Voxel Structure-Based Mesh Reconstruction From a 3D Point Cloud

Chenlei Lv [ID], Weisi Lin [ID], *Fellow, IEEE*, and Baoquan Zhao [ID]

Abstract—Mesh reconstruction from a 3D point cloud is an important topic in the fields of computer graphic, computer vision, and multimedia analysis. In this paper, we propose a voxel structure-based mesh reconstruction framework. It provides the intrinsic metric to improve the accuracy of local region detection. Based on the detected local regions, an initial reconstructed mesh can be obtained. With the mesh optimization in our framework, the initial reconstructed mesh is optimized into an isotropic one with the important geometric features such as external and internal edges. The experimental results indicate that our framework shows great advantages over peer ones in terms of mesh quality, geometric feature keeping, and processing speed. The source code of the proposed method is publicly available[1].

Index Terms—Mesh reconstruction, intrinsic metric, voxel structure, isotropic property.

I. INTRODUCTION

AS ONE kind of 3D object representation, polygonal mesh[1] has been widely used in different applications such as 3D object animation [1], simplification [2], compression [3], digital watermarking [4], [5], saliency detection [6], etc. Mathematically, it is a simplicial complex structure to fit a 3D object, and maintains more important geometric features such as topology, curvature, and edges to support accurate geometric analysis [7], [8]. Comparing to a point cloud, a mesh provides more comprehensive representation. Although it has many advantages, a mesh is difficult to be achieved. Most of the 3D scanning devices can only provide point clouds, not meshes. Few devices support direct mesh acquisition [9] from real world and such devices are bulky and expensive. As a consequence, point cloud based 3D mesh reconstruction has emerged as one of the most fundamental techniques of 3D modelling.

In general, there are three key requirements involved in mesh reconstruction from point clouds, including local region detection, geometric feature keeping, and resampling. For the first requirement, according to the MLS (moving least squares) surface

[1][Online]. Available: https://github.com/vvvwo/Parallel-Structure-for-Meshing

theory [10], the surface of a 3D object can be divided as local regions, which are represented by neighbor relationship of points in a point cloud. However, such relationship is usually missing in raw data captured by scanning devices and thus needs to be restored for further processing using local region detection. For the second requirement, geometric features of original 3D objects should be kept in the reconstructed mesh, which are useful for geometric analysis [7], [8]. For the third requirement, resampling refers to reconstruction toward an isotropic 3D mesh with a given point number. In an isotropic mesh, the distances between each point to its neighbor points are approximately equal in the mesh. With the same point number, isotropic meshes share a regular form which is useful for applications such as mesh-based shape registration [11], animation [1], and retrieval [12].

Most existing methods attempt to detect local regions in the local tangent space, such as Centroidal Voronoi Tessellation (CVT) [13] and Gaussian kernel [14]. In the local tangent space, the neighbor relationship of points can be conveniently constructed and optimized. However, some geometric features such as external and internal edges could be lost. External edges means the sharp edges outside the 3D object and internal edges are inside the 3D object like holes. Some existing methods focus on certain geometric feature rebuilding in mesh reconstruction [15], [16]. Focusing on certain geometric feature keeping in reconstruction, such methods have poor versatility and robustness generally. Most existing methods are incapable of reconstructing an isotropic mesh with the number of points specified by a user [17]–[19]. It is a big challenge for a mesh reconstruction method to meet all the three aforementioned requirements. We propose a voxel structure-based mesh reconstruction to meet the three requirements.

The proposed two-step voxel structure based framework includes initial mesh reconstruction and mesh optimization. Firstly, the framework harvests an initial reconstructed mesh by the intrinsic metric with which means the distances between different points are defined by the geodesic distance instead of Euclidean distance [20]. This improves the accuracy of local region detection and avoids the wrong neighbor relationship of points. Based on local regions, an initial mesh can be reconstructed. Secondly, the framework optimizes the initial reconstructed mesh into an isotropic one with geometric features. The point number does not change in the final reconstructed mesh. The key contributions of our work are summarized as follows.

