

Robust Mesh Reconstruction from Unoriented Noisy Points*

Hoi Sheung Charlie C.L. Wang[†]

Department of Mechanical and Automation Engineering
The Chinese University of Hong Kong

Shatin, N.T., Hong Kong, China

hsheung@mae.cuhk.edu.hk; cwang@mae.cuhk.edu.hk

ABSTRACT

We present a robust method to generate mesh surfaces from unoriented noisy points in this paper. The whole procedure consists of three steps. Firstly, the normal vectors at points are evaluated by a highly robust estimator which can fit surface corresponding to less than half of the data points and fit data with multi-structures. This benefits us with the ability to well reconstruct the normal vectors around sharp edges and corners. Meanwhile, clean point cloud equipped with piecewise normal is obtained by projecting points according to the robust fitting. Secondly, an error-minimized subsampling is applied to generate a well-sampled point cloud. Thirdly, a combinatorial approach is employed to reconstruct a triangular mesh connecting the down-sampled points, and a polygonal mesh which preserves sharp features is constructed by the dual-graph of triangular mesh. Parallelization method of the algorithm on a consumer PC using the architecture of GPU is also given.

Categories and Subject Descriptors

I.3.5 [Computational Geometry and Object Modeling]: Boundary representations – Curve, surface, solid, and object representations

Keywords

Surface reconstruction, robust approach, noisy points, parallel computing, GPU

1. INTRODUCTION

The surface reconstruction from unoriented noisy point clouds has been problematic for more than decades in applications of computer graphics, virtual reality and CAD/CAM. At present, there are many 3D surface scanning devices available (using different sorts of methods like structured light,

stereo vision based scanners). These devices always generate unstructured clouds of measurement points in \mathbb{R}^3 . With no surprise, measurement noises embedded in these points cannot be avoided, which make the downstream mesh reconstruction very troublesome.

The existing work in literature can be classified into two major groups: 1) computational geometry approaches and 2) volumetric reconstruction techniques. The computational geometry approaches are usually based on the Voronoi diagram of a given point cloud and reconstruct a mesh surface directly linking the input samples. Normal information is not required. However, it is generally difficult to avoid including noises in the final reconstructed surface. Moreover, as both the memory and the time cost to compute Voronoi diagram are expensive, these approaches are always applied to small or medium size of cloudy points (e.g., input with less than 30K points).

The volumetric reconstruction techniques attempt to build a signed implicit function that interpolates or approximates the point cloud samples, and then reconstruct its isosurface using, e.g., the Marching Cubes algorithm [28]. Nevertheless, the computation of such a signed implicit function requires the point cloud samples to be equipped with normal vectors, which can hardly be obtained directly from scanning devices. The estimation of normals on given cloudy points is actually one of the most critical steps in the reconstruction pipeline – especially when the points are in the presence of noise, sharp features, or thin structure. Although the input samples can be denoised slightly by applying the approximation scheme of implicit function reconstruction (e.g., [34]), the sharp features are always blurred together with noise. Furthermore, the approximation techniques like least-square fitting in general cannot satisfactorily handle the outliers.

[†]Corresponding Author.

*Submitted to ACM Symposium on Solid and Physical Modeling, October 5-8, 2009, San Francisco, California.

We present a robust mesh reconstruction method from noisy point in this paper. In contrast to existing approaches, we evaluate normal vectors on noisy point cloud by a highly robust estimator which allows us to reconstruct normals that well preserve sharp features. The points are projected to the robustly fitted surface. The resultant clean massive points are further down-sampled into user specified number of points. The subsampling is based on an iterative clustering algorithm and with the shape-approximation-error minimized. Note that, as only the position and the plane at a sample are required in the subsampling, the direction of normals determined in the first step is unnecessary to be con-

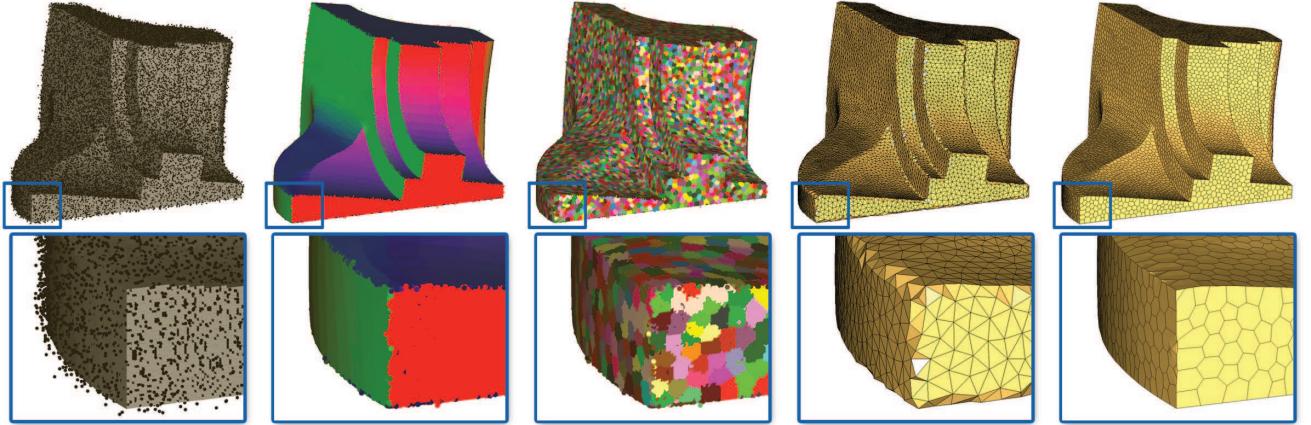


Figure 1: Overview of our robust mesh reconstruction pipeline: (leftmost) the given fandisk model with 18% Gaussian noises randomly distributed in the range of 0.5% of the bounding box’s diagonal length, (left) the fandisk model with normals estimated and outliers removed, (middle) the clustering result of subsampling and the points in different colors belong to different clusters, (right) the triangulation result on down-sampled points, and (rightmost) the final resultant mesh model with sharp features reserved.

sistently pointing outwards (or inwards). Lastly, a combinatorial approach is employed to connect the down-sampled points into a triangular mesh and its dual-graph, a polygonal mesh preserving sharp features, is computed. Figure 1 gives an overview of the steps in our approach. In order to borrow the advanced computational power available on consumer PCs, the time-consuming steps of robust normal estimation and subsampling have been parallelized using the architecture of GPU.

Main results The main results of this paper fall in the following aspects:

- A highly robust and parallel normal estimation and point projection method for noisy point cloud, which preserves the shape of sharp features;
- A parallel subsampling method of points with shape-approximation-error minimized;
- A new efficient mesh reconstruction pipeline based on the above two techniques.

Thanks to the most advanced parallel computation power which is available on consumer PCs, the efficiency of our approach becomes much outstanding.

2. RELATED WORK

The related works in the aspects of volumetric reconstruction, combinatorial approaches, robust statistics in surface reconstruction, down-sampling of massive points, and streaming and parallel computing are reviewed.

Volumetric reconstruction: Pioneered by the work of Hoppe et al. [17], the approaches in this category always start from estimating normals by a local principal component analysis (PCA), followed by a graph search to unify their inside/outside direction. Then, the samples equipped with normals are used to construct an implicit function in

the forms of signed distance field [17, 16], piecewise algebraic surfaces combined with α -shape [6], globally supported radial basis functions (RBF) [44, 9], compactly supported RBF [36], blended quadratic functions [33, 34, 47], or 3D indicator functions [22, 23, 7, 2]. Afterwards, a marching cubes algorithm [28] is employed to reconstruct the surface at the zero level-set of the implicit function. All these approaches rely on the input of oriented points, which is however difficult to obtain from scanning devices. Recently, Hornung and Kobbelt developed a method in [18] to reconstruct watertight 3D models from point clouds without normal information. They converted the surface reconstruction into a minimum cut problem of a weighted spatial graph structure. A mesh template fitting based method was proposed in [38] to realize a similar function. However, there is no direct extension of these methods to generate mesh surface preserving sharp features.

combinatorial approaches: The problem of mesh reconstruction was also approached from the computational geometric point of view. Amenta et al. in [3, 5] gave a provable guarantee of reconstructing a correct model given a minimum sampling density dependent on the local feature size. Recently, the approach was extended to be able to handle noisy input in [30]. However, as they did not remove outliers, the quality of resultant meshes was not good. Several variations of [3] are available in [4, 12, 13]. Recently, Kuo and Yau in [24] proposed a combinatorial algorithm to triangulate a given point cloud with sharp features. When applying their algorithm to practical data sets, there are two difficulties: 1) similar to other combinatorial approaches, it is sensitive to noise and 2) the measured points rarely locate along the sharp features. Furthermore, it is uneasy to apply this sort of method to massive points.

