: Bimba, Sheep, Chinese dragon and Grog.

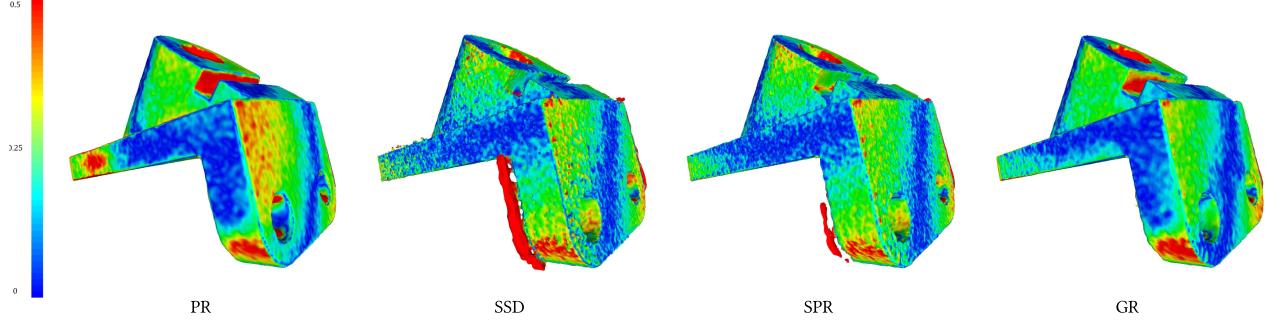
Fig. 8. The accuracy illustration. The function evaluation time: Bimba($|v| = 0.50$, PR: 63.79s, SSD: 27.83s, SPR: 95.85s, GR: 36.05s), Sheep($|v| = 0.16$, PR: 30.63s, SSD: 14.38s, SPR: 31.41s, GR: 14.26s), Chinese dragon($|v| = 0.66$, PR: 110.69s, SSD: 34.77s, SPR: 124.65s, GR: 64.88s), Grog($|v| = 0.88$, PR: 182.34s, SSD: 45.57s, SPR: 176.05s, GR: 99.28s). The number of samples is in millions.

coefficient. Then, the neighboring vertices $v \cdot \mathcal{V}$ of v in the octree is defined so that a grid vertex u is in $v \cdot \mathcal{V}$ if u and v are connected by an edge of a cube in $v \cdot B$. In this process, $w(v)$ and $w(u)$ differ significantly even when u and v are neighbors, and the resultant function $\tilde{\chi}$ may become rough. To address this issue, we smooth $w(v)$ by averaging the widths over the neighbors. Namely, set

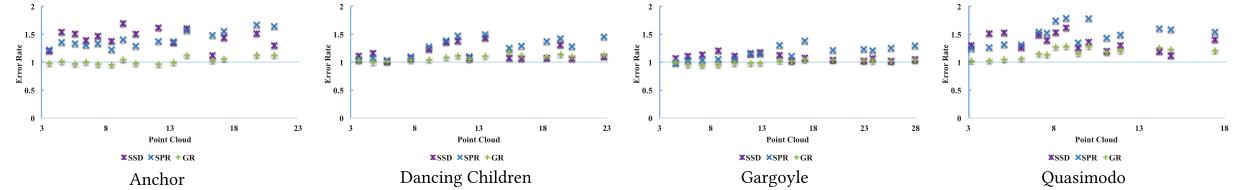
$$w(v) = \frac{\sum_{u \in v \cdot \mathcal{V}} w(u)}{|v \cdot \mathcal{V}|},$$

and repeat this averaging step for μ times. In this article, we set $\mu = 20$.

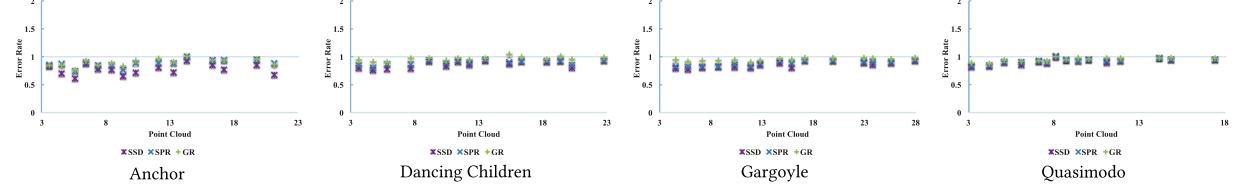
The width coefficient β provides a way to control the trade-off between the position accuracy and the smoothness of the reconstruction. Theoretically, the larger the value of β is, the smoother the reconstructed surface, but with a less accurate position. Detailed analysis about the coefficient β is shown in Section 4.4. We empirically set the value of β to 0.7 by default.



(a) Visualization of position errors for reconstruction of the Anchor model. Errors are visualized using a blue-green-red colormap, with blue corresponding to smaller errors and red to larger ones.



(b) Average angle error (relative to that of PR).



(c) Average position error (relative to that of PR).

Fig. 9. Results from the reconstruction benchmark.

Next, we introduce a way to calculate the iso-value and perform the interpolation. To make the samples locate uniformly inside and outside the surface $\partial\Sigma$, we set the iso-value γ to the median of $\tilde{\chi}$ at sample positions.

Regarding the interpolation, we assume two adjacent vertices $v_i \in \mathcal{V}$ and $v_j \in \mathcal{V}$ that came across the iso-surface, with $(\tilde{\chi}(v_i) - \gamma) \cdot (\tilde{\chi}(v_j) - \gamma) < 0$. From Equation (6), we observe that given the signed distance $d(v)$, the function value $\tilde{\chi}(v) - \frac{1}{2}$ is inversely proportional to its associated width $v.w$, where $\frac{1}{2}$ is the ideal iso-value. Then, the location of the crossing point $\tilde{v}_{ij} \in \partial\Sigma$ can be stated as the linear interpolation of $(\tilde{\chi}(v) - \gamma) \cdot w(v)$ given by:

$$\frac{\tilde{v}_{ij} - v_i}{0 - (\tilde{\chi}(v_i) - \gamma) \cdot w(v_i)} = \frac{v_j - v_i}{(\tilde{\chi}(v_j) - \gamma) \cdot w(v_j) - (\tilde{\chi}(v_i) - \gamma) \cdot w(v_i)}. \quad (8)$$

In addition, a hermite interpolation method inspired by Fuhrmann et al. (2015) was implemented. However, the accuracy of the two methods are practically the same. Thus, Equation (8) is applied to keep our method simple.

Finally, we follow (Kazhdan and Hoppe 2013) and use the marching cube method for adaptive grids to extract the iso-surface obtaining water-tight results.

Our method is summarized in Algorithm 2.

3.5 Parallel and GPU Implementation

For GPU implementation, we mainly refer to Burtscher and Pingali (2011) to solve the problem. The estimation of the

reconstruction function at different grid vertices is independent, which leads to a straightforward parallel implementation. In particular, we open new threads to execute the calls of $FMM(B_i^{d'}, B_j^{d''}, f)$ with $d', d'' \leq t$. The parameter t is selected to facilitate sufficient threads so that the load on each core is balanced, and the overhead of multi-threads is minimized simultaneously. In this article, we set $t = 5$ for CPU parallel and $t = 10$ for GPU implementation.