- We propose a voxel structure to reconstruct an initial mesh from a point cloud. It provides the intrinsic metric that
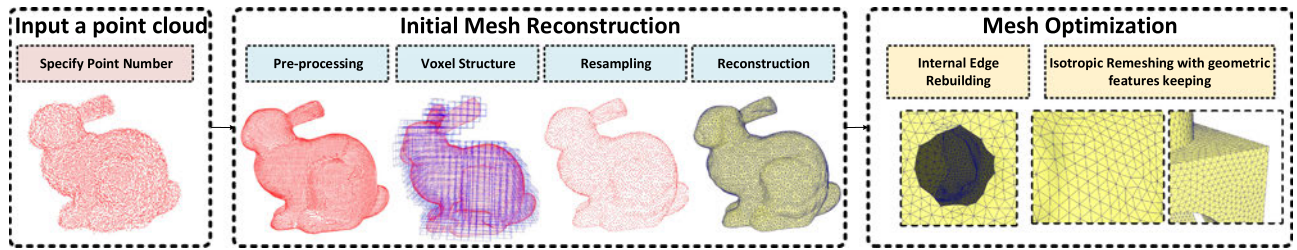
Fig. 1. The pipeline of the proposed mesh reconstruction method.

improves the accuracy of the initial reconstructed mesh with any given point number.

- Based on the voxel structure, we design a mesh optimization method for mesh quality improvement. With the proposed method, the initial reconstructed mesh can be optimized into an isotropic one while the important geometric features are kept.

The rest of the paper is organized as follows. In Sec. 2, we review existing classical methods for mesh reconstruction. We introduce the proposed initial mesh reconstruction method in Sec. 3, followed by the mesh optimization in Sec. 4. We demonstrate the effectiveness and efficiency of our method with extensive experimental evidence in Sec. 5, and Sec. 6 concludes the paper.

II. RELATED WORKS

There are different kinds of meshes such as triangular meshes [21], quadrilateral meshes [22], non-manifold meshes [23], and large-scale meshes [24]. In this paper, we focus on single 3D object triangular mesh reconstruction with the 2-manifold property. The related work can be broadly classified into three categories: Approximation-based, Delaunay-based, and Point resampling-based, as well as pre-processing and post-processing.

A. Approximation-Based Approaches

Approximation-based methods attempt to rebuild the 2-manifold mesh to fit a point cloud directly. The methods achieve the reconstruction by establishing an objective function, such as Poisson Function [19], [25], Scale Space [18], MLS optimization [26], [27], Implicit Function [28], Voxel-based reconstruction [29], [30], and Kernel Density Estimation [16]. The Poisson function was a classical method for mesh reconstruction [25]. The core idea was to obtain an indicator function by solving a Poisson formulation, which is a piece-wise constant function and signs the different sides of the surface. Scale space provided a smoothing operator for raw point clouds. By estimating the mean curvature and solving a mean curvature motion in point clouds, a smooth surface was constructed [18]. The MLS-based methods [26], [27] reconstructed a 2-manifold mesh based on local surface fitting. In summary, such methods reconstruct a smooth surface from a point cloud and are robust to noise. However, the local shape features are broken to a certain degree and the stability is not satisfied caused by the wrong estimation for face normal and incorrect approximate region.

B. Delaunay-Based Approaches

The Delaunay-based framework is regarded as the mainstream technology in 3D triangular meshing. It provides a simple and efficient point connect scheme for a point cloud without local surface approximation. A lot of mesh reconstruction methods based on Delaunay triangulation and their variations have been extensively investigated in the literature. Ameta [31] utilized the dual characteristics between Voronoi Diagram and Delaunay triangulation to rebuild the surface. He also provided some improvements, which are based on the crust algorithm to cover more geometric features in meshing [32], [33]. Cohen [17] proposed a greedy Delaunay triangulation to fix the local errors in the reconstructed mesh. Kuo [34] combined the advantages of Delaunay-based and region-growing methods to reconstruct the mesh. Peethambaran [35] built a graph structure to reconstruct a water-tight surface. Wang [36] proposed a new triangle selection strategy to reduce the topology errors and holes in Delaunay triangulation. Liu [37] extended the Delaunay-based strategy to image stitching. Such methods are restricted by the points' positions in general. The quality of the reconstructed mesh is poor when the points' distribution is nonuniform in a point cloud.

