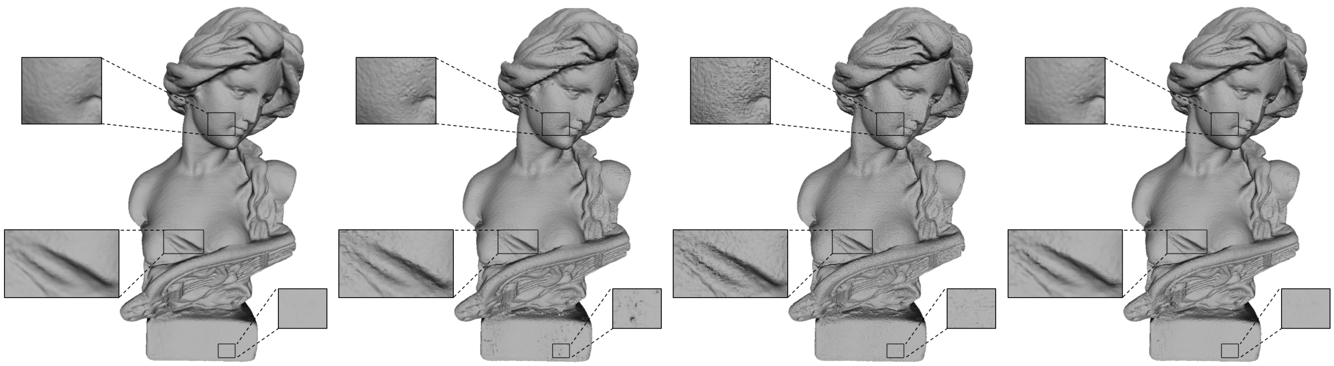
Gauss Surface Reconstruction



(a) *PR* ($|v| = 1.4$) ($t_1 = 105.82$) **(b)** *SSD* ($|v| = 1.2$) ($t_1 = 37.34$) **(c)** *SPR* ($|v| = 1.6$) ($t_1 = 77.20; t_{10} = 15.78$) **(d)** *GR* ($|v| = 1.4$) ($t_1 = 79.46; t_{10} = 8.87$)

Figure 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 Gauss Reconstruction (GR). The Lady model is a real-world scanned data with 0.5 millions samples. $|v|$ denotes the number of vertices in millions of the reconstructed mesh, and t_1 and t_{10} denote the running time in seconds of the reconstructions with single thread and 10 threads, respectively.*

Abstract

In this paper, we present a surface reconstruction method. We follow the strategy of Poisson reconstruction to estimate the indicator function and then obtain a triangle mesh by extracting an isosurface. The key observation of this work is that the indicator function can be estimated directly from Gauss Lemma without solving any Poisson system. This direct approach leads to a simple and more accurate reconstruction method which we call Gauss reconstruction. More importantly, our Gauss reconstruction can be paralleled with little overhead and therefore very efficient. We apply our reconstruction to both synthetic data and real-world scanned data, and demonstrate the accuracy, the robustness and the efficiency of our method. In addition, we compare its performance with that of several state-of-art methods.

CR Categories: I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling;

Keywords: Surface Reconstruction, Point Cloud, Gauss Lemma, Double Layer Potential

1 Introduction

Surface Reconstruction has been studied for more than three decades. Many elegant methods are available. Among them, Poisson reconstruction [Kazhdan et al. 2006] or its variant [Kazhdan and Hoppe 2013] is one of the most popular methods. The basic idea of Poisson reconstruction is to estimate the indicator function χ of a region in $\mathbb{R}^3$ and then extract a triangle mesh by isosurfacing χ to approximate the boundary of this region. In Poisson reconstruction, one needs to solve a Poisson equation in order to estimate χ. We observe that the indicator function χ can be estimated using an explicit integral formula based on the fundamental solution to Laplace equation, which in fact is given in the well-known Gauss Lemma in the potential theory (e.g.,[Wendland 2009]).

However, it is a non-trivial task to turn Gauss Lemma to an efficient and accurate reconstruction method. The singularity of the integral kernel and the discontinuity of the indicator function affects the accuracy of the reconstruction, and the globalness of the integral formula makes the algorithm quite slow. We propose an approach called *disk integration* to address the singularity issue, a smoothing scheme to solve the discontinuity issue. To improve the efficiency, we use the well-known fast multipole method [Greengard and Rokhlin 1987] to estimate the indicator function.

Our Gauss reconstruction algorithm inherits many nice properties of Poisson reconstruction, including its robustness against noise and missing data, and its being free of spurious surface sheets away from the input samples. Furthermore, our direct approach of estimating the indicator function without solving any linear system makes the reconstruction algorithm simple and accurate. More importantly, our Gauss reconstruction has a natural parallel implementation and the overhead of this implementation is almost negligible.

Figure 1 shows the comparison of our Gauss reconstruction with several state-of-art methods on the Lady model. All reconstructions are computed using an octree of the maximum depth 10. From Figure 1, we can see that our Gauss reconstruction generates a good quality reconstruction of the Lady model: it preserves the details while avoid overfitting the input samples. In addition, the parallel implementation of our Gauss reconstruction only has little overhead.

2 Related Work

Surface reconstruction from point cloud has attracted great attention in the past thirty years, both in theory and in practice. Many algoirthms have been proposed. We give a brief review to those relavant to our work. There are two main categories: combinatorial algorithms and implicit algorithms.

Combinatorial methods take (part of) input sample points as vertices and reconstruct output meshes by determining the connectivity of input samples. Many of them are based on 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 good theoretical results, however in practice are senstive to noise and may produce jagged surfaces. In [Kolluri et al. 2004], a spectral based approach is proposed to smooth the surface. More recently, in [Xiong et al. 2014], a learning approach is proposed to treat geometry and connectivity reconstruction as one joint optimization to improve reconstruction quality.

Implicit methods attempt to estimate implicit functions from input samples, and extract iso-surfaces to generate triangle meshes. Poisson reconstruction and its variant [Kazhdan et al. 2006; Kazhdan and Hoppe 2013] are most revalant to our work, which esitmate indicator functions of unknown models. In [Muraki 1991; Walder et al. 2005], Radial Basis Functions (RBFs) are used as bases for defining implicit functions, where coefficients of bases are determined by fitting input data. Since RBFs are global, fast multipole methods (FMM) are employed to improve the efficiency [Carr et al. 2001]. The signed distance function is a natural choice as implicit function for surface reconstruction, where implicit function can be estimated either locally as distances to tangent planes of nearby samples [Hoppe et al. 1992; Curless and Levoy 1996] or globally by minimizing the fitting error [Calakli and Taubin 2011]. Finally, in [Amenta and Kil 2004; Dey and Sun 2005; Levin 1998], moving least squares (MLS) is used to define implicit surfaces, which are extremal sets of certain energy. It is associated with a nice projection operator which can be used for surface smoothing. The surfaces reconstructed by implicit methods often do not interpolate input samples, and therefore are smoother than those reconstructed by combinatorial methods.

For iso-surface extraction, marching cubes [Lorensen and Cline 1987] and its adaptation to octree [Wilhelms and Van Gelder 1992] are the most popular methods. Delaunay refinement based methods [Boissonnat and Oudot 2005] produce good quality triangle meshes, though they are less efficient and difficult to parallelize.