4 RESULTS

In this section, we evaluate our method (GR) in terms of accuracy, noise resilience, and efficiency. We also compare its performance to several state-of-the-art methods, including Poisson reconstruction (PR) (Kazhdan et al. 2006) and its variant screened Poisson reconstruction (SPR) (Kazhdan and Hoppe 2013), smooth signed distance reconstruction (SSD) (Calakli and Taubin 2011), and the dictionary learning reconstruction (Xiong et al. 2014). We perform the comparisons using the recent implementation of these methods available online. In particular, the performance of SSD has greatly improved in the new version compared with those reported in Kazhdan and Hoppe (2013). We followed (Kazhdan and Hoppe 2013) and set the weights for the different terms of the energy functional in SSD as 1 for value, 1 for gradient, and 0.25 for Hessian, in which we set the data fitting weight $\alpha = 4$ in SPR subsequently. Unless explicitly stated, we set the maximum depth

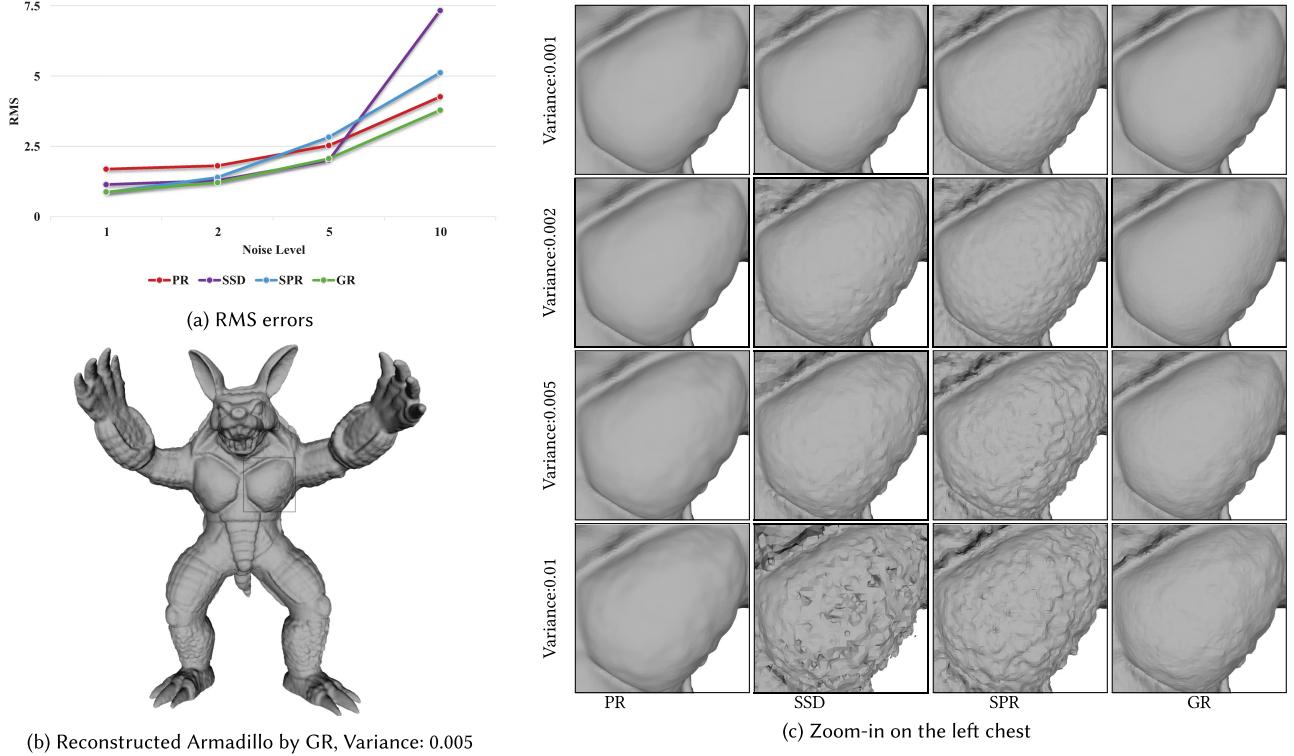


Fig. 10. Reconstructed surface of Armadillo from the samples perturbed by Gaussian noise of different variance. The variance is relative to the diameter of the bounding box.

$D = 10$ for octree construction in all methods and the width coefficient $\beta = 0.7$ in our method. All the experiments are performed on a Windows 7 workstation with an Intel Xeon E5-2690V3 CPU @2.6GHz and Nvidia GeForce GTX TITAN X GPU.

4.1 Accuracy

First, we consider the reconstruction of a unit sphere from samples, in which the accurate ground truth is known. We generated 1,000 to 8,000 samples according to a Gaussian mixture of eight Gaussian in \mathbb{R}^3 and then radially project them into a unit sphere. The average error RMS is used to measure the quality of the reconstructed surface.

Figure 6 shows the error statistics of the reconstructions by different methods. Our method shows higher accuracy compared with other methods. For 1,000 samples, we color the RMS error (relative to the bounding box diagonal) for each vertex to visualize the error distribution (see Figure 7). In this case, the sphere obtained by PR is visually not round.

Next, we consider general models. To estimate the numerical accuracy of the reconstruction results, we follow the same approach used by Berger et al. (2013). First, we sample points from a known mesh, or simply use its vertices, and then reconstruct the surfaces from this point set. Next, we use the Metro tool (Cignoni et al. 1998) to compute the Hausdorff distance (measuring the worse error) and the mean distance (measuring the average error) between the reconstructed mesh and the known mesh. Figure 8 shows the result. In general, SPR and GR have a comparable performance on

this set of models and have higher accuracy compared with PR and SSD.

We also apply the reconstruction methods to the data from the reconstruction benchmark (Berger et al. 2013). Due to limited space, we only report the results on four datasets, namely Anchor, Dancing Children, Gargoyle, and Quasimodo. Following (Kazhdan and Hoppe 2013), we set the maximum depth $D = 9$ in this experiment.

The error shown in Figure 9 is relative to that of PR. From Figure 9(a), PR and GR visually generate similar results, whereas SPR and SSD produces spurious sheets near the surface. However, the accuracy of GR is much better than that of PR. Figures 9(b) and 9(c) show the average angle error and the average position error, respectively. For this set of examples, PR shows the highest accuracy in terms of angle accuracy but the worst in position accuracy overall. In comparison, SSD shows the highest accuracy in terms of position accuracy. However, from Figure 9(a), SSD overfits the data. Our GR achieves a better balance between position and angle accuracy.

4.2 Noise Resilience

We evaluate GR on noisy data, including synthetic Gaussian noise and real-world scanned data with noise and missing data, in which we compare the performance of different reconstructions. At the end of this section, the ability of GR to fill holes is evaluated.

4.2.1 Synthetic Noise. In this example, we add different levels of noise to the Armadillo model by perturbing both positions and