C. Point Resampling-Based Approaches

Considering the drawback of the Delaunay-based methods, some studies attempt to resample a point cloud into an isotropic one. As discussed in being isotropic means that the distances between each point to its neighbor points are approximately equal. For a point cloud, such distances from all points form a distance field. The resampling process is used to optimize the distance field to achieve an isotropic mesh. The representative method was Centroidal Voronoi Tessellation (CVT) [13], [38] based resampling. Based on the Lloyd's relaxation, the Voronoi Diagram was optimized in local tangent space [39]. The iteration of Lloyd relaxation required heavy calculation to update the Voronoi Diagram and new centers. Some researchers [40]–[42] attempted to improve the effectiveness based on parallel computation. Some other methods used different kinds of point sampling schemes to achieve a similar result. Luo [43] proposed a point cloud resampling method based on the Gaussian-weight Laplacian graph. Zhong [14] used a Gaussian kernel function to achieve the isotropic point cloud for mesh reconstruction. Such methods optimize the distance field of the point cloud before meshing. The advantages of the methods include the high quality of the triangulation results and the robustness to different local points' density. However, some local geometric features are
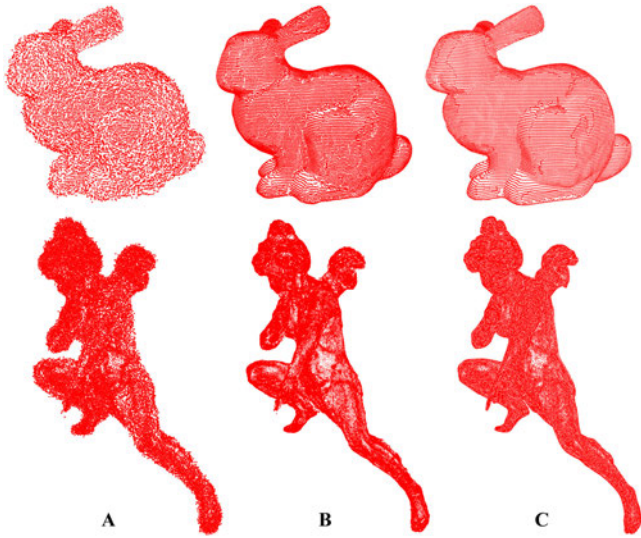
Fig. 2. Instances of denoising results. A: Point cloud models with noise; B: Denoising results; and C: Ground truth.
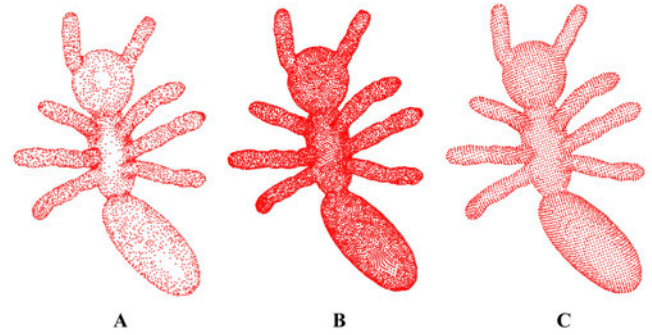

Fig. 3. An instance of octree-based density adjusting. A: A sparse point cloud with non-uniform density; B: Delaunay-based interpolation for point cloud up-sampling; C: octree-based density adjusting result.

lost during optimization. In our framework, the same scheme is adopted to achieve an initial reconstructed mesh from a point cloud resampling result. The distance field of the resampling result is optimized in the voxel structure. Our reconstruction method is classified into this part and attempt to keep geometric features in the reconstructed mesh.

D. Pre- and Post-Processing

The approaches for pre-processing and post-processing included mesh denoising [15], [44]–[46], isotropic remesh [47]–[51] and mesh repair [52]–[55]. Such methods are used to improve the mesh quality based on geometric evaluations [56]. In our framework, we provide similar functions in mesh optimization. Using mesh optimization with the voxel structure, the quality of a reconstructed mesh can be improved.