3 Gauss Reconstruction

Our problem can be stated as follows: the input data S is a set of oriented points $S = \{s_1, s_2, ..., s_n\}$, each consisting of a position $s.p$ and an outward normal $s.\vec{N}$, sampling the boundary $\partial\Sigma$ of an unknown region $\Sigma \in \mathbb{R}^3$, i.e., $s.p$ lies on or near the surface and $s.\vec{N}$ approximates the surface normal near the position $s.p$. Our goal is to reconstruct a triangle mesh approximating the boundary $\partial\Sigma$. Assume the region Σ satisfies certain regularity which often holds for 3D models in computer graphics. We follow Kazdhan et al. [Kazhdan et al. 2006] to estimate the indicator function χ of the region Σ and extract an appropriate isosurface. However, unlike [Kazhdan et al. 2006] where the indicator function is computed by solving a Poisson equation, our method estimates the indicator function using the following explicit integral formula, which is given in the well-known Gauss Lemma in the potential theory [Wendland 2009].

Lemma 3.1 (Gauss Lemma). *Let Σ be an open region in $\mathbb{R}^3$. Consider the following double layer potential: for any $x \in \mathbb{R}^3$*

$$\chi(x) = \int_{\partial\Sigma} \frac{\partial G}{\partial \mathbf{n}_y}(x,y)\mathrm{d}\tau(y), \tag{1}$$

where $\mathbf{n}_y$ is the outward normal of $\partial\Sigma$ at y, $\mathrm{d}\tau(y)$ is the surface area form of $\partial\Sigma$ at y, and G is the fundamental solution of the Laplace equation in $\mathbb{R}^3$, which can be written explicitly as:

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

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

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

Note that
$$\frac{\partial G}{\partial \mathbf{n}_y}(x,y) = -\frac{1}{4\pi}\frac{(x-y)\cdot\mathbf{n}_y}{\|x-y\|^3},$$
which we call the kernel function, denoted $K(x,y)$. Given the samples S, the indicator function conceptually can be estimated directly by the following summation

$$\chi(x) \approx -\frac{1}{4\pi}\sum_{s\in S}\frac{(x-s.p)\cdot s.\vec{N}}{\|x-s.p\|^3}s.A. \tag{4}$$

where $s.A$ is the surface area of the sample s, whose estimation will be described in Section 3.1. Our approach is direct and very simple. Note that the estimation of the indicator function χ at different points x is completely independent to each other, which leads to a natural parallel algorithm.

The integral formula (1) has many good properties. Nevertheless, to make our reconstruction practically useful and efficient, we need to address the following three issues.

(i) Singularity of the kernel function: Notice that the kernel function $K(x,y)$ becomes singular when x is approaching y. Based on the summation formula (4), to accurately evaluate the indicator function χ at the points close to the surface $\partial\Sigma$, one needs a very dense sampling of the surface, which becomes practically not plausible. To address this issue, we propose an approach called *disk integration*, where we associate each sample point s a disk to approximate the surface locally around the position $s.p$, and use the integral over the continuous disk domain, instead of over the discrete samples, to approximate the integral over the surface. See Section 3.1 for a detailed description. With disk integration, we are able to accurately estimate the indicator function χ even with a sparse sampling. For example, as shown in Figure 2, the indicator function of unit sphere can be accurately estimated from 1000 samples using disk integration so that the reconstructed surface is within 5×10^{-3} Hausdorff distance to unit sphere.

(ii) Globalness of the integral formula: Note the estimation of $\chi(x)$ using the integral formula (1) is global, i.e., one has to integrate the kernel function $K(x,y)$ over the entire surface $\partial\Sigma$ to obtain a correct estimation of $\chi(x)$. In particular, one can not perform thresholding based on the value of $K(x,y)$ and skip integrating the region where $K(x,y)$ is small. To see this, imagine Σ is a ball of radius r, and x is the center of the ball. For $y \in \partial\Sigma$, $K(x,y)$ can be made arbitrarily small by choosing the radius r large enough. However, $\chi(x)$ remains the constant 1, independent of r. Therefore, to estimate χ at m different locations, a native implementation requires at least $O(mn)$ operations. Recall that n is the number of samples in S. Fortunately, the kernel function $K(x,y)$ over two distant regions can be well-approximated by a constant function. This enables us to speed up the estimation of χ by using the well-known *fast multipole method (FMM)*. In the paper, we employ a simple FMM based on octree, see Section 3.2. This improves to $O(m + n\log n)$ the complexity for estimating χ at m points. Note there exists FMM

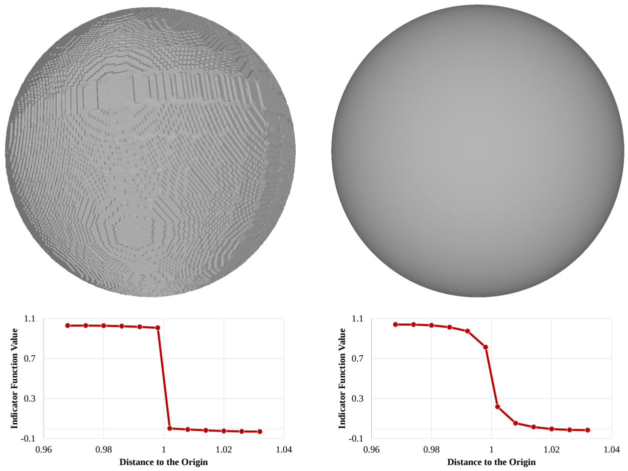


Figure 2: *Left column: The reconstruction from the indicator function. The top shows the resulting mesh and the bottom shows the indicator function around the north pole restricted to the diameter passing the north pole. Right column: The reconstruction from the smoothed indicator function. The top shows the resulting mesh and the bottom shows the smoothed indicator function around the north pole restricted to the diameter passing the north pole.*

which can improve the complexity to the linear order $O(n+m)$, which though is more involved and we will investigate in the future.

(iii) Discontinuity of the indicator function: Once the above two issues are addressed, the indicator function χ can be evaluated faithfully and efficiently. The resultant triangle mesh by isosurfacing χ, denoted M, lies in a small tubular neighborhood of the surface $\partial\Sigma$, namely the Hausdorff distance between M and $\partial\Sigma$ is small. However, since the function χ is discontinuous at $\partial\Sigma$, the normal of a triangle in M may not approximate the normals of $\partial\Sigma$ at the points close to the triangle, see Figure 2. To address this issue, we *smooth the indicator function* near the surface $\partial\Sigma$ by properly modifying the kernel function. See Section 3.3 for more details. Figure 2 shows the triangle mesh obtained by isosurfacing the smoothed indicator function of unit sphere, which becomes smooth.

3.1 Disk Integration

Recall that the input data S samples the surface $\partial\Sigma$. Imagine that each sample point $s \in S$ represents a neighboring region on $\partial\Sigma$, denoted $s.V$, so that the set $\{s.V\}_{s\in S}$ decomposes the surface $\partial\Sigma$. One can think of $s.V$ as the Voronoi region of s on $\partial\Sigma$. Then $\chi(x) = \sum_{s\in S} C(x,s)$ where

$$C(x,s) = \int_{s.V} K(x,y)\mathrm{d}\tau(y). \tag{5}$$

Note that $s.V$ is unknown and we use a disk perpendicular to $s.\vec{N}$ to approximate $s.V$. The radius of this disk is estimated as the average distance to the k-nearest samples in S. In this paper, we fix $k = 10$ for all samples. We denote this disk $s.D$, its radius $s.r$, and take the area of $s.D$ as the surface area $s.A$.