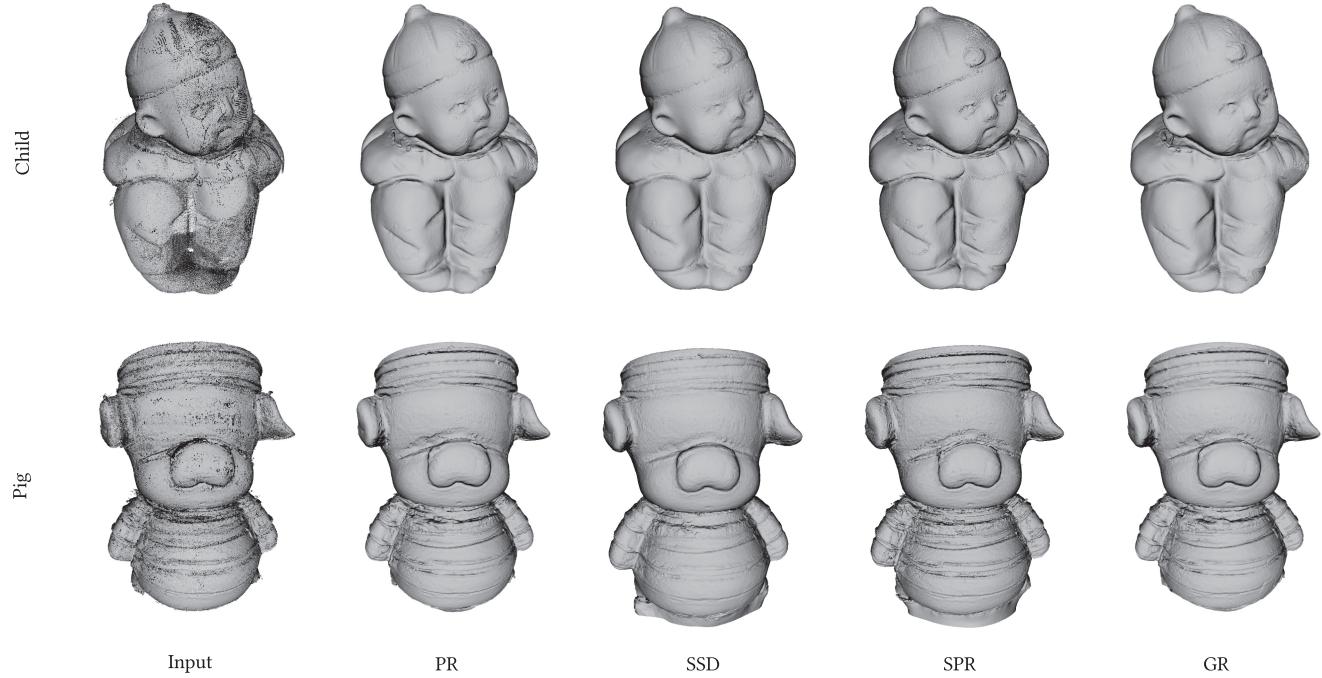


Fig. 11. The reconstructions of real-world scanned data.

normals of the samples according to Gaussian distribution of different variances.

Figure 10(b) shows the reconstructed surfaces using the proposed GR from the samples perturbed by a Gaussian with variance equal to 0.005 times the diagonal of the bounding box. Figure 10(c) shows the details of reconstructions at different noise levels by zooming in on the region marked in Figure 10(b). SPR and SSD appear to overfit the data, which results in sensitivity to noise and reconstruction of bumpy surfaces. In comparison, PR continuously produces smooth reconstructions, but its accuracy is the lowest (see Figure 10(a)). The reconstructions by GR are also smooth, with the added benefit of simultaneously preserving more details.

4.2.2 Real-world Scanned Data. We apply the reconstruction methods to the sampling obtained by scanning several real-world models using a Konica-Minolta Vivid 9i Laser Scanner. The obtained samplings contain noise and missing data and are highly nonuniform and unoriented. We use the “compute normals for point sets” function in Meshlab with default parameters to estimate the normal as a preprocess step (see the first column in Figure 11). In these examples, all methods suffer from overfitting with the default parameters described at the beginning of Section 4. Therefore, we tune the parameters to obtain the best result for each method. We set the value parameter to 0.25, the gradient parameter to 1, and the Hessian parameter to 1 in SSD. For SPR, the results are good if we set the weight $\alpha \in [0, 2]$. Figure 11 shows the results when $\alpha = 0$, which is equivalent to PR, and the results when the data fitting weight $\alpha = 1$. For GR, we set the width coefficient $\beta = 1.4$. The effects of different β are shown in Figure 24.

All the reconstruction results are visually comparable with careful parameter setting.

In addition, we apply our method to the Merlion model with noise and missing data and compared it to the dictionary learning method (Xiong et al. 2014). As Figure 12 shows, the two methods reconstruct smooth and accurate models. The left column shows the reconstruction result of GR where the octree depth is set to 10. The right column shows the detailed logo of Xiong et al. (2014) and GR where the octree depth is set to 9 and 10. With comparable output vertices, our method at depth 9 achieves accuracy similar to Xiong et al. (2014). From Figure 12, the mesh reconstructed by our method with depth 10 is of higher resolution with respect to the bricks and words.

In addition, as stated in Xiong et al. (2014), the running time of Xiong et al. (2014) is slower than that of SPR. The efficiency of our method is a bit faster than SPR (see Section 4.3 for details). It can be inferred that our method is faster than Xiong et al. (2014).

4.2.3 Datasets with Significant Noise and Missing Parts. We evaluate our method on the Owl model with large parts of missing data and significant noise. Figure 13 demonstrates that our method performs well in processing the missing portion. Moreover, our method is resilient to noise and recovers the feature in detail. We even manually cut the head of the owl to show that our method fills the large or small holes with several surfaces. In general, large holes are filled with flat surfaces.

We also run our method over a few models downloaded from EPFL, which are generated by photo scanning 3D statues that are full of noise and missing significant parts. Figure 15 shows the results from several viewpoints. Our method generates good quality

Table 1. Reconstruction Time on Different Models. The Output Mesh Vertices in Million: Bimba: PR(1.11), SSD(1.40), SPR(1.97), GR(1.11); Sheep: PR(0.57), SSD(0.69), SPR(0.59), GR(0.57); Chinese Dragon: PR(1.80), SSD(1.76), SPR(2.42), GR(1.83); Grog: PR(3.26), SSD(2.55), SPR(3.55), GR(3.28); Child: PR(1.88), SSD(1.67), SPR(2.32), GR(1.77); Pig: PR(2.10), SSD(1.63), SPR(2.63), GR(1.90). All the Tree Construction and Iso-surfacing are Processed on CPU, So the GPU Part is Omitted

Method	Cores	Time in Seconds							
		Bimba				Sheep			
		Tree Cons	Func-Eval	Iso-Surf	Total	Tree Cons	Func-Eval	Iso-Surf	Total
PR	CPU 1 core	6.00	63.79	30.94	100.72	2.08	30.63	13.20	45.92
SSD	CPU 1 core	2.90	27.83	9.59	40.33	0.96	14.38	4.51	19.85
SPR	CPU 1 core	8.13	95.85	29.93	133.92	2.78	31.41	9.58	43.77
	CPU 10 cores	8.36	14.86	10.55	33.77	2.80	5.60	3.30	11.70
GR	CPU 1 core	13.90	36.05	17.12	67.07	6.03	14.26	8.39	28.69
	CPU 10 cores	10.37	4.21	8.28	22.86	4.39	1.71	4.18	10.28
	GPU	-	1.72	-	-	-	0.77	-	-
Method	Cores	Time in Seconds							
		Chinese Dragon				Grog			
		Tree Cons	Func-Eval	Iso-Surf	Total	Tree Cons	Func-Eval	Iso-Surf	Total
PR	CPU 1 core	8.55	110.69	54.56	173.79	10.96	182.34	93.20	286.50
SSD	CPU 1 core	3.78	34.77	12.21	50.76	5.13	45.57	16.96	67.66
SPR	CPU 1 core	10.75	124.65	38.32	173.72	13.03	176.05	54.22	243.30
	CPU 10 cores	10.65	19.34	12.88	42.86	13.04	26.21	18.55	57.80
GR	CPU 1 core	22.10	64.88	27.87	114.85	36.00	99.28	46.50	181.78
	CPU 10 cores	16.42	7.32	13.68	37.42	24.72	11.14	23.18	59.04
	GPU	-	3.03	-	-	-	5.30	-	-
Method	Cores	Time in Seconds							
		Child				Pig			
		Tree Cons	Func-Eval	Iso-Surf	Total	Tree Cons	Func-Eval	Iso-Surf	Total
PR	CPU 1 core	9.48	128.60	58.15	196.22	12.29	168.62	71.00	251.91
SSD	CPU 1 core	4.25	41.20	11.56	57.01	5.17	47.03	11.73	63.93
SPR	CPU 1 core	11.85	127.83	37.96	177.64	13.27	143.48	42.00	198.74
	CPU 10 cores	11.74	21.91	12.59	46.24	13.18	22.90	13.84	49.92
GR	CPU 1 core	25.40	70.19	29.39	124.98	29.42	91.05	33.11	153.58
	CPU 10 cores	18.67	8.15	15.15	41.97	21.64	10.50	17.19	49.33
	GPU	-	3.44	-	-	-	4.37	-	-

meshes although large parts of the data are missing. To show a striking comparison, we compare our methods with the existing smoothest PR method. Figure 16 also shows that our method performs well with poor quality data, especially with great noise.