III. INITIAL MESH RECONSTRUCTION

As mentioned in Sec. 2, the initial mesh reconstruction task in our framework can be formulated as a point-based distance field optimization problem. Once the distance field is optimized, the local regions are convenient to be detected and the initial reconstructed mesh can be obtained. We propose a voxel structure-based framework to optimize the distance field. The framework mainly includes three main components: point cloud pre-processing, construction of voxel structure, and initial mesh reconstruction. The technical details of each component will be elaborated in the following subsections.

A. Point Cloud Pre-Processing

Limited by the accuracy of 3D scanning devices, raw point clouds generally carry noise and are with uncertain point density distribution, which could affect the points-based distance field optimization. Therefore, a point cloud pre-processing is

needed for input point cloud denoising and uniform density adjusting. For point cloud denoising, we use an MLS surface-based smoothing method [10]. It is formulated in (1). The parameters p_i and p_j represent the points from a point cloud P, p_j is the neighbor point of p_i and p_i' is the new position of point p_i after least squares fitting. It is achieved by weighted fitting from its neighbors. The weight function θ in (2) is suggested to fit a Gaussian function [57]. The h is a fixed parameter reflecting the anticipated spacing between neighboring points. In our paper, we compute distances $\{d_k(p_i)\}$ between all points and their k-th neighbors, $h = max\{d_k(p_i)\}$. Such processing can be regarded as a "pulling back" mapping from points to the MLS surface. In Figure 2, we show two denoising instances.

$$p_i' = \sum_{j=0}^{k-1} \frac{\theta(||p_i - p_j||)p_j}{\left|\left|\sum_{j=0}^{k-1} \theta(||p_i - p_j||)\right|\right|} \tag{1}$$

$$\theta(d) = e^{-\frac{d^2}{h^2}} \tag{2}$$

For uniform density adjusting, we utilize an octree-based method [58]. It is used to adjust the density of a point cloud based on the voxel boxes. The scale of the voxel box controls the distances between different points, which uniforms the points' density in a point cloud. In Figure 3, we show an instance of octree-based density adjusting. We compute the distance set $\{d_n(p_i)\}$ between each point to its nearest neighbor point. The scale is selected by the mean value from $\{d_n(p_i)\}$. In Figure 3 C, an instance of uniform density adjusting is shown. The local region can be detected from the uniform density point cloud conveniently in following step.

Using the octree-based density adjusting, the point number of a point cloud will be reduced. In some cases (sparse point clouds or user-specified point number is similar to a given point cloud), we would not like to reduce the point number while adjusting the point cloud density. Then an up-sampling is needed before the density adjusting. We propose an up-sampling processing based on Delaunay-based interpolation. The Delaunay-based interpolation builds the Delaunay triangulation region for each point and inserts the new points into the region. The up-sampling points are inserted into the triangle's edges at first. The insert point number in single edge s_l as defined in (3) (s = 6 in default).
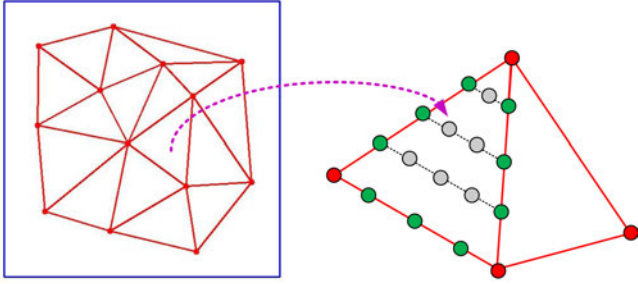
Fig. 4. An instance of the up-sampling. The gray points represent new points ($s = 6$) by the up-sampling. The points are proportional to the triangle's area. The green points in single edge ($s_l = 3$) are computed by s.

Fig. 6. Comparison of Euclidean distance and geodesic distance. The geodesic distance reflects the real distance between p_0 and p_1 on the curve. Based on the voxel structure, the global geodesic distance is fitted by the sum of local Euclidean distance in each voxel box.

cloud P with $|P|$ points.

$$v_{scale} = \left\lceil \frac{2 \times l}{\sqrt[3]{|P|}} \right\rceil \qquad (4)$$

C. Initial Mesh Reconstruction

Based on the voxel structure, we provide a point cloud resampling scheme for initial mesh reconstruction. The resampling optimizes the distance field. The point number after resampling is equal to the user-specified number. Based on the resampling result, the initial reconstructed mesh can be obtained.