We approximate $C(x,s)$ using $\int_{s.D} K(x,y)\mathrm{d}y$. Note that even over the simple domain $s.D$, the above integration can not be calculated explicitly. Our strategy is to approximate $s.D$ using k layers of partial annuli (See the shaded regions in Figure 3), over each of which the integration of the kernel function $K(x,y)$ can be calculated analytically. Let x' be the projection of x on the plane

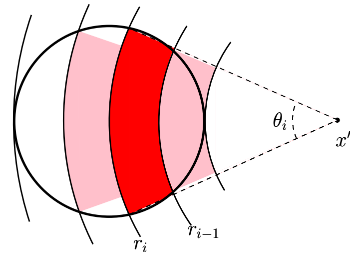


Figure 3: *Illustration of integral domain (shaded region) of disk integration.*

containing $s.D$. Denote $C(r)$ the circle centered at x' of the radius r, and $A(r,R)$ the annulus centered at x' of the inner radius r and the outer radius R. Let $r_0 = \min_{y\in s.D}\|x'-y\|$ and $r_k = \max_{y\in s.D}\|x'-y\|$, and $r_i = r_0 + \frac{i(r_k-r_0)}{k}$, for $0 \le i \le k$. r_0 is 0 if x' is in the disk. Let θ_i be the central angle of the arc $C(r_i) \cap s.D$, and F_i be the fan spanned by the same arc. The partial annulus at the ith layer is $F_i \cap A(r_{i-1}, r_i)$. Set $d = \|x - x'\|$. Then $C(x,s)$ is approximated by $DI(x,s) = \sum_{1\le i\le k} c_i$ where

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

In the paper, we fix the number of layers $k = 20$.

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

Figure 2 shows the indicator function of unit sphere restricted to points passing the center estimated using the above approach from 1000 random samples. The Hausdorff distance between the reconstructed triangle mesh and the original sphere is less than 5×10^{-3}.

3.2 Fast Multipole Method

In this subsection, we describe an implementation of FMM for speeding up the estimation of the indicator function χ. An octree is employed as the multi-resolution data structure in FMM and the same octree is also used for isosurfacing χ.

Given a set of samples S and a maximum tree depth D, the octree is the minimal octree so that each sample falls into a leaf node of depth D. For a non-uniform sampling, we follow [Kazhdan et al. 2006] and reduce the depth for the samples in the sparse regions. We denote $\mathcal{O}$ the resultant octree, and $\mathcal{V}$ set of grid vertices of the octree $\mathcal{O}$. Our goal is to evaluate the indicator function at $\mathcal{V}$. Now consider the cubes $\{B_i^k\}_i$ of $\mathcal{O}$ at depth k, see Figure 4. A cube B_i^k may be half open, i.e., does not contains the faces with the maximum x, or y, or z coordinate, unless they are on the boundary.

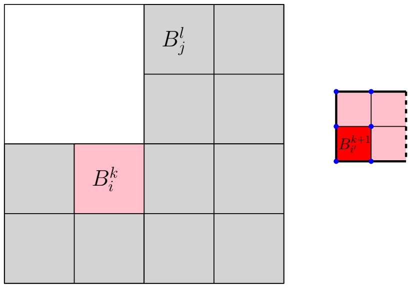


Figure 4: *The cubes $\mathcal{O}$ at depth k may not cover the entire domain due to the adaptivity of $\mathcal{O}$. The red cube $B_{i'}^{k+1}$ is a subcube of the pink cube B_i^k. The blue dots in B_i^k form set of grid vertices $\mathcal{V}_i^k$ in B_i^k.*

See the pink cube in Figure 4. Let $\mathcal{V}_i^k = \mathcal{V} \cap B_i^k$ (See the blue dots in B_i^k in Figure 4), and $S_i^k = S \cap B_i^k$. For a set X, denote $|X|$ the cardinality of X. Let $\bar{v}_i^k$ be the representative grid of B_i^k defined by

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

and $\bar{s}_i^k$ be the representative sample of B_i^k defined by

$$\begin{aligned} \bar{s}_i^k.p &= \frac{\sum_{s \in S_i^k} s.A \cdot s.p}{\sum_{s \in S_i^k} s.A}, \\ \bar{s}_i^k.\vec{N} &= \frac{\sum_{s \in S_i^k} s.A \cdot s.\vec{N}}{\sum_{s \in S_i^k} s.A}, \text{and} \\ \bar{s}_i^k.A &= \sum_{s \in S_i^k} s.A. \end{aligned}$$

The disk $\bar{s}_i^k.D$ is centered at $\bar{s}_i^k$, perpendicular to $\bar{s}_i^k.\vec{N}$, and of the area $\bar{s}_i^k.A$. Let a_k be the side length of the cubes at depth k. The basic idea of our implementation of FMM is as follows. We start with the cube at depth 1. In general, consider two cubes B_i^k and B_j^l at depth l and depth k respectively. Note that B_i^k and B_j^l may be the same cube. If $\|\bar{s}_i^k - \bar{v}_j^l\| \geq c a_k$, then for any grid vertex $v \in \mathcal{V}_j^l$, we approximate $\sum_{s \in S_i^k} C(v, s)$ using $C(\bar{v}_j^l, \bar{s}_i^k)$. Otherwise, we repeat the above procedure for any pairs of subcubes, one in B_i^k and the other in B_j^l until both are leaf nodes. Only when both are leaf nodes do we indeed estimate $C(v, s)$ for an individual sample $s \in S_i^k$ and an individual grid vertex $v \in \mathcal{V}_j^l$. In the paper, we fix the constant $c = \sqrt{2}$. Pseudocode 1 shows our FMM implementation.

3.3 Smooth the Indicator Function

In this subsection, we describe a way to smooth the indicator function to obtain a smooth reconstruction. Our strategy is to modify the kernel function. For a point $x \in \mathbb{R}^3$, we associate a width $x.w$ and modify the kernel function $K(x, y)$ for any $y \in \partial\Sigma$ as follows.

$$\tilde{K}(x, y) = \begin{cases} K(x, y), & \|x - y\| \geq x.w, \\ -\dfrac{(x-y) \cdot \mathbf{n}_y}{4\pi (x.w)^3}, & \|x - y\| < x.w. \end{cases} \quad (6)$$

The smoothed indicator function $\tilde{\chi}(x) = \int_{\partial\Sigma} \tilde{K}(x, y) \mathrm{d}\tau(y)$. Note that $\tilde{K}(x, y)$ remains the same as $K(x, y)$ for any $y \in \partial\Sigma$ with

function FMM(B_i^k, B_j^l, $f : \mathcal{V} \to \mathbb{R}$)
  if $\|\bar{s}_i^k - \bar{v}_j^l\| \geq c a_k$ **then**
    evaluate $e \approx C(\bar{v}_j^l, \bar{s}_i^k)$
    $f(v) = f(v) + e$ for any $v \in \mathcal{V}_j^l$.
  else
    if both B_i^k and B_j^l are leaves **then**
      for all $s \in S_i^k$ and $v \in \mathcal{V}_j^l$ **do**
        evaluate $e \approx C(v, s)$
        $f(v) = f(v) + e$;
      end for
    else if Neither B_i^k nor B_j^l is a leaf **then**
      for all $B_{i'}^{k+1} \subset B_i^k$ and $B_{j'}^{l+1} \subset B_j^l$ **do**
        FMM($B_{i'}^{k+1}$, $B_{j'}^{l+1}$, f)
      end for
    else if B_i^k is a leaf and B_j^l is not a leaf **then**
      for all $B_{j'}^{l+1} \subset B_j^l$ **do**
        FMM(B_i^k, $B_{j'}^{l+1}$, f)
      end for
    else
      for all $B_{i'}^{k+1} \subset B_i^k$ **do**
        FMM($B_{i'}^{k+1}$, B_j^l, f)
      end for
    end if
  end if
end function

Pseudocode 1: *FMM.*

$\|x - y\| \geq x.w$, and hence $\tilde{\chi}(x) = \chi(x)$ for any x with $d(x, \partial\Sigma) \geq x.w$.