4.2.4 Dataset with Sharp Features. Finally, we apply our method to models with sharp features. Figure 17 shows the result reconstructed from the well-known octa-flower with sharp edges. Our method performs well in such instances.

4.3 Efficiency

In this subsection, we discuss the efficiency of our method, particularly its parallel implementation. The reconstruction time shown in Table 1 includes the tree construction, implicit function evaluation, and iso-surfacing time. The function evaluation time of [S]PR includes the time that sets the boundary constraints. However, explicitly setting the boundary constraints is not necessary in our method.

As Table 1 shows, PR (version 3.0) is the slowest method among the four reconstructions. In the single thread implementation, SSD

(version 3.0) is the fastest mainly due to the employment of hash octree, and our GR is faster than that of SPR (version 9.01). The current implementation of PR, SPR, and GR does not use hash octree. Our implementation of iso-surfing is in fact the same as SPR. However, comparing to PR or SPR, our method often output the models with fewer vertices. We believe this is the reason that our surface extraction is faster than that of SPR.

For the multithreads implementation, Table 1 records that the parallel implementation of our GR has a practically negligible overhead and achieves a nearly perfect linear speed-up. In addition, the GPU implementation performed nearly 25 times faster than the single thread implementation. In Table 1, we also show the running time of the parallel SPR implementation, which is available to the public. On average, GR is about 1.5 times as fast as SPR in single CPU tests. With multiple CPUs, GR is also faster than SPR except in the test of “Grog.”

4.4 Parameter Analysis

In this subsection, we discuss the effects of different choices of parameters.

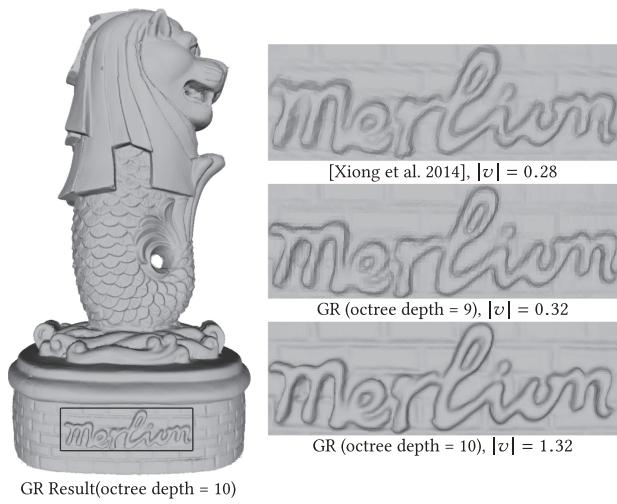


Fig. 12. The reconstructed model from merlion by Xiong et al. (2014) and GR of different octree depth. $|v|$ denotes the number of vertices in millions of the reconstructed mesh.

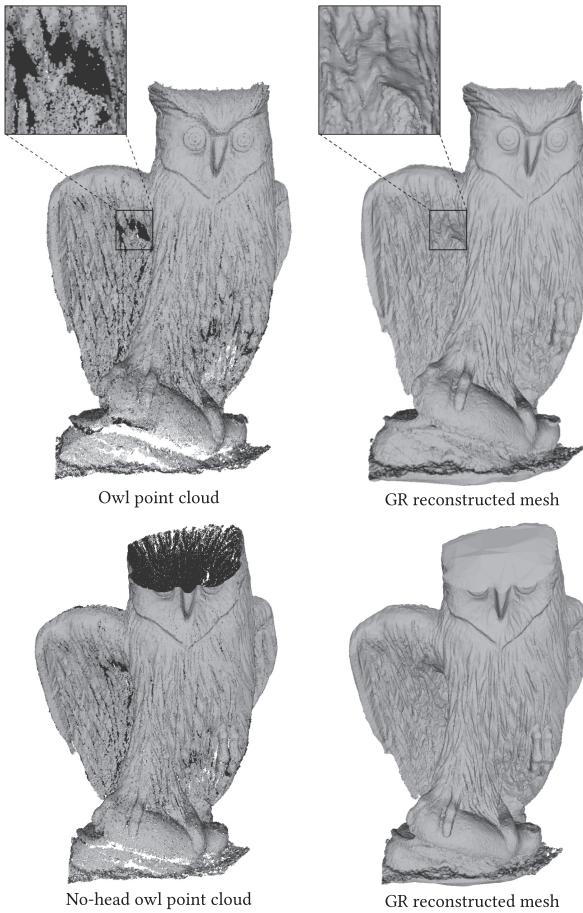


Fig. 13. The reconstructed model from the Owl point cloud with large missing data.

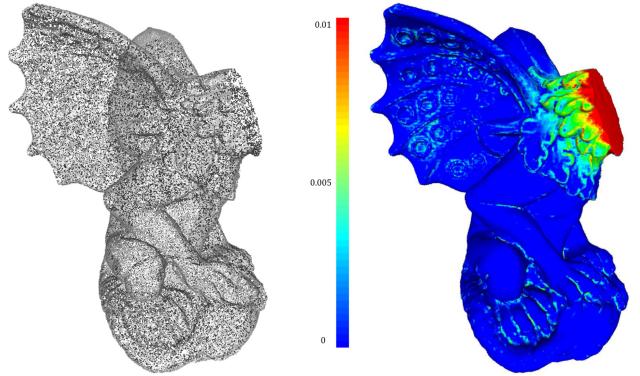


Fig. 14. Error colormap for the “gargoyle” model with head manually cut.

4.4.1 Kernel Coefficient α in Equation (4). As shown in Equation (6), the kernel coefficient is (almost) linear near the surface, which is desirable for iso-surfacing. Figure 19 shows the Hausdorff distance and RMS error with $\alpha = 0$ and $\alpha = 1$. Although the differences are very small, as shown in Figure 19, most of the measures are better when $\alpha = 0$. When $\alpha = 0$, Equation (6) is strictly a signed distance function, which can be accurately estimated by the modified linear interpolation.

4.4.2 Nearest Neighbors. Recall that we estimate the radius of the disk $D(y), y \in \partial\Sigma$ by computing the average distance to the n -nearest samples in \mathcal{P} . Figure 20 shows the RMS error and Hausdorff distance with different n in uniform and nonuniform samples. As we can see, RMS error and Hausdorff distances are barely affected by n .

4.4.3 Disk Layers. Another parameter in *disk integration* is the number of layers l . Figure 21 shows that the effect of layer numbers l in the accuracy. As shown in Figure 21, the accuracy becomes better as the number of layers increases. The trade-off is larger computational time.

4.4.4 Apply Disk Resampling in PR. The idea of disk integration can be applied to other methods like PR by resampling the disks. In this part, we resample the original point cloud of the Gargoyle model by adding points on the related disks and see its effect in PR. As Figure 22 shows, Initially, as the number of resampling points increases PR does benefit from disk resampling to get high accuracy. Later, the error seems to be saturated and does not decrease anymore.

4.4.5 Width Coefficient. The width coefficient β is the parameter to balance the position accuracy and the smoothness of the reconstructions. Figure 23 shows that when β becomes larger, the error in angle decreases, but position accuracy is compromised. If β is too large (e.g., beyond 0.6 in this example), the position accuracy and angle accuracy both decrease. This suggests that we can not use too large β . However, with high noise level in the input, we need large β to make the model resilient to noise, as shown in Figure 24.