In our scheme, the resampling is processed in different voxel boxes independently. The geodesic computation can be approximated by Euclidean measure. It reduces the computation cost while keeping the intrinsic metric for resampling. Following the property, we design the resampling scheme to be a sub-division strategy. The resampling task for a point cloud is divided into local resampling tasks in different voxel boxes. According to the resampling rate R_s or user-specified number $|P_s|$, the local resampling point number $|P_{vs}|$ in the voxel box v is computed by proportional calculation in (5). The parameter $|P_v|$ represents the point number in v.

$$|P_{vs}| = |P_v| \times R_s, |P_{vs}| \in N^+ \qquad (5)$$
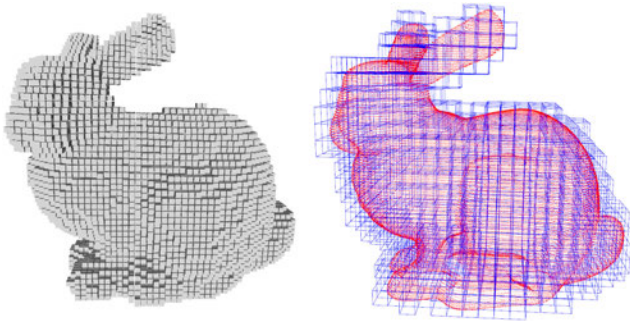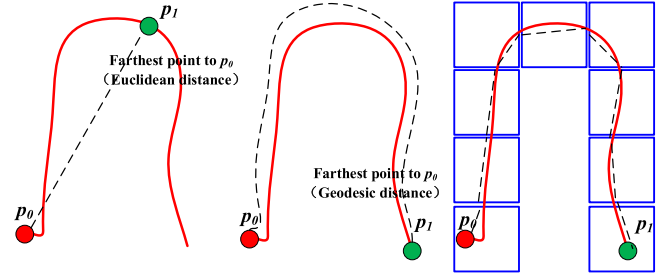$$R_s = |P| / |P_s|$$

Fig. 5. Comparison between voxelization and voxel structure.

Then we insert the points into the triangle region as shown in Figure 4. In Fig. 3(b), we show the up-sampling instance. After the pre-processing, a point cloud can be used for the next steps.

$$s_l = \left\lceil \sqrt{2s + 1/2} - 1/\sqrt{2} \right\rceil \qquad (3)$$

B. Construction of Voxel Structure

The voxel structure is proposed for the point-based distance field optimization. It is similar to the octree and voxelization. In Figure 5, we compare the voxelization and the voxel structure. The points are aggregated into voxel blocks by voxelization, but divided into different voxel boxes in the voxel structure. The voxel structure provides an intrinsic metric for a point cloud to avoid the incorrect local region detection by Euclidean distance. In Figure 6, we compare the two different distance definitions.

The intrinsic metric is based on the geodesic distance, which is similar to the shortest path search between different voxel boxes in the voxel structure [20]. Inside a voxel box with a small scale, curvature transfer of points is controlled in a small range. The geodesic distance is similar to the Euclidean distance, and therefore approximate errors are decided by the scale of the voxel box. Once the scale is determined, the voxel structure is established. The scale should balance the approximate error and point number in different boxes (if the scale is too small, there will be no enough points in a single voxel box for the further process). We provide a scale calculation formula as a default suggestion. In 4, we show the formula of the scale v_{scale}. The length l is the longest border of the bounding box from a point