Robust statistics in surface reconstruction: The computer graphics community pays more attention to the robust statistics based methods recently. Ivrissimtzis et al. [19] employed neural network as a triangular mesh to con-

nect sample points and updated the mesh respectively. A method to quantify uncertainty in point cloud data by analyzing how far a point agrees with locally weighted planes has been proposed in [29]. The authors in [42] used support vector machine for reconstruction, hole filling and morphing between data sets. Schall et al. in [37] employed locally defined kernels to analyze the point neighborhood, and then computed a global surface probability distribution. However, all these approaches did not solve reconstruction problems of a surface fitting corresponding to less than 50% of the data points or a surface fitting to multi-structure. Such problem was first addressed in [15] by a forward search approach, but they projected points onto moving least squares (MLS) surfaces instead of reconstructing explicit meshes. Another projection operator, Locally Optimal Projection operator (LOP), is introduced by Lipman et al. in [26]. The LOP shows high robustness to noise and outliers, but the problem of preserving sharp features is not considered. The techniques employed in [14] and [20] are quite similar, where both used a Gaussian error model in conjunction with surface priors and performed numerical optimization to maximize the posterior probability of the model. The work in [14] focuses on a given triangular mesh, and Jenke et al. in [20] processed point clouds into well-sampled ones with noise removed. Nevertheless, since they are based on region growing, their computations are very time-consuming and can hardly be parallelized. Unlike [15] and ours, the method in [14] relies on the sharp edge identification and it may fail if the noise is less than the selected curvature threshold. Moreover, as shown in Fig.3, ours outperforms the method of [15] on the point set with a very low signal-to-noise ratio.

Down-sampling of massive points: Given a point set, the decimation process in [1] repeatedly removes the point that contributes the smallest amount of information to the shape. Kalaiah and Varshney [21] represented surfaces by a sampled collection of differential points and offered a novel point-based simplification technique that factored in the complexity of local geometry. Song and Feng [41] studied the problem of point cloud simplification by searching for a subset of the original input data set according to a user-specified number of points. The time cost of computing in these approaches however is very expensive, and they are difficult to be parallelized. A technique very similar to ours was [45] where Valette et al. proposed a local update scheme, but not K-means or Lloyd relaxation [27, 11], to compute Constrained Voronoi Diagram (CVD) on a given mesh surface, and then remeshed the given surface according to CVD. The most important difference between theirs and our work is that they tried to locate the seeds of Voronoi diagram along sharp features, which however in general cannot be guaranteed (e.g., the result in the last figure of [45]). Here, we approximate the given cloud points with a set of proxies (i.e., Voronoi diagram). No matter the seeds of Voronoi diagram are along the sharp features or not, we can still reconstruct sharp features by using the points coupled with the normals.

Streaming and parallel computing: The Streaming technique has been employed in surface reconstruction from massive points for a long time, where the most recent work was [7]. Recently, Buchart et al. [8] proposed a GPU implementation of reconstruction by local Delaunay triangulation.

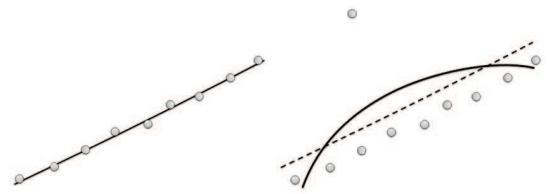


Figure 2: A single outlier can greatly distort a least squares fit: (left) no outlier and (right) one outlier only. Increasing the order of fitted model does not help either.

However, the noisy input has not been considered. In [48], a GPU-based implementation of [23] has been developed using NVIDIA’s CUDA. Nevertheless, the reconstruction with sharp features and the reconstruction on noisy input have not been addressed.

3. ROBUST NORMAL ESTIMATION AND PROJECTION

In our work we deal with fitting a surface to the local shape at a sample point in \mathbb{R}^3 . Then the point is projected onto the fitted surface and the normal vector of the projected point is estimated. The most classical method for fitting a model to data is linear regression using least-squares. However, as carefully discussed in [15], a single sample with a large error, an *outlier*, can change the fitted model arbitrarily. More specifically, as shown in Fig.2, a single outlier can fail a least-squares fit. Robust estimation techniques try to fit a model to data that contain outliers. Here, we choose a very robust one – Maximum Density Power Estimator (MDPE) and adopt it in a highly parallel algorithm of normal estimation and projection.

3.1 Robust Estimator

A robust estimator of local shape is very useful and important when the given point sets are in the presence of noises. Generally speaking, when a model is correctly fitted, the following two criteria should be satisfied:

- There are as many as possible data points on or near the model;
- The residuals of inliers should be as small as possible.

The least squares method uses the second criterion as its objective function to minimize the residuals without distinguishing the inliers from outliers. MUSE [31], the technique employed in [15] tries to minimize the scale estimate provided by the k th ordered absolute residual instead of minimizing the residual of inliers. Wang and Suter presented an estimator, MDPE, in [46] which considers both of these two criteria in its objective function. In comparison, it outperforms other estimators (RESC, ALKS, LMedS, RANSAC and Hough Transform) by tolerating more than 85% of outliers. MDPE is based on the strategy of random sampling to choose p points (called a p -subset) and then determine the parameters of a model for this p -subset, where for example $p = 2$ for a line, $p = 3$ for a circle or plane, and $p = 6$ for

a quadratic curve. It finally outputs the parameters determined by a p -subset with the minimum or maximum of the respective objective function.

Briefly, if the model to fit has been correctly estimated in MDPE, the data points on or near the fitted structure should have high score in the following probability density power function

$$DP = \frac{\sum_{X_i \in W_c} \hat{f}(X_i)}{\exp(|X_c|)} \quad (1)$$

where X_c is the center of the converged window W_c obtained by applying the mean-shift procedure, and $\hat{f}(X_i)$ is the multivariate kernel density estimator defined on a set of points $\{X_i\}_{i=1,\dots,n}$ in a d -dimensional Euclidean space \Re^d as

$$\hat{f}(x) = \frac{1}{nh^d} \sum_{i=1}^n K\left(\frac{x - X_i}{h}\right) \quad (2)$$

with the window band-width h and the Epanechnikov kernel K yielding minimum-mean integrated square-error. The kernel is defined as

$$K(X) = \begin{cases} \frac{1}{2}c_d^{-1}(d+2)(1-X^T X) & X^T X < 1 \\ 0 & \text{else} \end{cases} \quad (3)$$

where c_d is the volume of a unit d -dimensional sphere, e.g. $c_1 = 2$, $c_2 = \pi$, and $c_3 = 4\pi/3$. $d = 1$ is employed in our following normal estimation and outlier removal. The method used to compute the center X_c of a converged window W_c via mean-shift has been listed in Appendix A.

3.2 Normal Estimation and Projection

The above estimator is conducted to find a quadratic surface best fitting the local shape around a sample x . The basic idea is that, p points are randomly selected from the neighbors $N(x)$ of the given sample x to fit a quadratic surface S , and then the probability density power DP according to this fit S is evaluated by the residuals of points in $N(x)$ to S . The estimation will be repeated for m times, and among the m fits, the surface with the maximal score in DP is utilized as the robust fitting result.