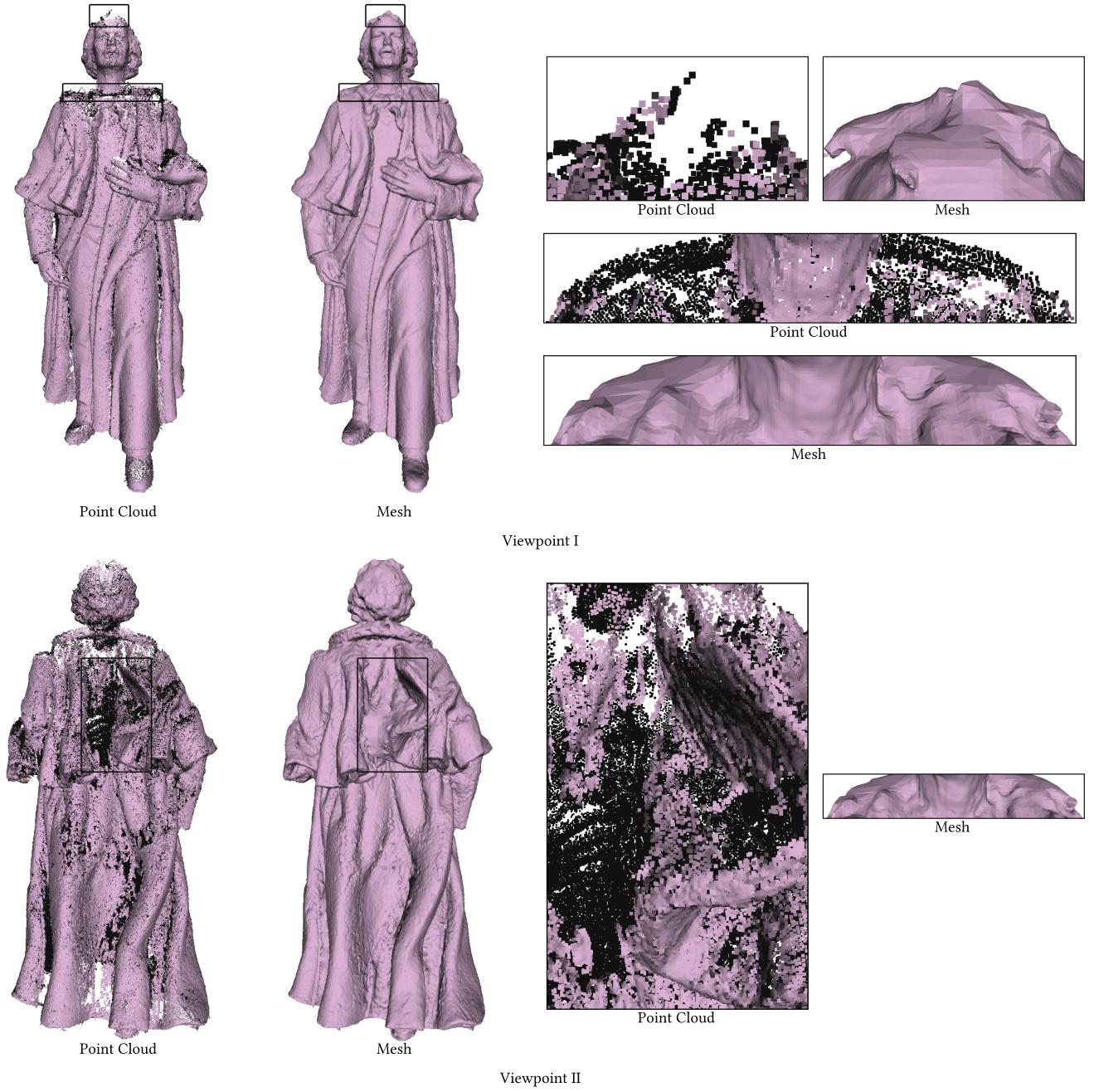


Fig. 15. The reconstructed models from Paderwski point clouds, which contains large part of missing data and noise.

5 LIMITATIONS

Our method based on the modified Gauss Formula mainly suffers the following limitations:

- **Nonclosed Surfaces.** Our method does not apply to non-closed surfaces such as “bas-relief” because the Gauss lemma (Lemma 3.1) does not apply. Although our method still works when large holes exist (such as Figure 13), evaluating accurate function values near the open boundary is difficult.

Specifically, big errors are obtained in the resultant mesh near the open boundary. As shown in Figure 14, the error is much larger on the open boundary compared with other parts. Figure 25 shows a failure result of our method when coming with nonclosed surfaces, the “bas-relief” was closed in our result.

- **Disk Integration.** For the disk integration, each data point is replaced by a small tangent disk. This operation can increase the approximation order from linear to quadratic in smooth



Fig. 16. The reconstructed model from noisy dataset Perpetual Pace Woman.

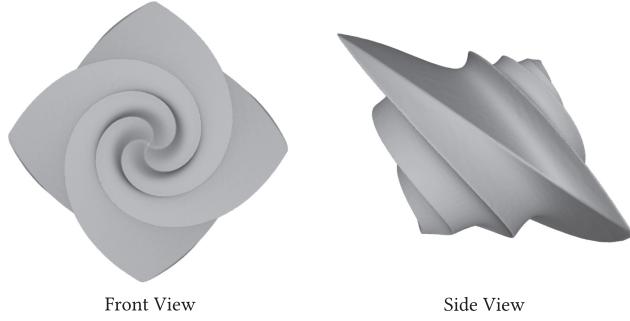


Fig. 17. The reconstructed model from octa-flower with sharp edges.

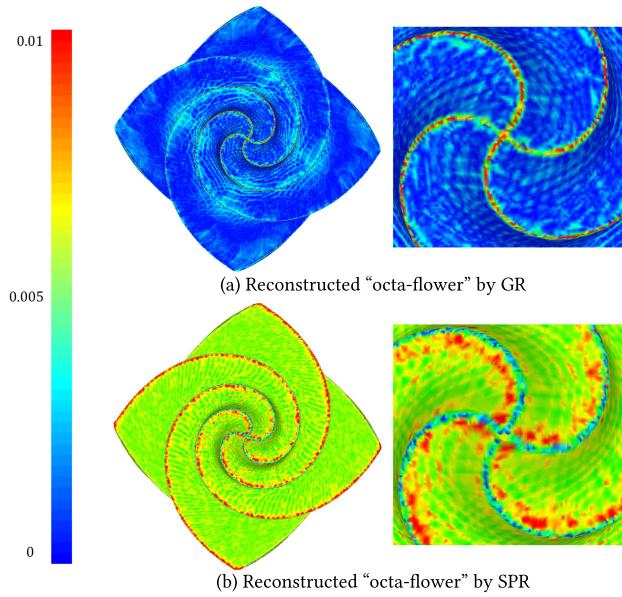
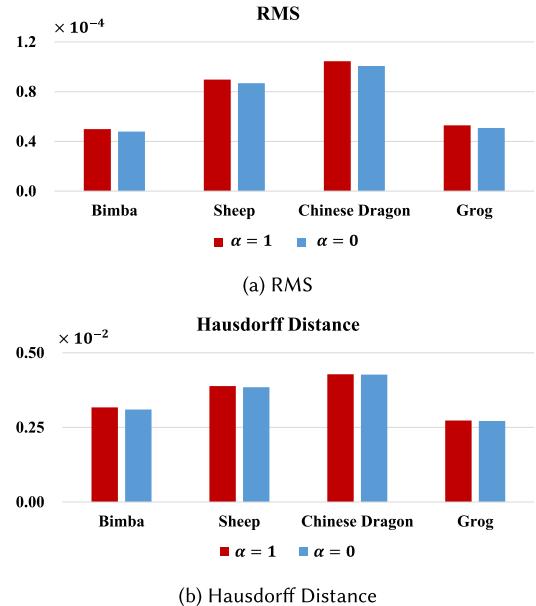
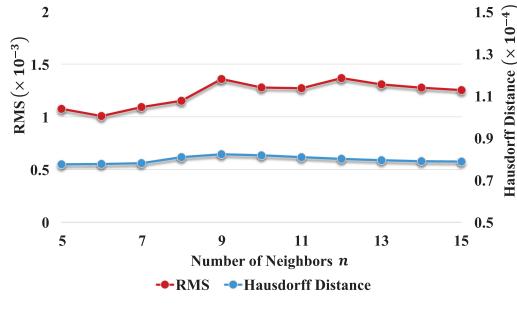


Fig. 18. Error colormap for the ‘octa-flower’ for different methods.