To see $\tilde{\chi}(x)$ at a point x with $d(x, \partial\Sigma) < x.w$, we consider a simplified case where the surface $\partial\Sigma$ is simply a plane. Let $d(x)$ be the signed distance from x to $\partial\Sigma$. In this simplified case, we have $d(x) = (x - y) \cdot \mathbf{n}_y$, for any $y \in \partial\Sigma$. Let $B_x(r)$ be the ball in $\mathbb{R}^3$ centered at x and of radius r. Then we have

$$\begin{aligned} \tilde{\chi}(x) &= \int_{B_x(x.w) \cap \partial\Sigma} \tilde{K}(x, y) \mathrm{d}\tau(y) + \int_{\partial\Sigma \setminus B_x(x.w)} K(x, y) \mathrm{d}\tau(y) \\ &= -\frac{d(x)}{4(x.w)^3}((x.w)^2 - d^2) - \frac{d(x)}{2(x.w)} \\ &= -\frac{3d(x)}{4(x.w)^3} + \frac{d^3(x)}{(x.w)^3}. \end{aligned}$$

Therefore, when $d(x)$ is small, i.e., the point x is close to the surface, $\tilde{\chi}(x)$ is dominated by a linear function of the signed distance $d(x)$, which is very desirable for extracting isosurface [Calakli and Taubin 2011].

It remains to specify the width $x.w$. 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 set of the leaf nodes in $\mathcal{O}$ having v as one of their vertices. Set $v.w$ to be β times the side length of the smallest cube in $v.B$, where β is a constant which we call width coefficient. Define the neighboring vertices $v.\mathcal{V}$ of v in the octree so that a grid vertex u is in $v.\mathcal{V}$ if u and v are connected by an edge of a cube in $v.B$. It is possible that $v.w$ and $u.w$ differ a lot even when u and v are neighbors, and the resultant function $\tilde{\chi}$ may become rough. To address this issue, we further smooth $v.w$ by averaging the widths over the neighbors, namely set

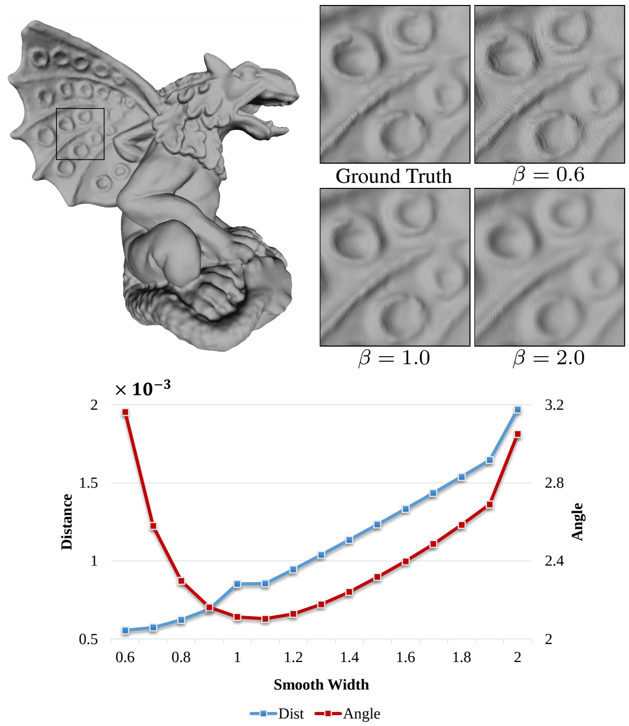


Figure 5: *Choice of width coefficient. The first row shows visual effects; the second row shows the average position error (Dist) and the average angle error using the reconstruction benchmark [Berger et al. 2013].*

function GaussRecon(S: samples, D: maximum depth, β: width coefficient)
  Estimate $s.r$ for each sample $s \in S$
  Given D, construct an adaptive octree $\mathcal{O}$
  Compute representative samples $\bar{s}$ for all cubes in $\mathcal{O}$.
  Compute representative grid vertices $\bar{v}$ for all cubes in $\mathcal{O}$.
  Given β, estimate $v.w$ for each grid vertex $v \in \mathcal{V}$
  Initialize $f : \mathcal{V} \to \mathbb{R}$ to be zero.
  Call **FMM**(B_1^1, B_1^1, f).
  Set the isovalue as the median of f.
  Extract the isosurface M using marching cube over $\mathcal{O}$.
  Output M.
end function

Pseudocode 2: *GaussRecon*

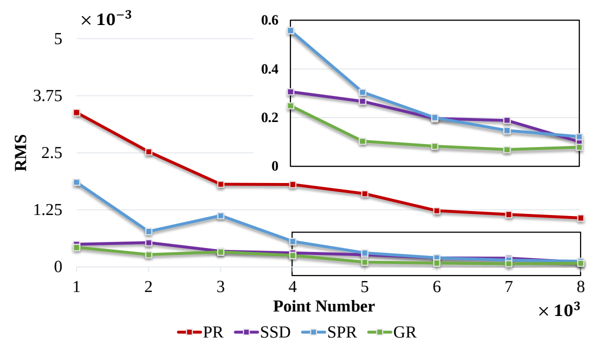


Figure 6: *The average error RMS of the reconstructions by different methods. The sub-figure on top-right is the zoom-in on the marked box.*

$$v.w = \frac{\sum_{u \in v.\mathcal{V}} u.w}{|v.\mathcal{V}|},$$

and repeat this averaging step for k times. In the paper, we set $k = 20$.

Note that although the smoothed kernel function $\tilde{K}$ is not singular, disk integration can still be employed for $\tilde{K}$ and significantly improves the accuracy of estimating $\tilde{\chi}$. Moreover, when we invoke the estimation of $C(\bar{v}, \bar{s})$ for a representative grid vertex $\bar{v}$ and a representative sample $\bar{s}$, we assume that $\bar{v}$ and $\bar{s}$ are far away to each other and compute $DI(\bar{v}, \bar{s})$ or $DC(\bar{v}, \bar{s})$ using the kernel function K. Therefore, there is no need to associate a width to a representative grid vertex $\bar{v}$.

The width coefficient provides a way to control the trade-off between the position accuracy and the smoothness of the reconstruction. See Figure 5. The bigger the β is, the smoother but less accurate in position of the reconstructed surface. Of course, if β is chosen too big, both position accuracy and angle accuracy decreases. A typical value of β is set to be 1.

Finally, we summarize our Gauss reconstruction in Pseudocode 2.

3.4 Parallel Implementation

For the grid vertices v, the estimation of the indicator function $\chi(v)$ is independent to each other, which leads a straightforward parallel implementation. In particular, we open new threads to execute the calls of $FMM(B_i^k, B_j^l, f)$ with $k, l \le c$. The parameter c is chosen so that we have just enough threads so that the load on each core is balanced and at the same time the overhead of multi-threads is minimized. In the paper, we set $c = 5$.

4 Results

In this section, we evaluate our Gauss reconstruction (GR) in terms of accuracy, robustness, and efficiency, and compare its performance to that of the state-of-art methods, including Poisson reconstruction [Kazhdan et al. 2006] (PR) and its variant screened Poisson reconstruction [Kazhdan and Hoppe 2013] (SPR), and smooth signed distance reconstruction [Calakli and Taubin 2011] (SSD). Note that we compare with the most recent implementation of these methods available online. In particular, using the most recent implemenation, the performance of SSD improves a lot comparing to those reported in [Kazhdan and Hoppe 2013]. We follow [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, 0.25 for Hessian, and set the data fitting weight $\alpha = 4$ in SPR. Unless we state explicitly using other values, we by default set the maximum depth $D = 10$ for octree construction in all methods and the width coefficient $\beta = 1$ in our Gauss reconstruction. All the experiments are performed on a Windows 7 workstation with an Intel Xeon E5-2690V3 CPU @2.6GHz.