More specifically, the robust estimation starts from choosing a search window radius h for MDPE and a repetition count m . The value of h greatly affects the robustness, the smaller h is used, the more sensitive to noises the estimator is. However, some inliers may be ignored if h is too small. By experiences, we choose $h = 2\bar{L}$ in all our examples (except the model with nonuniform point density), where \bar{L} is the average of point distances on the given model. Theoretically, the value of m relates to the probability P that at least one clean p -subset is chosen from m p -subsets as

$$m = \frac{\log(1-P)}{\log[1 - (1-\varepsilon)^p]}, \quad (4)$$

where ε is the fraction of outliers.

After randomly selecting p -subset, the points are used to form a quadratic surface S . Here, we first compute the centroid of the p points, and employ the principal component analysis (PCA) to form a local coordinate-frame at the average position (ref. [35]). Then, the surface S

$$S(s, t) = as^2 + bt^2 + cst + ds + et \quad (5)$$

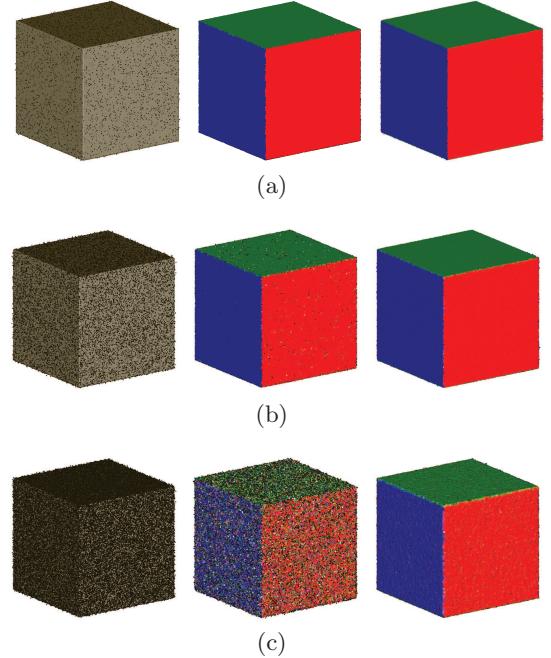


Figure 3: Robust estimator. (left column) The point set with 240K samples of a cube model embedding (a) 6% of noises, (b) 25% of noises and (c) 70% of noises. (middle column) RMLS starts to fail at 25% of noises. (right column) Our approach with the points successfully projected even at 70% noises. Points are displayed with color-coded normals.

is fitted by the mapped coordinates of these p points at this local coordinate-frame. Fleishman et al. in [15] suggested to let p equal the number of parameters in a quadratic surface to fit. Here we do not let $p = 5$ although there are only five parameters (a, b, c, d, e) to be determined in Eq.(5). Instead, we use $p = 6$ and then determine $S(s, t)$ by computing the least-square solution with singular value decomposition (SVD), which makes the model fitting numerically more stable.

By a fit, the residuals of all points in $N(x)$ to the determined surface $S(s, t)$ are used as input to the mean-shift procedure to compute a converged window. Note that, the mean shift is conducted in one-dimensional space – signed residual space. Lastly, the value of the probability density power function, DP in Eq.(1), is scored for this fit, S . The surface fitting will be applied for m times. Among all m fits, the fitted surface with maximum DP is regarded as the best surface S^* . Then the projected position x' of x is the closest point $x_c \in S^*$ to x which is searched by Newton's method. The normal of surface S^* at x_c is employed as the normal vector to equip x' .

Using the value of m defined in Eq.(4) as the number of repetition is impractical. There are two reasons for this. Firstly, we do not know the value of ε , the fraction of outliers. Secondly, using a value of m computed by Eq.(4) cannot guarantee to find a good fit among random selections, and it can be much higher as discussed in [43]. Therefore,

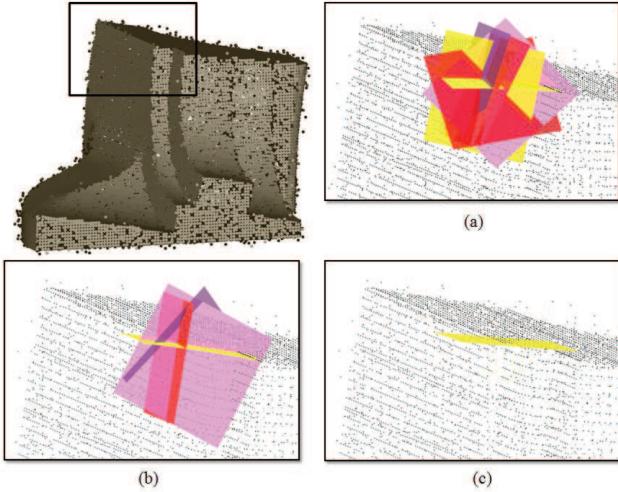


Figure 4: Performance comparison of different estimators at a highly noisy corner region: (a) the forward search misclassifies the regions as LMS fails to obtain a good initial fit, (b) the k th order estimator over classifies the regions, thus a surface region is mistakenly recognized on outliers, and (c) our MDPE based estimator does not have these problems – only the best fitted surface is estimated.

we use a more practical solution in our algorithm. After assigning a fixed number for m (e.g., $m = 300$), we can obtain a relatively clean point cloud with singular normal on very few isolated samples. Taking the cloudy points of the cubes in Fig.3 with different percentages of noises as an example, we can successfully project the points onto the cubes while [15]’s RMLS starts to fail at 25% of noise. The ability to fit surface corresponding to less than 50% of the data points is very important to the correct normal estimation on samples near sharp features where there are multi-structures.

3.3 Comparison with RMLS

The approach in [15] requires a robust initial estimator to start the forward search algorithm. It is essential for the initial estimator to fit an outlier-free surface as the forward search is carried out based on this initial guess. They adopted k th ordered-statistics [31] to grade the fitted surfaces instead of using least median of squares (LMS). In Fig.4, we compare the influence of LMS and k th order to the forward search with the MDPE at a corner of noisy region. We can clearly notice from Fig.4(a) that the forward search misclassifies the region as LMS fails to obtain a good initial fit. With k th order statistics, a good initial fit can be obtained but the region is over classified to four surfaces at a corner which actually contains three surfaces only as shown in Fig.4(b). This is because one forward search is conducted on outliers. It is however difficult to determine whether the regions classified by forward search belong to outliers or real surfaces. Hence, the point would be projected to a wrong position. In contrast, our approach only estimate the best fitted surface among the noisy region with MDPE as demonstrated in Fig.4(c). This ensures that the point is projected to the correct surface. Note that if the point is outside of the actual model, there is a chance for the point to be pro-

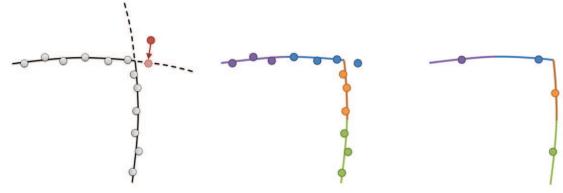


Figure 5: A miss-projected point outside the model (left) can be eliminated by the clustering (middle) and the subsequent subsampling (right).

jected on an invalid position. Nevertheless, such points are eliminated in the following subsampling step as illustrated in Fig.5.

3.4 Highly Parallel Implementation

Similar to all other random sampling techniques, the computation of robust estimators is very costly in time. Running the above algorithm on an advanced PC with 250K points takes about one and a half hours. Different from the techniques employed in [15, 14, 20], the proposed estimation method in this paper can be parallelized using the single-instruction-multiple-data (SIMD) parallelism and the architecture that is available on the consumer graphics hardware with the graphics processing unit (GPU). We first pass and store all samples in the given point cloud to the texture memory of graphics card. Then, the ANN KD-tree [32] is adopted to find k nearest neighborhoods of every sample. Following the suggestion of [15] using $k = 6p$ neighbors gives good results in our tests. Since the KD-tree does not allow multiple accesses at the same time, we store the query result in a neighborhood information table for later usage. Afterwards, records of the table are passed to the **Algorithm Normal Estimation and Point Projection** to evaluate the normal vectors and identify outliers in a streaming manner. Because of the texture memory limitation on a graphics card, we process 2048 samples in each pass. The parallelism is easy to implement using NVIDIA’s CUDA library.