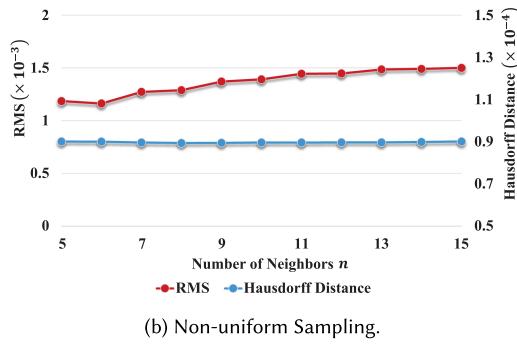
Fig. 19. RMS error and Hausdorff distance of reconstructions from bimba, sheep, Chinese dragon, and grog with different kernel coefficient α .

regions but may cause severe errors near sharp features due to normal discontinuity. As the first row of Figure 18 shows, the accuracy of our method is very high; that is, most of the octa-flower is blue, indicating a small error. However, the accuracy near the sharp features is relatively low due to disk integration. Unlike our method, SPR did not experience the limitation of disk integration. The accuracy near the sharp features is much higher compared with other parts.

- **Inconsistent Normals.** Like the other reconstruction methods as PR, SPR, and SSD, our method requires the normals to be consistently oriented. Inconsistent normals may result in a wrong implicit function and fail the reconstruction.



(a) Uniform Sampling.



(b) Non-uniform Sampling.

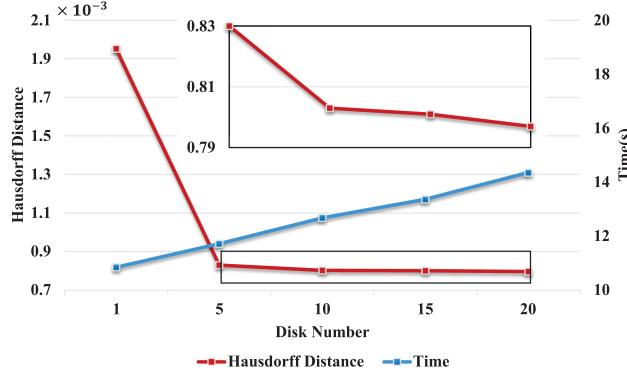
Fig. 20. RMS error and Hausdorff distance with different number of neighbors n to estimate the radius for disk D .

Fig. 21. Relationship between disk layers and Hausdorff distance.

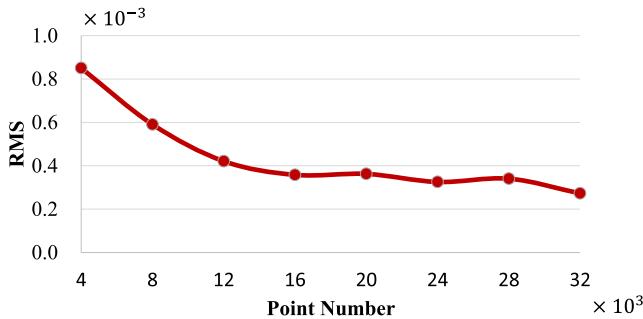


Fig. 22. RMS for PR after applying disk resampling in the gargoyle model. The original point number is 4,000. As the number of sampling increases, the RMS error decreases.

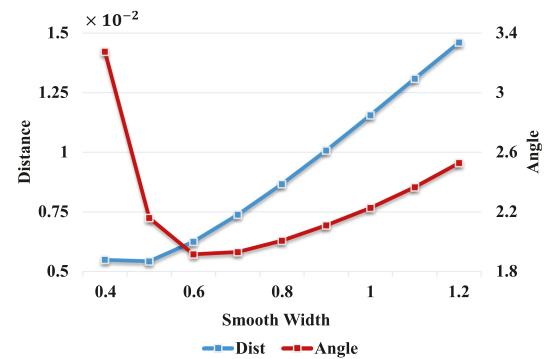
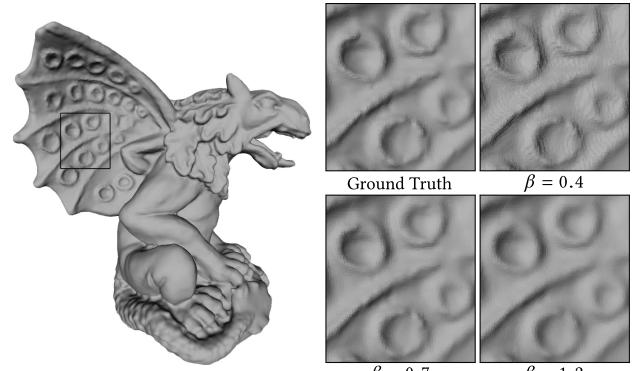


Fig. 23. Choice of width coefficient. The first row shows reconstructions; the second row shows the average position error (Dist) and the average angle error using the reconstruction benchmark (Berger et al. 2013).

6 CONCLUSIONS

We presented a surface reconstruction method based on a modified Gauss formula. Our method is based on an implicit function that combines the near-surface signed distance function and off-surface indicator function. By applying this function, the proposed method avoided the smoothing operation where the magnitude of error is difficult to estimate. Thus, the resultant mesh is of higher accuracy, and the error analysis is relatively easier because all the operations are explicit. Moreover, the proposed method exhibited the following advantages of both functions: resiliency against noise and missing data, less spurious sheets. Moreover, our reconstruction function is estimated directly based on Gauss lemma without solving any linear system.

This direct approach, aided by disk integration and FMM, enables our method to be simple, accurate, and achieve parallel implementation easily. Therefore, the proposed method is very efficient.

In the future, we will try to further speed up the algorithm by using hash octrees or achieve enhanced implementation of the FMM algorithm on the GPU. In addition, we will try to work out a closed form formula for disk integration as stated in the work of Hubert (2012). Finally, we plan to investigate the theoretical property of the proposed method to analyze the position approximation and normal approximation errors.

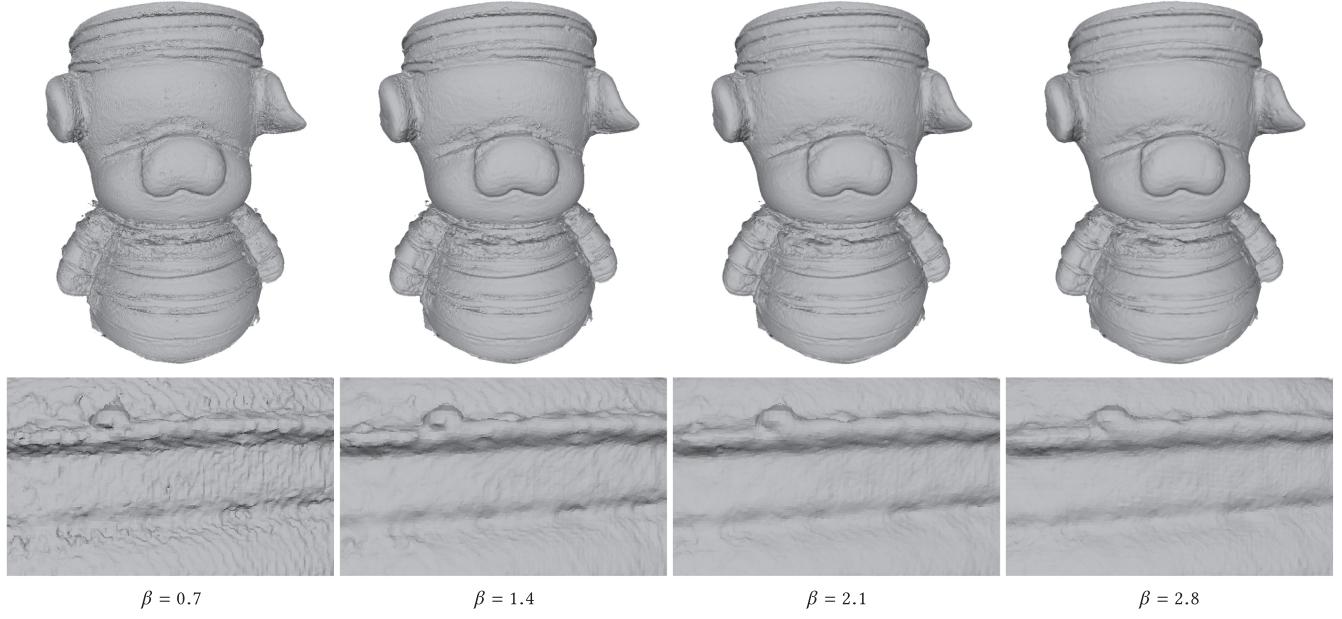
Fig. 24. Reconstructions with width different β .