4.1 Accuracy

First, we consider the reconstruction of unit sphere from samples where the accurate ground truth is known. We generate 1000 to 8000 samples according to a Gaussian mixture of eight Gaussian in $\mathbb{R}^3$ and then radially project them into unit sphere. We use the average error RMS to measure the quality of the reconstructed surface.

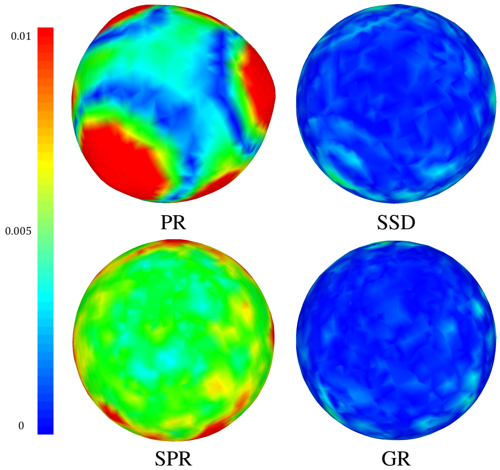


Figure 7: *The reconstructed unit sphere from* 1000 *random samples. The color illustrates the RMS (relative to the bounding box diaganol) error distribution: small error in blue and big error in red.*

Figure 6 shows the error statistics of the reconstructions by different methods. Our Gauss reconstruction performs the best and Poisson reconstruction has the largest error. For 1000 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 Poisson reconstruction is visually not round.

Next, we consider general models. To estimate the numerical accuracy of the reconstruction results, we follow the same strategy as in [Berger et al. 2013], we first sample points from a known mesh, or simply take its vertices, and then reconstruct surfaces with 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 both outperform PR and SSD.

We also apply the reconstruction methods to the data from the reconstruction benchmark [Berger et al. 2013]. Due to the limited space, we only report the results on four data sets: 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 9a, we can see that PR and GR generate visually similar results while SPR and SSD produces extra spurious sheets near the surface. However, the accuracy of GR is much better than PR. Figure 9b and 9c show the average angle error and the average position error, respectively. For this set of examples, overall, PR performs the best in angle accuracy but the worst in position accuracy, and SSD performs the best in position accuracy. However, from Figure 9a, SSD may overfit the data. Our GR seems achieving a better balance between position accuracy and angle accuracy.

4.2 Noise Resilience

In this subsection, we test our Gauss reconstruction over the noisy data including both synthetic Gaussian noise and real-world scan data with noise and possibly missing data, and compare the performance of different reconstructions.

Synthetic Noise In this example, we add to the Armadillo model the different levels of noise by perturbing the positions of the samples according to Gaussian distribution of different variances.

Figure 10b shows the reconstructed surfaces by our Gauss reconstruction from the samples perturbed by a Gaussian with variance equal to 0.005 times the diagonal of the bounding box. Figure 10c show the details of reconstructions at different noisy levels by zooming in on the region marked in Figure 10b. SPR and SSD apparently overfit the data and therefore sensitive to noise and reconstruct bumpy surfaces. PR always produces smooth reconstructions, whose accuracy however is the lowest. See Figure 10a. The surfaces reconstructed by our Gauss reconstruction are also smooth, and at the same time preserve more details, and therefore more accurate.

Real-world Scanned Data We apply the reconstruction methods to the sampling obtained by scanning several real-world models using Konica-Minolta Vivid 9i Laser Scanner. The obtained sampling contains noise and missing data, and are highly non-uniform. See the first column in Figure 11. In these examples, we set the width coefficient $\beta = 2$ in our Gauss reconstruction. Visually, the reconstructions generated by PR and GR are comparable, and have better quality than those by SSD and SPR, which again obviously overfit the data.

4.3 Efficiency

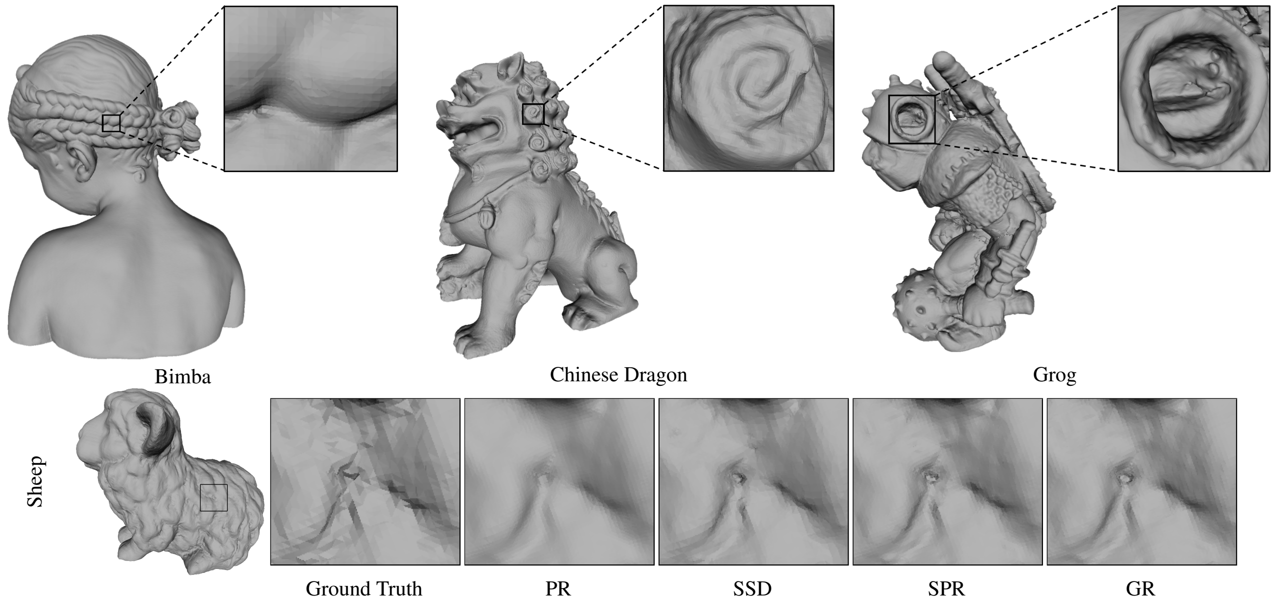
In this subsection, we show the efficiency of our Gauss reconstruction, in particular its parallel implementation. The running time shown in Table 1 excludes the time for data input/output.

As Table 1 shows, Poisson reconstruction (version 3.0) is the slowest method among four reconstructions. In the single thread implementation, SSD (version 3.0) is the fastest mainly due to the employment of hash octree, and our Gauss reconstruction is comparable to that of screened Poisson reconstruction (version 8.0). Note that the current implementation of PR, SPR and GR does not use hash octree.