Algorithm 1 Normal Outlier Estimation

```

1: Initialize  $DP_{\max}$  of a given sample  $x$  by zero;
2: for  $i = 1$  to  $m$  do
3:   Randomly choose  $p$  points to form a  $p$ -subset,  $P_i$ ;
4:   Fit a quadratic surface  $S$  to  $P_i$ ;
5:   Compute the signed residuals for all neighbors of  $x$ ;
6:   Use the mean shift procedure to determine the center
       $X_c$  of converged window on the residuals;
7:   Evaluate  $DP_i$  by Eq.(1);
8:   if  $DP_i > DP_{\max}$  then
9:      $DP_{\max} \leftarrow DP_i$  and  $P_{\max} \leftarrow P_i$ ;
10:  end if
11: end for
12: Fit a quadratic surface  $S^*$  to  $P_{\max}$ ;
13: Find the closest point  $x_c$  on  $S^*$  to  $x$ ;
14:  $x' \leftarrow x_c$ ;
15: Let the surface normal of  $S^*$  at  $x_c$  be the normal of  $x'$ ;

```

4. SUBSAMPLING

It is impractical to generate a mesh connecting the cleaned massive points as the number is huge. Therefore, we downsample the point set into user specified number of points

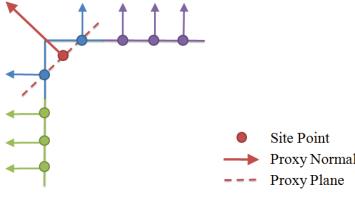


Figure 6: Using the shape error measurement obtained by the proxy plane passing site point cannot avoid moving site point away from the surface near sharp features; however, this can be avoided using our shape prior energy term $E_{shape}(x)$ which gives larger error. The samples in different colors belong to different clusters.

to be further triangulated. In order to control the quality of subsampling, we develop an energy minimization based method that groups the given massive points into clusters. The shape of points in a cluster is then approximated by a proxy represented by a site point, which is the average position of all points in this cluster.

4.1 Energy Function

The formulation of energy function in clustering is based on two criteria:

- The distribution of clusters should enable their proxy best approximate the shape of the given model.
- Clusters should maintain a disk-like shape.

To satisfy them, we define two energy terms to score clusters.

Distance Energy To control the disk-like shape of clusters, we introduce an energy term based on distance according to the site point p_i of a cluster C_i as

$$E_{dist}(x) = \|x - p_i\|^2. \quad (6)$$

Shape Prior Energy Our shape prior energy is

$$E_{shape}(x) = \|(x - p_i) \cdot n_x\|^2. \quad (7)$$

Note that, different from [11, 41], we employ the normal vector n_x at a sample x but not using the normal vector of a proxy as $E_S^P(x) = \|(x - p_i) \cdot n_{p_i}\|^2$ with n_{p_i} being the average normal of all samples in the proxy. This is because using a metric with the proxy normal does not sensitively reflect the shape-approximation-error when the cluster is crossing a sharp edge. For example, in Fig.6, the site point is far from the original surface but shows zero energy by E_S^P . Locating a site point far from the original surface will introduce large shape error on later generated mesh surfaces. Such error can hardly be recovered and should be prevented. Differently, using E_{shape} defined in Eq.(7) will draw the site points of optimization result near the given point cloud, which will generate mesh surfaces with small shape errors.

The energy on a cluster C_i is

$$E(C_i) = w_1 \sum_{x \in C_i} E_{dist}(x) + w_2 \sum_{x \in C_i} E_{shape}(x), \quad (8)$$

and the global energy of a clustering is defined by adding the energy terms of all clusters together as

$$E_{global} = \sum_i E(C_i). \quad (9)$$

We choose $w_1 = 0.1$ and $w_2 = 0.9\varsigma$ with

$$\varsigma = \frac{\text{avg}\{E_{dist}(x)\}}{\text{avg}\{E_{shape}(x)\}} \quad (10)$$

defined according to the initial clustering to balance the weight between E_{dist} and E_{shape} in Eq.(8).

4.2 Clustering Optimization and Subsampling

Firstly, an initial clustering is performed to partition the cleaned point cloud into n , a user specified number, clusters. n points are randomly selected from the point cloud as site points of clusters and are inserted into a KD-tree. Then, the cluster ID of every rest point is assigned by querying the closest site point through the KD-tree. This initialization step in general can be completed very fast.

Secondly, the clustering is optimized by minimizing the global energy in Eq.(9) with an iterative algorithm. A clustering can be optimized just according to the tests on the boundary points between different clusters. Given a boundary point $x_b \in C_i$ adjacent to another cluster C_j ($i \neq j$), shifting p from the cluster C_i to the cluster C_j will change the energy locally on $E(C_i)$ and $E(C_j)$. Therefore, the cluster ID of x_b is changed from i to j if such shifting reduces the global energy. Due to the local manner of clustering update, the optimization step can be parallelized to employ the computing power of GPU. The k -nearest neighbors of each sample are pre-computed by the spatial partition techniques (e.g., KD-tree) and stored in a neighboring information table. Choosing $k = 10$ balances both the quality and the speed. Through this table, we can detect every sample if it is on the boundary in parallel by comparing its cluster ID with the cluster IDs of its neighbors. The cluster shifting of a boundary point by comparing the local energy change can also be performed in parallel. We have implemented the **Algorithm Clustering Optimization** on GPU.

Algorithm 2 Clustering Optimization

```

1: repeat
2:   for each cluster  $C_i$  in parallel do
3:     Update the site point to the average position of all
   points in  $C_i$ ;
4:   Find the boundary points in  $C_i$ ;
5: end for
6: for each cluster  $C_i$  in parallel do
7:   for each boundary point  $x_b \in C_i$  do
8:     for neighbors  $x_j \in C_j$  of  $x_b$  do
9:       if moving  $x_b$  to  $C_j$  reduces the energy then
10:        Update the cluster ID of  $x_b$ ;
11:       end if
12:     end for
13:   end for
14: end for
15: until the change of  $E_{global}$  is less than 1%;
```

Lastly, we take the site points equipped with the normals at their closest points as the subsampling result.

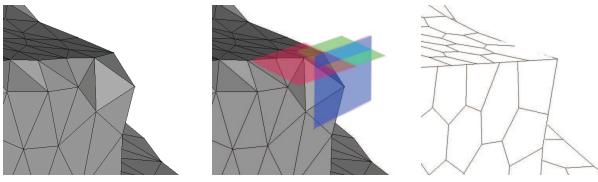


Figure 7: Reconstruction of sharp features: (left) sharp features are destroyed on the mesh surface generated by Tight-CoCone, (middle) the vertices (i.e., down-sampled points) are equipped with robustly estimated piecewise – planes are displayed for showing the normal directions at vertices, and (right) the sharp feature can be reconstructed by the vertices on the dual graph mesh whose positions are determined by minimizing QEF using SVD.

5. MESH GENERATION

We first triangulate the down-sampled points into a surface M by the Tight-CoCone algorithm [13]; however, the sharp features are chamfered (as seen in the left of Fig.7). To recover sharp features, we generate a dual-graph \tilde{M} of M by converting each vertex in M to a polygon and each triangle $T_i \in M$ into a vertex $v_i \in \tilde{M}$. As every vertex in M generated from the down-sampled point cloud is equipped with a piecewise normal vector, we locate vertex $v_i \in \tilde{M}$ by the position which minimizes the Quadratic Energy Function (QEF) defined by the three vertices in T_i and their normals. To be robust, the position is computed using the *Singular Value Decomposition* (SVD). The resultant polygonal mesh \tilde{M} thus preserves sharp features quite well.