Fig. 25. The reconstructed “bas-relief” models.

APPENDIX

A DERIVATION OF EQUATION (6)

Recall that the modified Gauss formula is given as follows:

$$\tilde{\chi}(x) = \int_{\partial\Sigma} \tilde{K}(x, y) d\tau(y), \quad (9)$$

where $\tilde{K}(x, y)$ is the modified kernel function,

$$\tilde{K}(x, y) = \begin{cases} K(x, y), & \|x - y\| \geq w(x), \\ -\alpha \frac{(x - y) \cdot \vec{N}(y)}{4\pi \cdot w^3(x)}, & \|x - y\| < w(x), \alpha \in \{0, 1\}. \end{cases}$$

and $K(x, y)$ is the kernel function in the original Gauss formula,

$$K(x, y) = -\frac{1}{4\pi} \frac{(x - y) \cdot \vec{N}(y)}{\|x - y\|^3}.$$

For each point x , we split the integral to two parts based on the small parameter $w(x)$ associated to x .

$$\tilde{\chi}(x) = \int_{\partial\Sigma} \tilde{K}(x, y) d\tau(y) \quad (10)$$

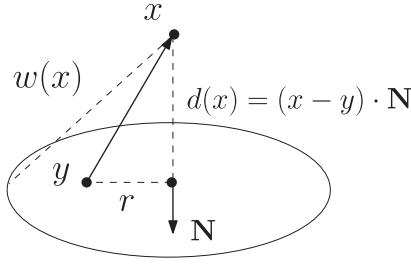
$$= \int_{B(x, w(x)) \cap \partial\Sigma} \tilde{K}(x, y) d\tau(y) + \int_{\partial\Sigma \setminus B(x, w(x))} K(x, y) d\tau(y) \quad (11)$$

$$= \int_{B(x, w(x)) \cap \partial\Sigma} \tilde{K}(x, y) d\tau(y) + \int_{\partial\Sigma} K(x, y) d\tau(y) \quad (12)$$

$$- \int_{B(x, w(x)) \cap \partial\Sigma} K(x, y) d\tau(y) \quad (13)$$

$$= \int_{B(x, w(x)) \cap \partial\Sigma} \tilde{K}(x, y) d\tau(y) \quad (14)$$

$$- \int_{B(x, w(x)) \cap \partial\Sigma} K(x, y) d\tau(y) + \chi(x) \quad (15)$$

Fig. 26. Disk approximates $B(x, w(x)) \cap \partial\Sigma$.

The last equality comes from the Gauss formula and $\chi(x)$ is the characteristic function in Gauss formula.

$w(x)$ is a small number, under the assumption that the surface Σ is smooth, $B(x, w(x)) \cap \partial\Sigma$ can be well approximated by a disk. It is easy to verify that for $\tilde{\chi}(x)$, the accuracy of this approximation is $O(|w(x)|)$. The integral in the disk can be calculated explicitly using polar coordinate, as shown in Figure 26.

$$\int_{B(x, w(x)) \cap \partial\Sigma} \tilde{K}(x, y) d\tau(y) \quad (16)$$

$$= \frac{1}{4\pi} \int_0^{2\pi} \int_0^{\sqrt{w^2(x) - d^2(x)}} \frac{d(x)}{(d^2(x) + r^2)^{3/2}} r dr d\theta + O(|w(x)|) \quad (17)$$

$$= -\frac{\alpha}{4} \frac{d(x)}{w^3(x)} (w^2(x) - d^2(x)) + O(|w(x)|). \quad (18)$$

and

$$\int_{\partial\Sigma \setminus B(x, w(x))} K(x, y) d\tau(y) \quad (19)$$

$$= \frac{1}{4\pi} \int_0^{2\pi} \int_0^{\sqrt{w^2(x) - d^2(x)}} \frac{d(x)}{(d^2(x) + r^2)^{3/2}} r dr d\theta + O(|w(x)|) \quad (20)$$

$$= \frac{d(x)}{2} \left(\frac{1}{|d(x)|} - \frac{1}{w(x)} \right) + O(|w(x)|). \quad (21)$$

Together with the expression of $\chi(x)$, $\chi(x) = \frac{1}{2} - \frac{d(x)}{2|d(x)|}$, we have

$$\tilde{\chi}(x) = \frac{1}{2} - \left(\frac{1}{2} + \frac{\alpha}{4} \right) \frac{d(x)}{w(x)} + \frac{\alpha d^3(x)}{4w^3(x)} + O(|w(x)|). \quad (22)$$

When $\alpha = 0$, up to the approximation error $O(|w(x)|)$, $\tilde{\chi}(x)$ is exactly a signed distance function near the surface, which is desirable for extracting iso-surface. Based on this calculation, we use the modified Gauss formula with $\alpha = 0$.

ACKNOWLEDGMENTS

We would like to thank Professor Hui Zhang and her students for providing the lady, child and pig model, AIM SHAPE for the bimba, sheep, Chinese dragon, grog, and octa-flower model, EPFL Scanning 3D Statues from Photos course for the Owl, Paderwski, and Perpetual Pace Woman model. We would also like to thank the authors of Xiong et al. (2014) for providing the merlion model and Hualiang Xie for addressing some GPU issues. Finally, we would like to thank the anonymous reviewers for valuable suggestions.