Once the local resampling point numbers for different voxel boxes are determined, the resampling can be processed in parallel. We use the Farthest Point Sampling (FPS) [59] to resample the point cloud in different voxel boxes. Using FPS can achieve an approximate isotropic local resampling result. Even the local resampling results satisfy the requirements of resampling, the final resampling result can not be achieved by the combination of local resampling results. The reason is that the distances between points in adjacent voxel boxes are not approximately the same. To solve the problem, we divide the parallel computation for local resampling into different rounds. The adjacent voxel boxes are not resampled in the same round. Besides, resampling points of neighbor boxes should be included by FPS for the processing voxel box. In Figure 7, we show an instance to explain the resampling scheme. The parallel computation is divided into 8 rounds which guarantees the approximate isotropic property of resampling. Based on the resampling result, we use a Delaunay
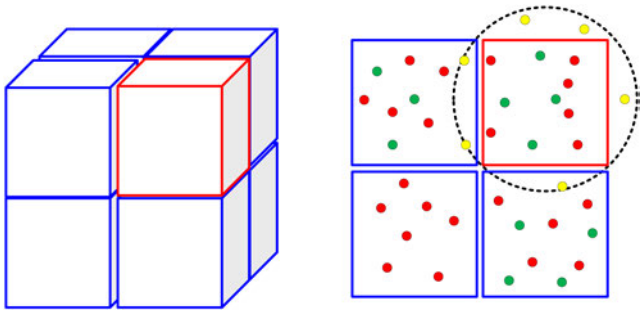
Fig. 7. An instance of the resampling scheme for a voxel box (red). The red points represent the original points in the point cloud, while the green ones are resampled points by FPS. The yellow points are resampled points in neighbor voxel boxes, which are considered by FPS for the red voxel box.
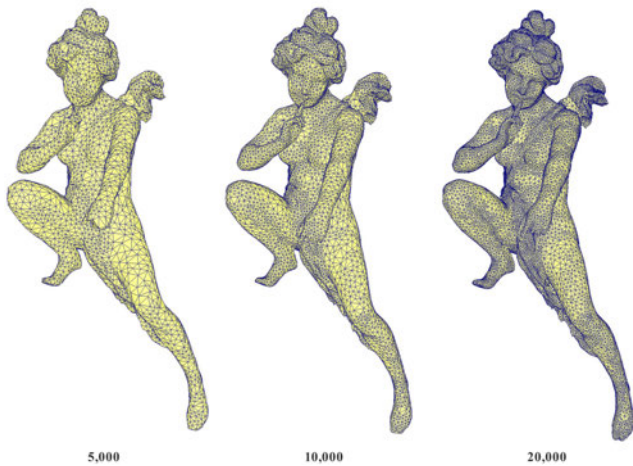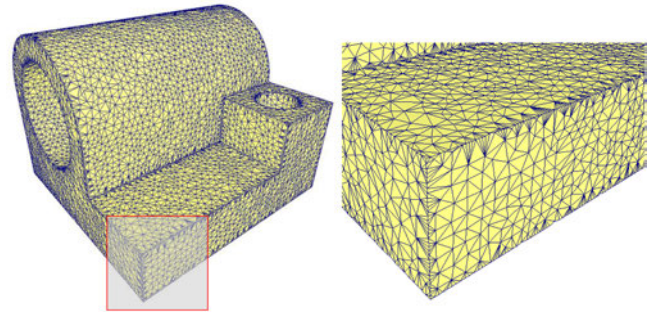

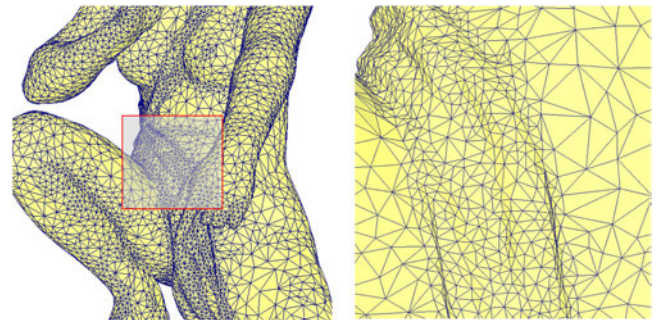Fig. 9. An instance of reconstructed mesh with external edges.

Fig. 10. An instance of reconstructed mesh with curvature sensitive keeping. The points are classified into five categories based on curvature values, the proportion of the rates is 2:3:4:5:6.