5.1 Initial Surface Reconstruction

Our task here is to reconstruct an initial surface by connecting the down-sampled points. Based on the clustering result, we can form triangular faces using the connectivity of clusters which can be easily obtained by comparing the cluster ID of neighboring points. Although such triangulation can be performed locally resulting in very fast processing time, the reconstructed mesh cannot be avoided from containing holes and non-manifold parts. Hence, we adopt the Tight-CoCone algorithm which can guarantee the two-manifold topology on resultant triangular meshes.

5.2 Computing Dual-Graph

Locating vertex $v_i \in \tilde{M}$ actually is a task to find the optimal position in collapsing the triangles $T_i \in M$. The ideal vertex position should minimize the deviation from all the planes defined on $v_i \in T_i$ that are equipped with normals n_i . Here, the normal at each vertex is computed by the robust estimation method described in section 3. In other words, such position must have the minimum distance to the planes $(x - v_i) \cdot n_i = 0$.

More specifically, consider one vertex $v_i \in T_i$, the quadratic distance of an arbitrary point x from the supporting plane of that vertex is defined as

$$(n_i^T x - d_i)^2 \quad (11)$$

where n_i is the normal vector of v_i and

$$d_i = (v_i - \bar{v}) \cdot n_i \quad (12)$$

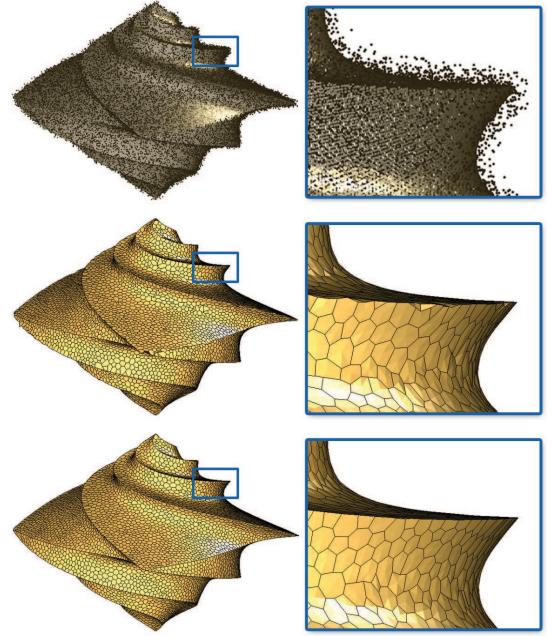


Figure 8: Reconstruction of the Octa-flower model: (top) input points with 18% of noises added, (middle) reconstruction using RMLS for normal estimation and point projection. and (bottom) our approach can successfully reconstruct the sharp features that are damaged when using RMLS to estimate normals and project points.

where \bar{v} is the average of $v_i \in T_i$. Thus, the objective function for minimization is given by

$$E(x) = \sum_i (n_i^T x - d_i)^2 \quad (13)$$

The iso-contours of this error function are ellipsoids and hence this function is called the *Quadratic Energy Function* (QEF). The minimized position is given by the solution of

$$\left(\sum_i n_i n_i^T \right) x = \left(\sum_i n_i d_i \right) \quad (14)$$

The above system of equations can be solved directly if the matrix has full rank, i.e. the normal vectors n_i are not parallel to each other. In order to make the solution robust and avoid handling of special cases, the SVD is used here. As illustrated in the middle of Fig.7, two normals are almost parallel; the sharp corner however can still be reconstructed on \tilde{M} .

6. RESULTS AND DISCUSSION

We have implemented the proposed algorithms into a prototype system by C++. NVIDIA’s CUDA library is utilized to implement the parallel algorithms on GPU. Our test platform is a PC with two Intel Xeon 2.8GHz quad-core CPU and 4GB main memory. The PC also has a graphics card with NVIDIA GeForce 9800 GTX GPU and 512MB memory, and runs Windows Vista operating system.

To test the robustness of our approach in handling the noises and sharp features, we added noises (see Appendix B for de-

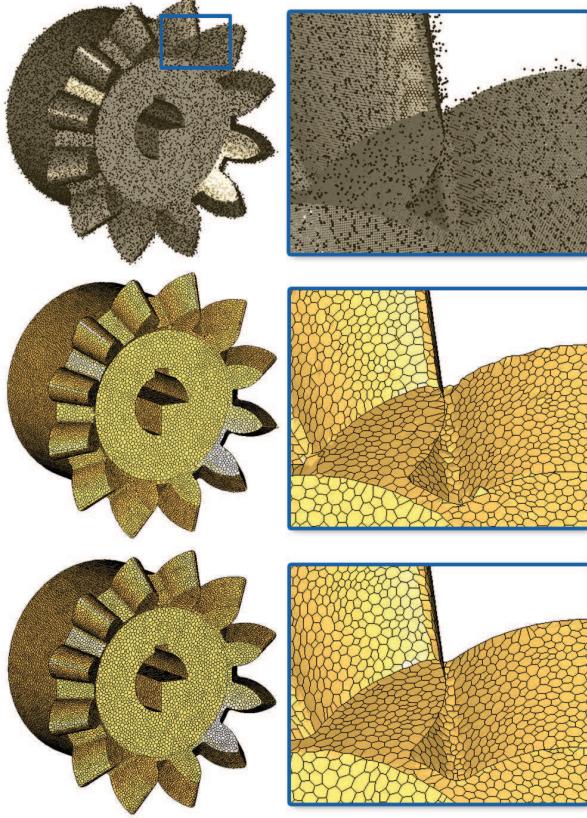


Figure 9: Reconstruction of the Gear model: (top) input points with 11% of noises added, (middle) reconstruction using RMLS for normal estimation and point projection, and (bottom) our approach can successfully reconstruct the surface with sharp features from large amount of points.

tails of noises generation) to the original point cloud with specified percentage of amount. In addition, we compare our results with RMLS [15] by embedding RMLS into our mesh reconstruction pipeline for the estimation of normals and projection of points. The first example we tested is the Octa-flower with 18% of noises as shown in Fig.8. The thin and sharp features have been successfully reconstructed by our approach but damaged by RMLS. A mechanical model Gear which contains huge number of points is distorted with 11% of noises and the reconstructed surfaces using RMLS and our approach are given in Fig.9. A much more complicated mechanical model Hub is distorted with large amount of noises and the results are demonstrated in Fig.10. In order to test the performance of our method on sculpture objects, we choose the Dragon model in Fig.11 with 18% of noises. The computational statistics of all these examples are listed in Table 1. Benefited from the parallelization of our algorithms, the speed of our surface reconstruction outperforms other methods using robust estimators. Model of half a million points only needs about five minutes to process. Another interesting study is to measure the shape error of reconstructed models to original models. We adopt the publicly available Metro tool [10] to measure the relative E_{mean} and E_{max} errors w.r.t. the diagonal lengths of bounding boxes. To visualize these errors, we use another

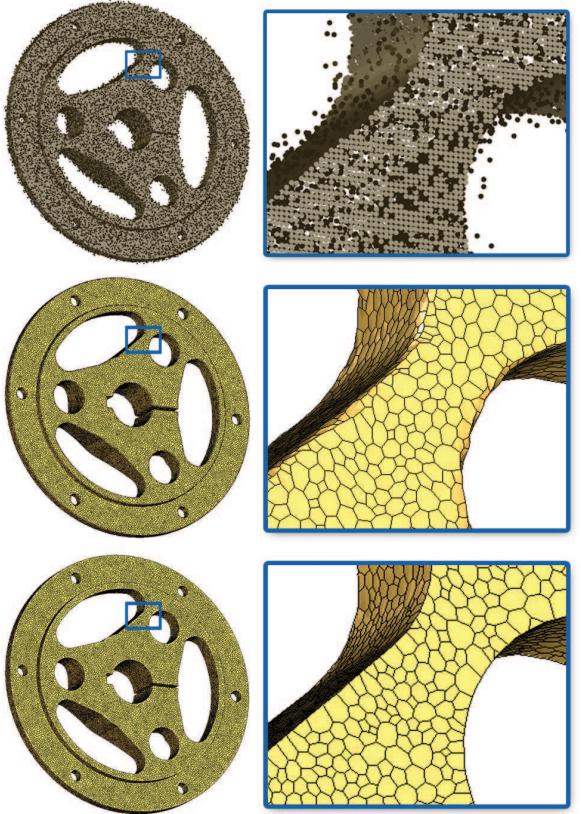


Figure 10: Reconstruction of the Hub model: (top) input points with 21% of noises added, (middle) reconstruction using RMLS for normal estimation and point projection, and (bottom) our approach can reconstruct the mesh surface from the input with high amount of noises.

mesh comparison tool, PolyMeCo [40], to illustrate the geometric differences between the reconstructed meshes and the original ones. Figure 12 shows the comparison results of the corresponding models using a common color scale. It can be concluded from the statistics in Table 1 and the visualizations that our method generates very small errors on reconstructed models.