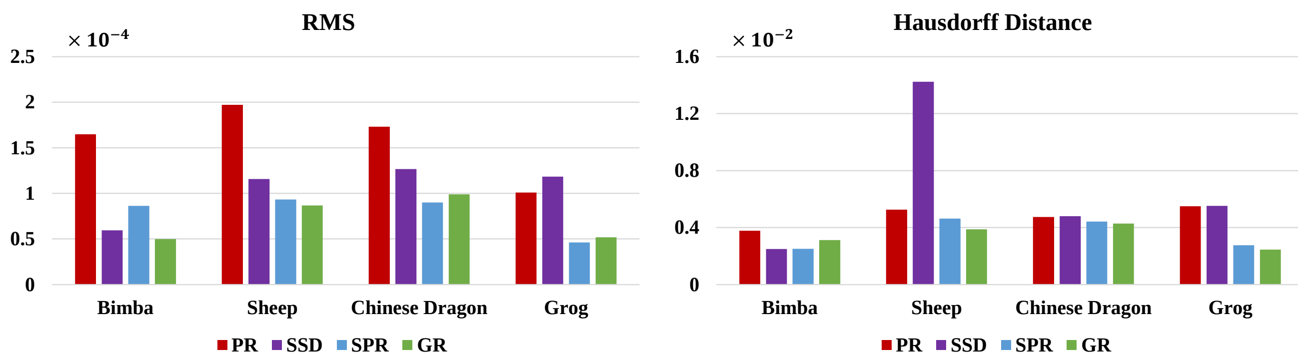
For the multi-threads implementation, we can see from Table 1, the parallel implementation of our Gauss reconstruction has almost negligible overhead and achieves a nearly perfect linear speedup. In Table 1, we also show the running time of the parallel implementation of screened Poisson reconstruction, which is available to the public. GR is about twice as fast as SPR.

Model	Cores	Time in Seconds			
		PR	SSD	SPR	GR
Grog	1	178.68	59.44	133.68	127.23
$\|v\| = 0.8$	10	–	–	27.48	14.96
Bimba	1	62.19	35.04	73.15	42.31
$\|v\| = 0.5$	10	–	–	15.46	5.59
Pig	1	169.64	58.16	116.69	122.62
$\|v\| = 0.9$	10	–	–	20.93	13.90
Child	1	135.51	50.44	105.24	95.64
$\|v\| = 0.7$	10	–	–	18.67	9.94

Table 1: *Running time on different models.* $|v|$ *denotes the number of vertices in millions of input point cloud.*



(a) *Reconstructed meshes.*



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

Figure 8: *The accuracy illustration. The running time: Bimba($|v| = 0.50$, PR: 62.20s, SSD: 35.04s, SPR: 73.15s, GR: 42.31s), Sheep($|v| = 0.16$, PR: 31.66s, SSD: 22.38s, SPR: 24.99s, GR: 18.60s), Chinese dragon($|v| = 0.66$, PR: 109.43s, SSD: 44.28s, SPR: 96.02s, GR: 77.89s), Grog($|v| = 0.88$, PR: 178.68s, SSD: 59.44s, SPR: 133.68s, GR: 127.23s). The number of samples is in millions.*

4.4 Reconstruction of details

Finally, we show two more reconstructions to demonstrate that our Gauss method can reconstruct very detailed features. We set the maximum depth $D = 11$ to recover small features. Figure 12 shows the reconstruction result of the Raptor model and Figure 13 shows the reconstruction result of the Statuette model. As we can see, comparing to the ground truth, our Gauss reconstruction can reconstruct very detailed features.

5 Conclusions

We have presented a surface reconstruction method called Gauss reconstruction where the indicator function is estimated directly based on Gauss lemma without solving any linear system. This direct approach makes our Gauss reconstruction simple, accurate, and easy to parallel and therefore very efficient. In the future, we will consider the GPU implementation of FMM to further speed up the algorithm. In addition, we plan to study the theoretical property of Gauss reconstruction, in particular to analyze both position approximation error and normal approximation error.

References

AMENTA, N., AND KIL, Y. J. 2004. Defining point-set surfaces. *ACM Trans. Graph. 23*, 3 (Aug.), 264–270.

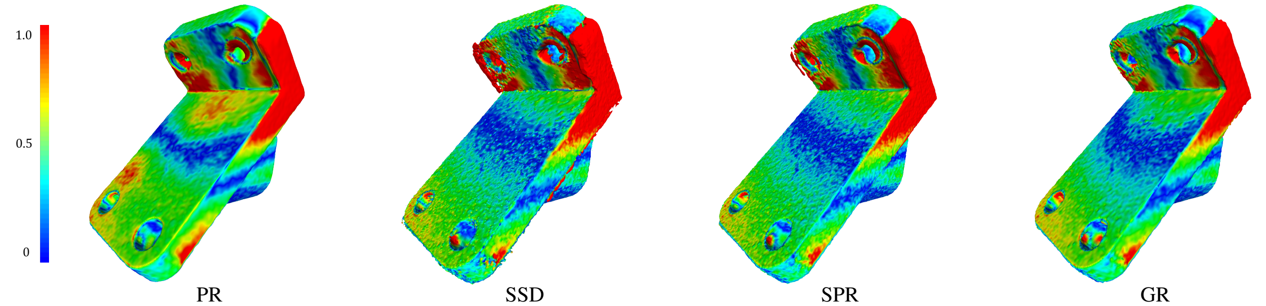
AMENTA, N., BERN, M., AND KAMVYSSELIS, M. 1998. A new voronoi-based surface reconstruction algorithm. In *Proceedings of SIGGRAPH '98*, ACM, 415–421.

AMENTA, N., CHOI, S., AND KOLLURI, R. K. 2001. The power crust, unions of balls, and the medial axis transform. *Computational Geometry 19*, 2, 127–153.

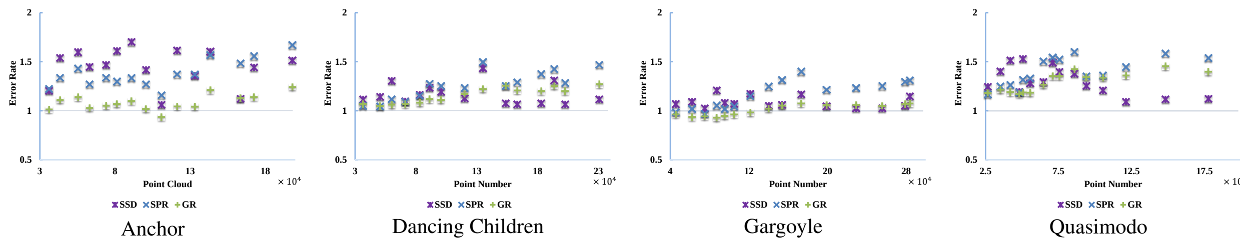
AMENTA, N., CHOI, S., DEY, T. K., AND LEEKHA, N. 2002. A simple algorithm for homeomorphic surface reconstruction. *Internat. J. Comput. Geom. & Applications 12*, 125–141.

BERGER, M., LEVINE, J. A., NONATO, L. G., TAUBIN, G., AND SILVA, C. T. 2013. A benchmark for surface reconstruction. *ACM Trans. Graph. 32*, 2 (Apr.), 20:1–20:17.

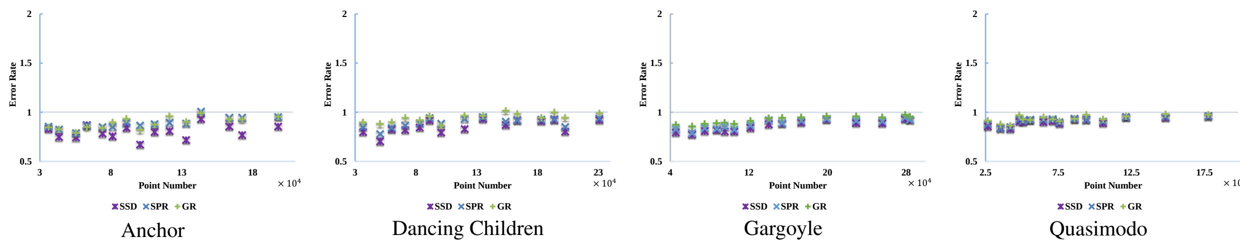
BOISSONNAT, J.-D., AND OUDOT, S. 2005. Provably good sampling and meshing of surfaces. *Graph. Models 67*, 5 (Sept.), 405–451.