Fig. 8. An instance of the initial reconstructed mesh with different point numbers by user specification.

different resampling rates, the curvature sensitive initial reconstructed mesh can be achieved. In Figure 10, we show an instance of the mesh reconstruction with curvature sensitive keeping.

IV. MESH OPTIMIZATION

After the processing in the voxel structure, an initialized reconstructed mesh is achieved from the point cloud. The intrinsic metric of the voxel structure avoids the wrong adjacency in the initial reconstructed mesh. Geometric features such as external edges can be kept in the mesh if needed. To further improve the mesh quality, we provide a mesh optimization framework. The framework includes internal edge rebuilding and isotropic remeshing.

A. Internal Edge Rebuilding

Internal edge is one kind of important geometric features. Compared to the external edge, the internal edge is inside a mesh like a hole, which affects the mesh topology structure. Most reconstruction methods fill the internal edge by default. In our framework, an internal edge rebuilding is provided as an optional function. The difficulty of the internal edge rebuilding is that the non-uniform density distribution of a point cloud affects the internal edge detection. After the density adjusting in the voxel structure, the point cloud is transferred into a uniform density one. The initial reconstructed mesh inherits the uniform density from the voxel structure. It means the triangle of the mesh should not cross the non-adjacent voxel boxes. In Figure 11, we show the illegal triangles. Once such triangles are deleted from

triangulation method [17] to reconstruct mesh. Since the distance field is optimized with intrinsic metric, the reconstructed mesh is more accurate. In Figure 8, we show a reconstruction instance with different point numbers by user specification.

To keep important geometric features, the resampling scheme provides a convenient solution by specifying different resampling rates for different point sets. Based on the different resampling rates, the target geometric feature can be maintained in the reconstructed mesh. For instance, if we want to keep external edges, the resampling rate for the points on the external edge should be larger than other points. It guarantees the quality of external edges in the initial reconstructed mesh. The points are classified into two categories (edge points and ordinary points) with two user-specified resampling rates. The edge points can be detected by [60]. According to the rates, the local resampling is processed twice and external edges are kept. In practice, we set the proportion of two rates to 7:3. In Figure 9, we show an instance of the mesh reconstruction with external edge keeping. For curvature sensitive keeping, it can be processed by the same scheme. The points are classified by curvature value. Using
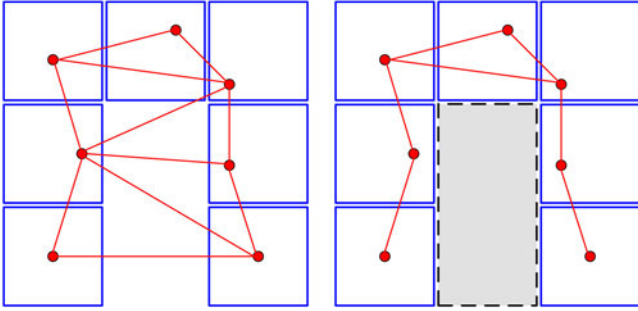
Fig. 11. An instance of illegal triangles which cross the non-adjacent voxel boxes.

Fig. 12. An instance of internal edge rebuilding.

IMR   IMR+OIR   IMR+MO

Fig. 13. Comparisons of mesh reconstruction results by IMR(initial mesh reconstruction), IMR+OIR(initial mesh reconstruction with original isotropic remeshing [47]), and IMR+MO(initial mesh reconstruction with mesh optimization).