REFERENCES

- Pierre Alliez, David Cohen-Steiner, Yiyang Tong, and Mathieu Desbrun. 2007. Voronoi-based variational reconstruction of unoriented point sets. In *Symposium on Geometry processing*, Vol. 7. 39–48.
- Nina Amenta, Marshall Bern, and Manolis Kamvysselis. 1998. A new Voronoi-based surface reconstruction algorithm. In *Proceedings of SIGGRAPH'98*. ACM, 415–421. DOI : <https://doi.org/10.1145/280814.280947>
- N. Amenta, S. Choi, T. K. Dey, and N. Leekha. 2002. A simple algorithm for homeomorphic surface reconstruction. *Int. J. Comput. Geom. Appl.* 12 (2002), 125–141.
- Nina Amenta, Sunghee Choi, and Ravi Krishna Kolluri. 2001. The power crust, unions of balls, and the medial axis transform. *Comput. Geom.* 19, 2 (2001), 127–153.
- Nina Amenta and Yong Joo Kil. 2004a. Defining Point-set Surfaces. *ACM Trans. Graph.* 23, 3 (Aug. 2004), 264–270. DOI : <https://doi.org/10.1145/1015706.1015713>
- Nina Amenta and Yong Joo Kil. 2004b. The domain of a point set surface. In *SPBG*. 139–147.
- Matthew Berger, Joshua A. Levine, Luis Gustavo Nonato, Gabriel Taubin, and Claudio T. Silva. 2013. A benchmark for surface reconstruction. *ACM Trans. Graph.* 32, 2, Article 20 (April 2013), 17 pages. DOI : <https://doi.org/10.1145/2451236.2451246>
- Matthew Berger, Andrea Tagliasacchi, Lee M. Seversky, Pierre Alliez, Gael Guennebaud, Joshua A. Levine, Andrei Sharf, and Claudio T. Silva. 2017. A survey of surface reconstruction from point clouds. *Comput. Graphics Forum* 36, 1 (2017), 301–329.
- Jean-Daniel Boissonnat and Steve Oudot. 2005. Provably good sampling and meshing of surfaces. *Graph. Models* 67, 5 (Sept. 2005), 405–451. DOI : <https://doi.org/10.1016/j.gmod.2005.01.004>
- Martin Burtscher and Keshav Pingali. 2011. An efficient CUDA implementation of the tree-based barnes hut n-body algorithm. In *GPU Computing Gems Emerald Edition*. 75.
- Fatih Calakli and Gabriel Taubin. 2011. SSD: Smooth signed distance surface reconstruction. *Comput. Graphics Forum* 30, 7 (2011), 1993–2002.
- J. C. Carr, R. K. Beatson, J. B. Cherrie, T. J. Mitchell, W. R. Fright, B. C. McCallum, and T. R. Evans. 2001. Reconstruction and representation of 3D objects with radial basis functions. In *Proceedings of SIGGRAPH'01*. ACM, 67–76. DOI : <https://doi.org/10.1145/383259.383266>
- Jiazhou Chen, Gael Guennebaud, Pascal Barla, and Xavier Granier. 2013. Non-oriented MLS gradient fields. *Comput. Graphics Forum* 32, 98–109.
- Paolo Cignoni, Claudio Rocchini, and Roberto Scopigno. 1998. Metro: measuring error on simplified surfaces. *Comput. Graphics Forum* 17, 2 (1998), 167–174.
- Brian Curless and Marc Levoy. 1996. A volumetric method for building complex models from range images. In *Proceedings of SIGGRAPH'96*. ACM, 303–312. DOI : <https://doi.org/10.1145/237170.237269>
- Tamal K. Dey and Samrat Goswami. 2004. Provable surface reconstruction from noisy samples. In *Proceedings of SCG'04*. ACM, 330–339. DOI : <https://doi.org/10.1145/997817.997867>
- Tamal K. Dey and Jian Sun. 2005. An adaptive MLS surface for reconstruction with guarantees. In *Proceedings of SGP'05*. Eurographics Association, Article 43, 43–52.
- Herbert Edelsbrunner. 2003. Surface reconstruction by wrapping finite sets in space. In *Discrete and Computational Geometry. Algorithms and Combinatorics*, Vol. 25. Springer, Berlin, 379–404. DOI : https://doi.org/10.1007/978-3-642-55566-4_17
- Simon Fuhrmann and Michael Goesele. 2014. Floating scale surface reconstruction. *ACM Trans. Graph.* 33, 4, Article 46 (July 2014), 11 pages. DOI : <https://doi.org/10.1145/2601097.2601163>
- Simon Fuhrmann, Michael Kazhdan, and Michael Goesele. 2015. Accurate isosurface interpolation with hermite data. In *2015 International Conference on 3D Vision (3DV)*. IEEE, 256–263.
- Joachim Giesen and Matthias John. 2008. The flow complex: A data structure for geometric modeling. *Comput. Geom.* 39, 3 (2008), 178–190. DOI : <https://doi.org/10.1016/j.comgeo.2007.01.002>
- Leslie Greengard and Vladimir Rokhlin. 1987. A fast algorithm for particle simulations. *J. Comput. Physics* 73, 2 (1987), 325–348.
- Hugues Hoppe, Tony DeRose, Tom Duchamp, John McDonald, and Werner Stuetzle. 1992. Surface reconstruction from unorganized points. In *Proceedings of SIGGRAPH'92*. ACM, 71–78. DOI : <https://doi.org/10.1145/133994.134011>
- Evelyne Hubert. 2012. Convolution surfaces based on polygons for infinite and compact support kernels. *Graphical Models* 74, 1 (2012), 1–13.
- Alec Jacobson, Ladislav Kavan, and Olga Sorkine-Hornung. 2013. Robust inside-outside segmentation using generalized winding numbers. *ACM Trans. Graph.* 32, 4, Article 33 (July 2013), 12 pages. DOI : <https://doi.org/10.1145/2461912.2461916>
- Michael Kazhdan, Matthew Bolitho, and Hugues Hoppe. 2006. Poisson surface reconstruction. In *Proceedings of SGP'06*. Eurographics Association, 61–70.
- Michael Kazhdan and Hugues Hoppe. 2013. Screened Poisson surface reconstruction. *ACM Trans. Graph.* 32, 3, Article 29 (July 2013), 13 pages. DOI : <https://doi.org/10.1145/2487228.2487237>
- Michael Kazhdan, Allison Klein, Ketan Dalal, and Hugues Hoppe. 2007. Unconstrained isosurface extraction on arbitrary octrees. In *Symposium on Geometry Processing*, Vol. 7.

- Ravikrishna Kolluri, Jonathan Richard Shewchuk, and James F. O'Brien. 2004. Spectral surface reconstruction from noisy point clouds. In *Proceedings of SGP'04*. ACM, 11–21. DOI :<https://doi.org/10.1145/1057432.1057434>
- David Levin. 2004. Mesh-independent surface interpolation. In *Geometric Modeling for Scientific Visualization*. Springer, 37–49.
- William E. Lorensen and Harvey E. Cline. 1987. Marching cubes: A high resolution 3D surface construction algorithm. In *Proceedings of SIGGRAPH'87*. ACM, 163–169. DOI :<https://doi.org/10.1145/37401.37422>
- Shigeru Muraki. 1991. Volumetric shape description of range data using “blobby model.” In *Proceedings of SIGGRAPH'91*. ACM, 227–235. DOI :<https://doi.org/10.1145/122718.122743>
- A. Cengiz Özti̇rel, Gael Guennebaud, and Markus Gross. 2009. Feature Preserving Point Set Surfaces based on Non-Linear Kernel Regression. *Comput. Graphics Forum* 28, 493–501.
- Scott Schaefer and Joe Warren. 2004. Dual marching cubes: Primal contouring of dual grids. In *Proceedings of the 12th Pacific Conference on Computer Graphics and Applications (PG'04)*. IEEE, 70–76.
- Chen Shen, James F. O'Brien, and Jonathan R. Shewchuk. 2004. Interpolating and approximating implicit surfaces from polygon soup. In *ACM SIGGRAPH 2004 Papers (SIGGRAPH'04)*. ACM, 896–904. DOI :<https://doi.org/10.1145/1186562.1015816>
- Christian Walder, Olivier Chapelle, and Bernhard Schölkopf. 2005. Implicit Surface Modelling As an Eigenvalue Problem. In *Proceedings of ICML'05*. ACM, 936–939. DOI :<https://doi.org/10.1145/1102351.1102469>
- W. L. Wendland. 2009. On the double layer potential. In *Analysis, Partial Differential Equations and Applications: The Vladimir Maz'ya Anniversary Volume* (2009), 319–334.
- Jane Wilhelms and Allen Van Gelder. 1992. Octrees for faster isosurface generation. *ACM Trans. Graph.* 11, 3 (July 1992), 201–227. DOI :<https://doi.org/10.1145/130881.130882>
- Shiyao Xiong, Juyong Zhang, Jianmin Zheng, Jianfei Cai, and Ligang Liu. 2014. Robust surface reconstruction via dictionary learning. *ACM Trans. Graph.* 33, 6, Article 201 (Nov. 2014), 12 pages. DOI :<https://doi.org/10.1145/2661229.2661263>

Received November 2016; revised July 2018; accepted August 2018