(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).*

Figure 9: *Results from the reconstruction benchmark.*

CALAKLI, F., AND TAUBIN, G. 2011. Ssd: Smooth signed distance surface reconstruction. *Computer Graphics Forum 30*, 7, 1993–2002.

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. In *Proceedings of SIGGRAPH '01*, ACM, 67–76.

CIGNONI, P., ROCCHINI, C., AND SCOPIGNO, R. 1998. Metro: measuring error on simplified surfaces. *Computer Graphics Forum 17*, 2, 167–174.

CURLESS, B., AND LEVOY, M. 1996. A volumetric method for building complex models from range images. In *Proceedings of SIGGRAPH '96*, ACM, 303–312.

DEY, T. K., AND GOSWAMI, S. 2004. Provable surface reconstruction from noisy samples. In *Proceedings of SCG '04*, ACM, 330–339.

DEY, T. K., AND SUN, J. 2005. An adaptive mls surface for reconstruction with guarantees. In *Proceedings of SGP '05*, Eurographics Association, 43–52.

EDELSBRUNNER, H. 2003. Surface reconstruction by wrapping finite sets in space. In *Discrete and Computational Geometry*, vol. 25 of *Algorithms and Combinatorics*. Springer Berlin Heidelberg, 379–404.

GIESEN, J., AND JOHN, M. 2008. The flow complex: A data structure for geometric modeling. *Computational Geometry 39*, 3, 178–190.

GREENGARD, L., AND ROKHLIN, V. 1987. A fast algorithm for particle simulations. *Journal of computational physics 73*, 2, 325–348.

HOPPE, H., DEROSE, T., DUCHAMP, T., MCDONALD, J., AND STUETZLE, W. 1992. Surface reconstruction from unorganized points. In *Proceedings of SIGGRAPH '92*, ACM, 71–78.

KAZHDAN, M., AND HOPPE, H. 2013. Screened poisson surface reconstruction. *ACM Trans. Graph. 32*, 3 (July), 29:1–29:13.

KAZHDAN, M., BOLITHO, M., AND HOPPE, H. 2006. Poisson surface reconstruction. In *Proceedings of SGP '06*, Eurographics Association, 61–70.

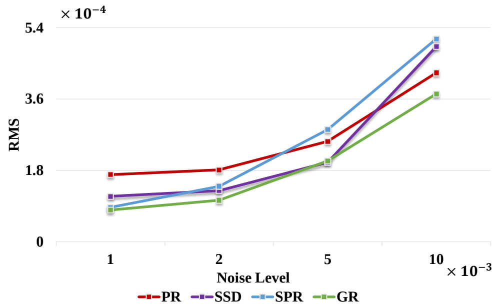
KOLLURI, R., SHEWCHUK, J. R., AND O'BRIEN, J. F. 2004. Spectral surface reconstruction from noisy point clouds. In *Proceedings of SGP '04*, ACM, 11–21.

LEVIN, D. 1998. The approximation power of moving least-squares. *Math. Computation 67*, 1517–1531.

LORENSEN, W. E., AND CLINE, H. E. 1987. Marching cubes: A high resolution 3d surface construction algorithm. In *Proceedings of SIGGRAPH '87*, ACM, 163–169.

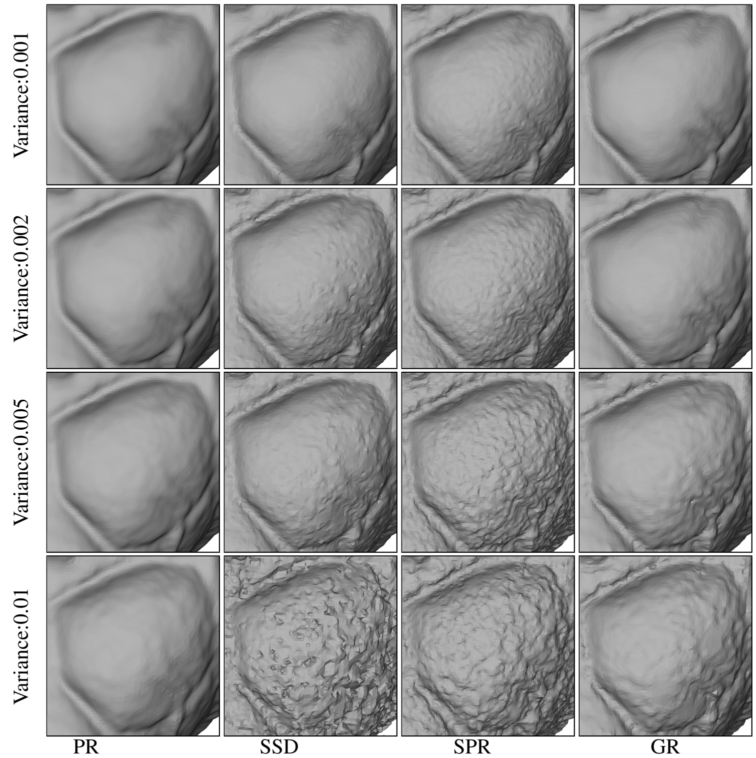
MURAKI, S. 1991. Volumetric shape description of range data using "blobby model". In *Proceedings of SIGGRAPH '91*, ACM, 227–235.

WALDER, C., CHAPELLE, O., AND SCHÖLKOPF, B. 2005. Implicit surface modelling as an eigenvalue problem. In *Proceed-*



(a) *RMS errors*

(b) *Reconstructed Armadillo by GR, Variance: 0.005*



(c) *Zoom-in on the left chest*

Figure 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.*

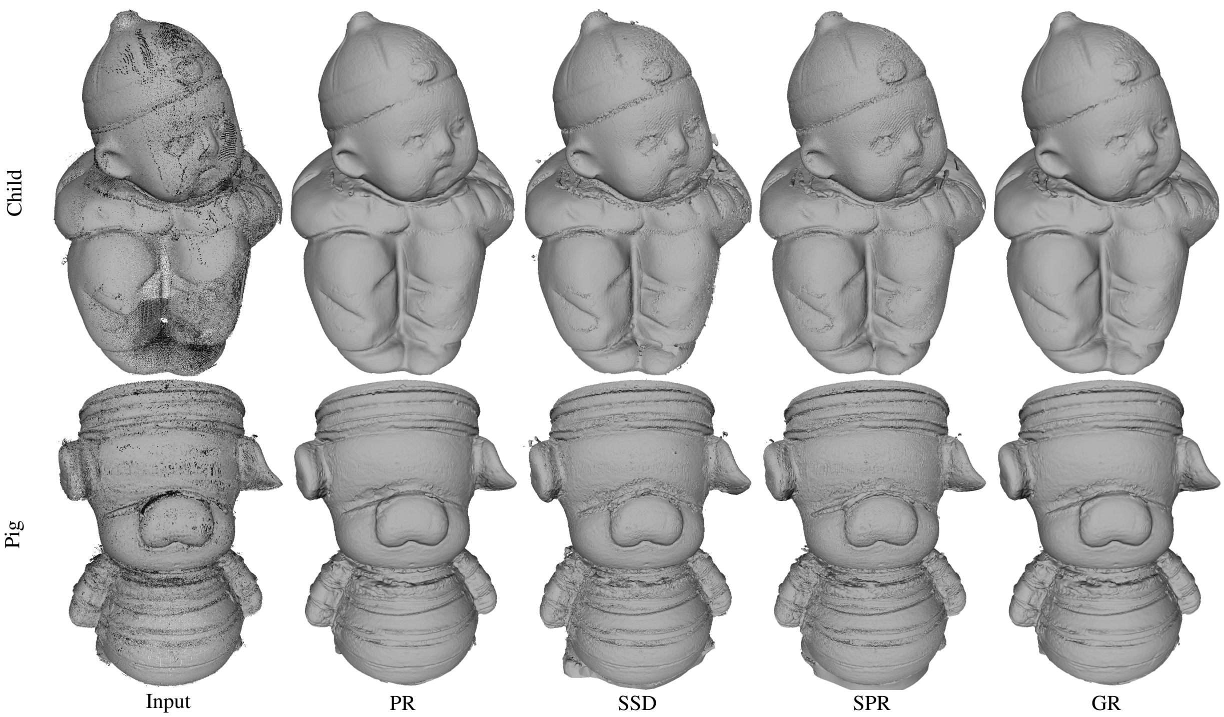


Figure 11: *The reconstructions of real-world scanned data.*

ings of ICML '05, ACM, 936–939.