In Fig.13, we further compare the performance of normal estimation and point projection between our approach and RMLS on an example of edge. When the noise level is increasing, our approach can still project the points onto the surface and estimate the normals correctly. Contrary, RMLS starts to produce errors at 10% noise and totally fails at 60% noise as demonstrated by the points that are color-coded with normals in Fig.13(b). In short, our approach is much more robust with the use of MDPE.

For real raw data, noises usually are not present everywhere. It is therefore impractical to perform the robust estimation for all points. It would waste a lot of time as the results of our projection and conventional Levin's projection [25] are nearly the same for inliers. Hence, we can first check the region around the point as described in [15]. If it is identified as smooth, we simply project the point by Levin's projec-

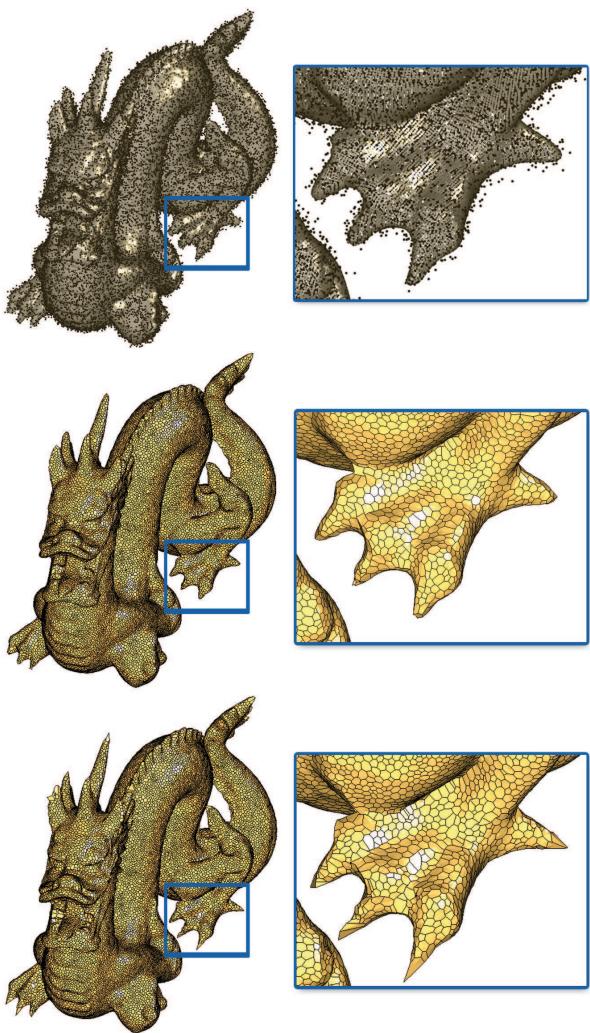


Figure 11: Reconstruction of the Dragon model: (top) input points with 18% of noises added, (middle) reconstruction using RMLS for normal estimation and point projection, and (bottom) our approach can successfully reconstruct the sculpture model.

tion. Otherwise, the point is projected using our proposed approach.

One of the difficulties in dealing with the real scanned data is the presence of structural noises which exist in a larger amount and distributes uniformly as shown in Fig.14(a). Surface will be generated from the structural noises since they behave the same as the real data point in local space. To deal with this, we increase the neighborhood size and process iteratively such that the signal-to-noise ratio is large enough for our robust estimation. Figure 14(c) shows the process dealing with structural noise in three iterations.

Lastly, to check whether the clustering based subsampling algorithm converges, we draw the bar chart of shape approximation error energy, E_{global} , and study its variation during the optimization. From Fig.15, we can easily observe

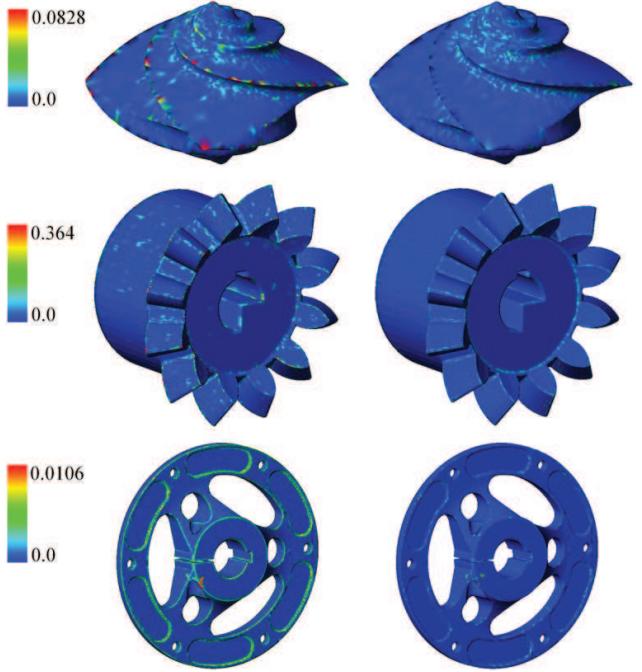


Figure 12: Illustration of the geometric differences between the reconstructed meshes and the original meshes: (left column) reconstructions using RMLS for normal estimation and point projection, and (right column) our approach shows much smaller errors at the sharp features.

that the energy, E_{global} , drops very fast using our local update based clustering algorithm. Usually, the iteration stops within ten steps.

The major limitation of our approach is to deal with the structural noises. As our approach is performed in a local manner, it is hard to distinguish between such kind of noise and the real surface. [39] introduced a new approach for removing non-isolated surface outlier clusters. In our future work, we will consider this as a preprocessing step followed by the method proposed in this paper to achieve the desired result in dealing with all the measurement errors generated from scanning devices.

7. CONCLUSION

In this paper, we have presented a robust and parallel technique to reconstruct mesh surfaces from unoriented noisy points. To solve the most difficult problem of surface reconstruction – preserving sharp features, we adopt a highly robust estimator which can fit surface corresponding to less than half of the data points. Therefore, the sharp features, which is in the form of data with multi-structures, can be reconstructed. After projecting the points and equipping the normals to them, an energy minimized subsampling is applied to result in a well-sampled point cloud. Lastly, a combinatorial approach is employed to reconstruct a triangular mesh connecting the down-sampled points and its dual-graph polygonal mesh is computed to recover sharp features. Parallelization method of our algorithms has also been given in the paper. The approach has been applied

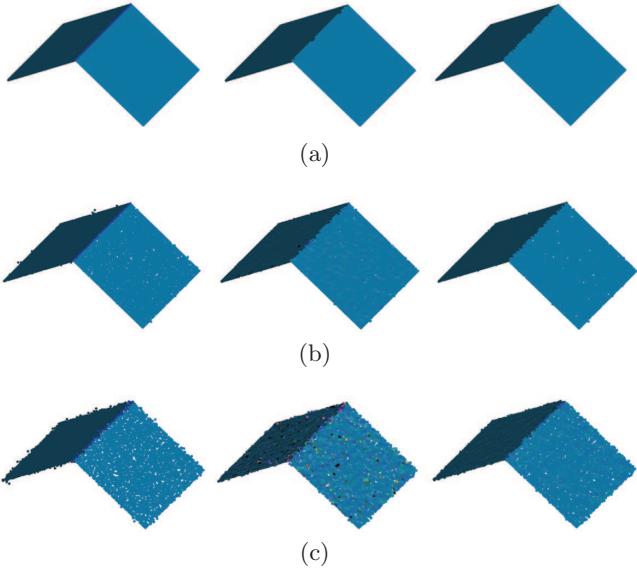


Figure 13: Projection test of RMLS versus our approach on an edge: (left column) original points, (middle column) RMLS result and (right column) our approach by MDPE. (a) Clean data. (b) Distorted with 10% Gaussian noises in range of 1% of diagonal. (c) Distorted with 60% Gaussian noises in range of 1% of diagonal.

to reconstruct several piecewise-smooth surfaces with sharp features from noisy cloud points in this paper, which demonstrates the functionality of our robust mesh reconstruction pipeline.