WENDLAND, W. 2009. On the double layer potential. *Analysis, Partial Differential Equations and Applications: The Vladimir Mazya Anniversary Volume*, 319–334.

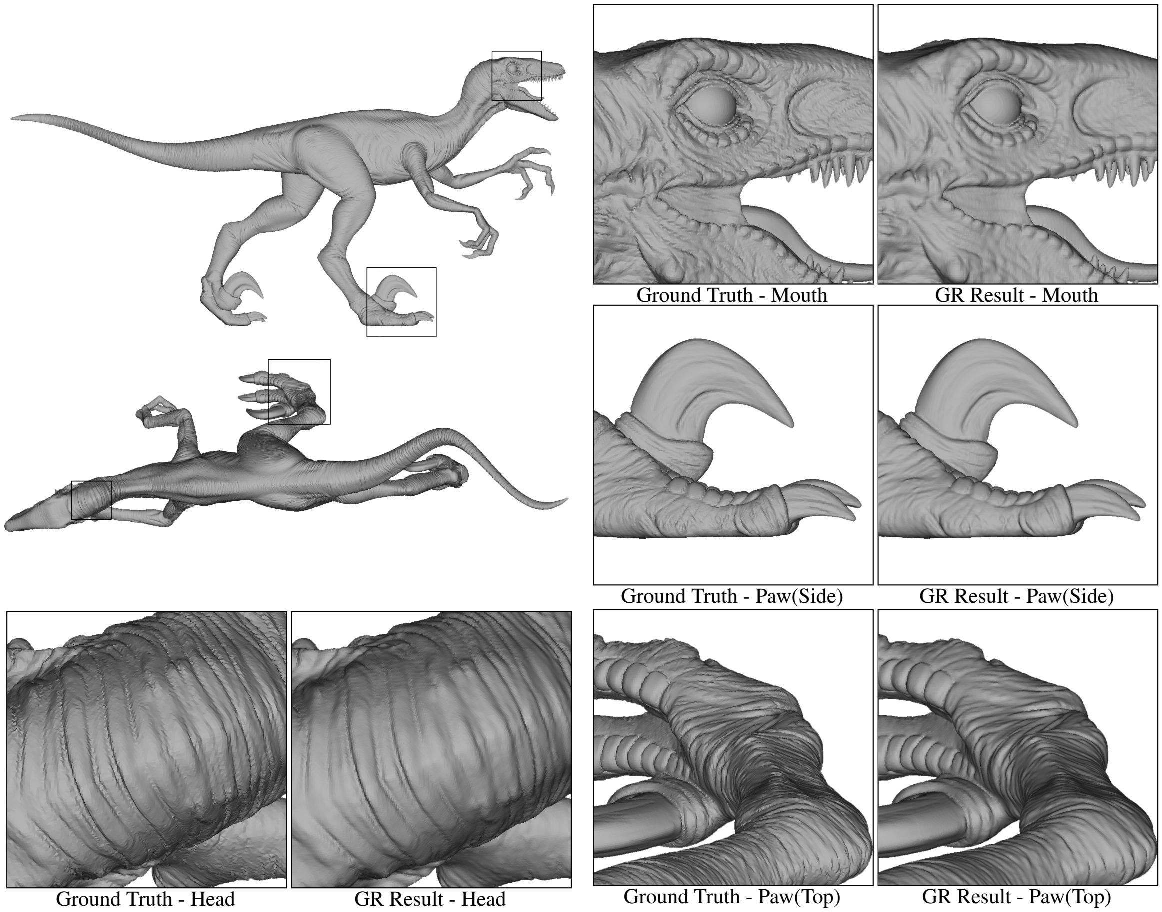


Figure 12: *Raptor Model with* 1 *million input samples. The running time is* $127.58s$ *with single thread and* $17.25s$ *with 10 threads.*

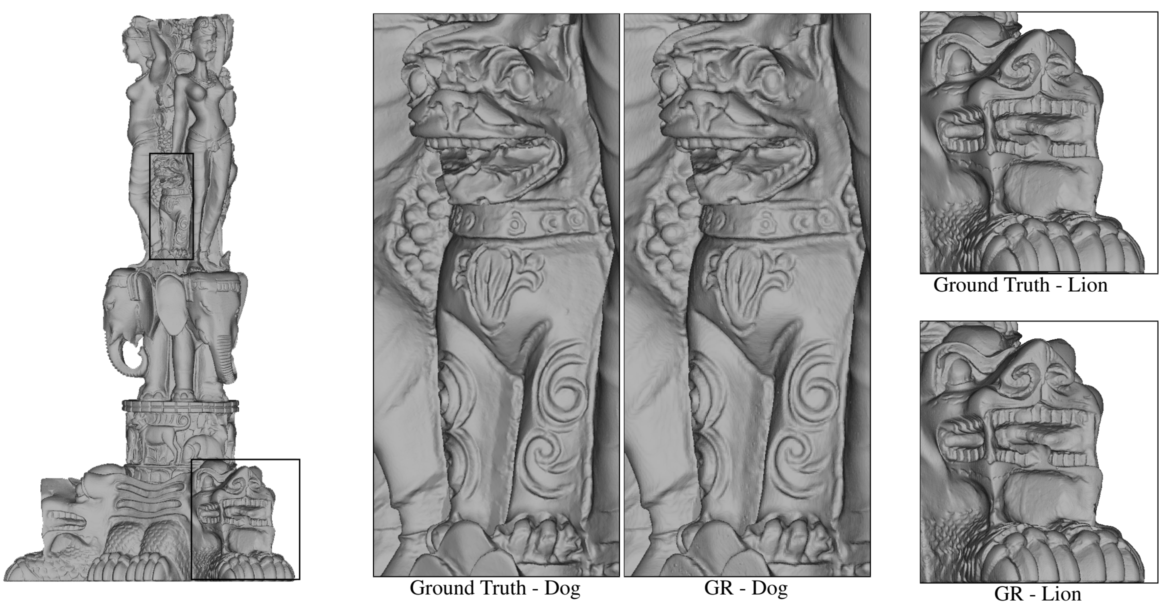


Figure 13: *Statuette Model with* 5 *million input samples. The running time is* $348.59s$ *with single thread and* $46.73s$ *with 10 threads.*

WILHELMS, J., AND VAN GELDER, A. 1992. Octrees for faster isosurface generation. *ACM Trans. Graph. 11*, 3 (July), 201–227.

XIONG, S., ZHANG, J., ZHENG, J., CAI, J., AND LIU, L. 2014. Robust surface reconstruction via dictionary learning. *ACM Trans. Graph. 33*, 6 (Nov.), 201:1–201:12.