8. ACKNOWLEDGMENT

The research presented in this paper is partially supported by CUHK Direct Grant CUHK/2050400 and Hong Kong RGC CERG Grant CUHK/416307 and CUHK/417508.

9. REFERENCES

- [1] M. Alexa, J. Behr, D. Cohen-Or, S. Fleishman, D. Levin, and C. T. Silva. Computing and rendering point set surfaces. *IEEE Transactions on Visualization and Computer Graphics*, 9(1):3–15, January 2003.
- [2] P. Alliez, D. Cohen-Steiner, Y. Tong, and M. Desbrun. Voronoi-based variational reconstruction of unoriented point sets. In *SGP ’07*, pages 39–48, 2007.
- [3] N. Amenta, M. Bern, and M. Kamvysselis. A new voronoi-based surface reconstruction algorithm. In *SIGGRAPH ’98*, pages 415–421, 1998.
- [4] N. Amenta, S. Choi, T. K. Dey, and N. Leekha. A simple algorithm for homeomorphic surface reconstruction. In *SCG ’00: Proceedings of the sixteenth annual symposium on Computational geometry*, pages 213–222, 2000.
- [5] N. Amenta, S. Choi, and R. K. Kolluri. The power crust. In *SMA ’01: Proceedings of the sixth ACM symposium on Solid modeling and applications*, pages 249–266, 2001.
- [6] C. L. Bajaj, F. Bernardini, and G. Xu. Automatic reconstruction of surfaces and scalar fields from 3d scans. In *SIGGRAPH ’95*, pages 109–118, 1995.
- [7] M. Bolitho, M. Kazhdan, R. Burns, and H. Hoppe. Multilevel streaming for out-of-core surface reconstruction. In *SGP ’07*, pages 69–78, 2007.
- [8] C. Buchart, D. Borro, and A. Amundarain. A gpu interpolating reconstruction from unorganized points. In *SIGGRAPH ’07: ACM SIGGRAPH 2007 posters*, page 8, 2007.
- [9] J. C. Carr, R. K. Beatson, J. B. Cherrie, T. J. Mitchell, W. R. Fright, B. C. McCallum, and T. R. Evans. Reconstruction and representation of 3d objects with radial basis functions. In *SIGGRAPH ’01*, pages 67–76, 2001.
- [10] P. Cignoni, C. Rocchini, and R. Scopigno. Metro: measuring error on simplified surfaces. *Computer Graphics Forum*, 17(2):167–174, 1998.
- [11] D. Cohen-Steiner, P. Alliez, and M. Desbrun. Variational shape approximation. In *SIGGRAPH ’04*, pages 905–914, 2004.
- [12] T. K. Dey, J. Giesen, and J. Hudson. Delaunay based shape reconstruction from large data. In *PVG ’01: Proceedings of the IEEE 2001 symposium on parallel and large-data visualization and graphics*, pages 19–27, 2001.
- [13] T. K. Dey and S. Goswami. Tight cocone: a water-tight surface reconstructor. In *SM ’03: Proceedings of the eighth ACM symposium on Solid modeling and applications*, pages 127–134, 2003.

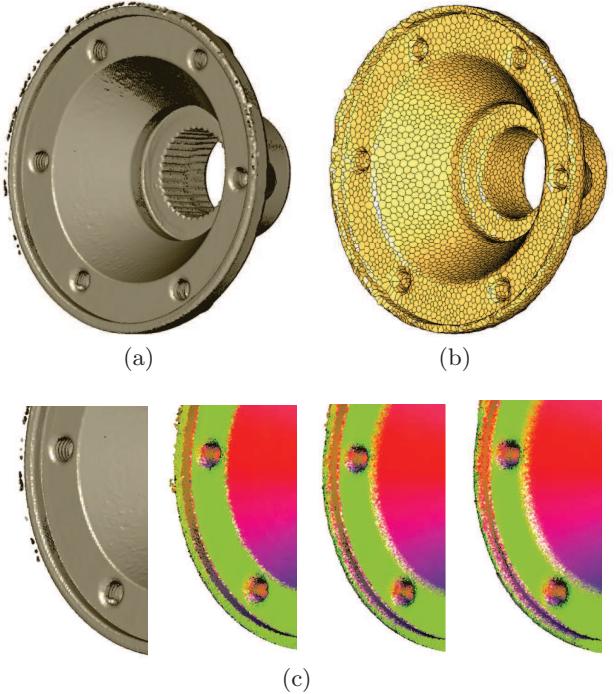


Figure 14: The mesh reconstruction of real scanned data *Carter*: (a) raw input with 546K points, (b) resultant mesh generated from 10K down-sampled points, and (c) the robust estimation step is performed three times in order to project the structural noise.

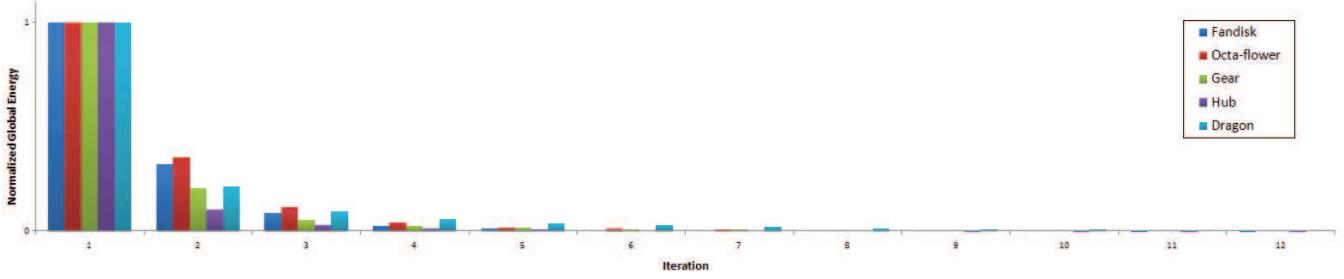


Figure 15: The change of energy E_{global} during clustering optimizations. The energy is normalized for display.

Table 1: Computational Statistics

Model	Noise Level	Given	Down-sampled	Robust Estimation	Subsampling	Mesh Generation	E_{mean}	E_{max}
Fandisk	18%	550K	10K	4.7 min	4.6 sec	5.4 sec	1.8×10^{-5}	2.3×10^{-3}
Octa-flower	18%	556K	10K	5.0 min	5.2 sec	5.5 sec	5.5×10^{-5}	3.8×10^{-3}
Gear	11%	884K	30K	8.2 min	7.0 sec	18.5 sec	5.2×10^{-5}	1.6×10^{-3}
Hub	21%	474K	50K	4.0 min	4.3 sec	35.6 sec	2.9×10^{-5}	2.5×10^{-2}
Dragon	18%	550K	50K	4.4 min	5.0 sec	34.0 sec	1.2×10^{-4}	2.6×10^{-2}

- [14] J. R. Diebel, S. Thrun, and M. Brünig. A bayesian method for probable surface reconstruction and decimation. *ACM Trans. Graph.*, 25(1):39–59, 2006.
- [15] S. Fleishman, D. Cohen-Or, and C. T. Silva. Robust moving least-squares fitting with sharp features. In *SIGGRAPH ’05*, pages 544–552, 2005.
- [16] H. Hoppe, T. DeRose, T. Duchamp, M. Halstead, H. Jin, J. McDonald, J. Schweitzer, and W. Stuetzle. Piecewise smooth surface reconstruction. In *SIGGRAPH ’94*, pages 295–302, 1994.
- [17] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle. Surface reconstruction from unorganized points. In *SIGGRAPH ’92*, pages 71–78, 1992.
- [18] A. Hornung and L. Kobbelt. Robust reconstruction of watertight 3d models from non-uniformly sampled point clouds without normal information. In *SGP ’06*, pages 41–50, 2006.
- [19] I. P. Ivrissimtzis, W.-K. Jeong, and H.-P. Seidel. Using growing cell structures for surface reconstruction. In *SMI ’03: Proceedings of the Shape Modeling International 2003*, page 78, Washington, DC, USA, 2003.
- [20] P. Jenke, M. Wand, M. Bokeloh, A. Schilling, and W. Straßer. Bayesian point cloud reconstruction. *Comput. Graph. Forum*, 25(3), 2006.
- [21] A. Kalaiah and A. Varshney. Modeling and rendering of points with local geometry. *IEEE Transactions on Visualization and Computer Graphics*, 9(1):30–42, 2003.
- [22] M. Kazhdan. Reconstruction of solid models from oriented point sets. In *SGP ’05*, page 73, 2005.
- [23] M. Kazhdan, M. Bolitho, and H. Hoppe. Poisson surface reconstruction. In *SGP ’06*, pages 61–70, 2006.
- [24] C.-C. Kuo and H.-T. Yau. A new combinatorial approach to surface reconstruction with sharp features. *IEEE Trans. Vis. Comput. Graph.*, 12(1):73–82, 2006.
- [25] D. Levin. Mesh-independent surface interpolation. In *Geometric Modeling for Scientific Visualization*, pages 37–49, 2003.
- [26] Y. Lipman, D. Cohen-Or, D. Levin, and H. Tal-Ezer. Parameterization-free projection for geometry reconstruction. In *SIGGRAPH ’07: ACM SIGGRAPH 2007 papers*, page 22, New York, NY, USA, 2007. ACM.
- [27] S. P. Lloyd. Least squares quantization in pcm. *IEEE Transactions on Information Theory*, 28:129–136, 1982.
- [28] W. E. Lorensen and H. E. Cline. Marching cubes: A high resolution 3d surface construction algorithm. *SIGGRAPH Comput. Graph.*, 21(4):163–169, 1987.
- [29] L. G. Mark Pauly, Niloy Mitra. Uncertainty and variability in point cloud surface data. In *Symposium on Point-Based Graphics 2004*, 2004.
- [30] B. Mederos, N. Amenta, L. Velho, and L. H. de Figueiredo. Surface reconstruction from noisy point clouds. In *SGP ’05: Proceedings of the third Eurographics symposium on Geometry processing*, page 53, 2005.
- [31] J. V. Miller and C. V. Stewart. Muse: Robust surface fitting using unbiased scale estimates. In *CVPR*, pages 300–, 1996.
- [32] D. M. Mount and S. Arya. Ann: A library for approximate nearest neighbor searching. Aug. 2006. <http://www.cs.umd.edu/~mount/ANN/>.
- [33] Y. Ohtake, A. Belyaev, M. Alexa, G. Turk, and H.-P. Seidel. Multi-level partition of unity implicits. *ACM Trans. Graph.*, 22(3):463–470, 2003.
- [34] Y. Ohtake, A. Belyaev, and H.-P. Seidel. 3d scattered data interpolation and approximation with multilevel compactly supported rbfs. *Graph. Models*, 67(3):150–165, 2005.
- [35] S. Petitjean. A survey of methods for recovering quadrics in triangle meshes. *ACM Comput. Surv.*, 34(2):211–262, 2002.

- [36] M. Samozino, M. Alexa, P. Alliez, and M. Yvinec. Reconstruction with voronoi centered radial basis functions. In *SGP '06*, pages 51–60, 2006.
- [37] O. Schall, A. Belyaev, and H.-P. Seidel. Robust filtering of noisy scattered point data. In M. Pauly and M. Zwicker, editors, *IEEE/Eurographics Symposium on Point-Based Graphics*, pages 71–77, Stony Brook, New York, USA, 2005.
- [38] A. Sharf, T. Lewiner, A. Shamir, L. Kobbelt, and D. Cohen-Or. Competing fronts for coarse-to-fine surface reconstruction. *Computer Graphics Forum*, 25(3):389–398, 2006.
- [39] J. Shen, D. Yoon, D. Shehu, and S.-Y. Chang. Spectral moving removal of non-isolated surface outlier clusters. *Computer-Aided Design*, In Press, Corrected Proof:-, 2008.
- [40] S. Silva, J. Madeira, and B. S. Santos. Polymeco ” a polygonal mesh comparison tool. In *IV '05: Proceedings of the Ninth International Conference on Information Visualisation*, pages 842–847, Washington, DC, USA, 2005. IEEE Computer Society.
- [41] H. Song and H.-Y. Feng. A global clustering approach to point cloud simplification with a specified data reduction ratio. *Computer Aided Design*, 40(3):281–292, 2008.
- [42] F. Steinke, B. Schölkopf, and V. Blanz. Support vector machines for 3d shape processing. *Comput. Graph. Forum*, 24(3):285–294, 2005.
- [43] B. Tordoff and D. W. Murray. Guided sampling and consensus for motion estimation. In *ECCV '02: Proceedings of the 7th European Conference on Computer Vision – Part I*, pages 82–98, London, UK, 2002. Springer-Verlag.
- [44] G. Turk and J. F. O'Brien. Shape transformation using variational implicit functions. In *SIGGRAPH '99*, pages 335–342, 1999.
- [45] S. Valette, J. M. Chassery, and R. Prost. Generic remeshing of 3d triangular meshes with metric-dependent discrete voronoi diagrams. *IEEE Transactions on Visualization and Computer Graphics*, 14(2):369–381, 2008.
- [46] H. Wang and D. Suter. MDPE: A very robust estimator for model fitting and range image segmentation. *Int. J. Comput. Vision*, 59(2):139–166, 2004.
- [47] H. Xie, K. T. McDonnell, and H. Qin. Surface reconstruction of noisy and defective data sets. In *VIS '04: Proceedings of the conference on Visualization '04*, pages 259–266, 2004.
- [48] K. Zhou, M. Gong, X. Huang, and B. Guo. Highly parallel surface reconstruction. 2008. Microsoft Technical Report, MSR-TR-2008-53 <http://www.kunzhou.net/2008/MSR-TR-2008-53.pdf>.

APPENDIX

A. MEAN-SHIFT METHOD

To compute the center X_c of the converged window W_c on a given set of n data points $\{X_i\}_{i=1,\dots,n}$, the mean-shift update vector can be derived from the gradient of the kernel density estimated in Eq.(2). In short, the mean-shift update

vector is defined as

$$M_h(x) = \frac{1}{n_x} \sum_{X_i \in S_h(x)} X_i - x \quad (15)$$

where the region $S_h(x)$ is a hypersphere with radius h and contains n_x data points. The mean-shift procedure is as follows.

Algorithm 3 Mesh Shift

```

1: Choose the radius  $h$  of the search window;
2: Initialize the location  $X_c$  of the window with zero;
3: repeat
4:   Compute the mean shift vector  $M_h(x)$  by Eq.(15);
5:    $X_c \leftarrow X_c + M_h(x)$ ;
6: until reach the terminal condition

```

We stop the iteration when either $|(\|M_h^{last}(x)\| - \|M_h(x)\|)| < 0.01\|M_h^{last}(x)\|$ or the loop has run more than 300 times.

B. NOISES GENERATION

The noises added to the models are generated by first computing distortion magnitudes of each data point. The distortion magnitudes are distributed with zero mean Gaussian in the range of 0.5% of the bounding box's diagonal. After that, we randomly select specified percentage of points and shift those points with corresponding distortion magnitudes in a random direction. This ensures our noises generated are totally in unpredicted positions.