Fig. 14. Instances of isotropic remeshing. From left to right: isotropic remesh result; isotropic remesh result with external edges; adaptive isotropic result with curvature sensitive property.
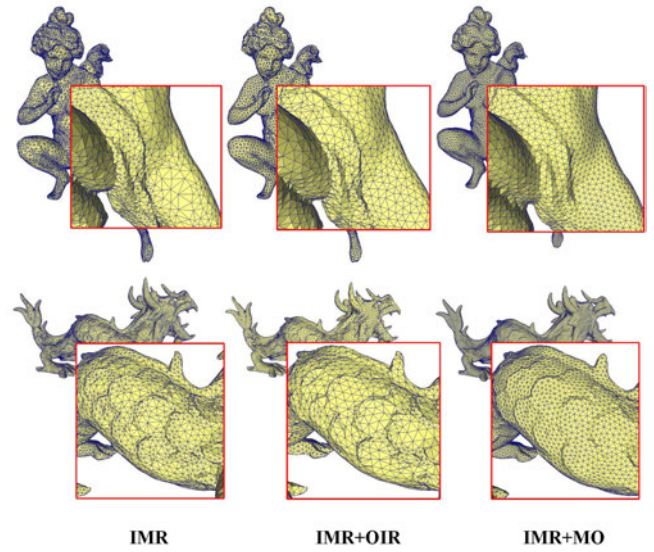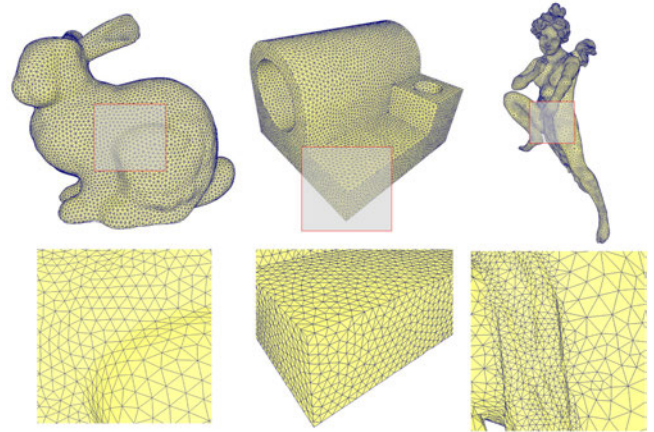
the mesh, the internal edges are rebuilt. In Figure 12, we show an instance of internal edge rebuilding.

B. Isotropic Remeshing

Isotropic remeshing is used to optimize a mesh into an isotropic one. It is constructed by three basic steps: split, collapse, and flip [47], which are used to cut the long edge, delete the short edge, and optimize the valence of the mesh after split and collapse, respectively. Compared to the original isotropic remeshing, we change the implementations to achieve better results. The important changes include point number, collapse, and flip control. In the original isotropic remeshing, the point number can not be controlled precisely. We reduce the collapse frequency in each iteration to control the point number of the mesh. For external edge keeping, some collapse and flip should not be processed to avoid the feature lost. The remeshing includes the following steps:

1) Input initial reconstructed mesh with iteration number and point classification (external edge points and ordinary points).
2) Compute the average triangle edge length l of the mesh.
3) Split the triangle edge if its length is larger than $\frac{4}{3}\tilde{}l$, count the split number i_s.
4) Collapse the triangle edge into the middle point if its length is smaller than $\frac{4}{5}\tilde{}l$. If the triangle edge includes an external edge point with more than two external edge neighbors, the collapse should not be processed. If the collapse number is equal to i_s, then stop.
5) Flip triangle edge to optimize the valence. If the flip generates obtuse angles or the flip triangle edge includes two external edge points, the flip should not be processed.

6) If the point is not the external edge point, smooth points in tangent space.
7) If the current iteration number is equal to the input one, then output the mesh. If not, loop to Step 2.

The details of valence optimization and tangent space smoothing are introduced in [47]. Based on the modification, the point number can be controlled and the generation of obtuse triangle is limited which improves the quality of the mesh. Compared with the original isotropic remeshing method [47], our mesh optimization improves the quality of meshes. In Figure 13, we compare the reconstructed meshes from initial mesh reconstruction (first step of our framework), initial mesh reconstruction with original isotropic remeshing [47], and initial mesh reconstruction with mesh optimization (our whole framework). It is clear

that our framework achieves better isotropic property. For curvature sensitive keeping, the adaptive isotropic remeshing [49] can be used to meet the requirement. The principle of adaptive isotropic remeshing is to change the length l for different triangles. The details can be found in [49]. In Figure 14, we show some isotropic remeshing instances. In summary, the mesh optimization improves the quality of the reconstructed mesh while keeping the important geometric features. Combining the initial mesh reconstruction and mesh optimization, our framework provides complete functions to solve the three key requirements of mesh reconstruction.

Fig. 15. An instance of the color map and histogram for a reconstructed mesh (Buddha).

V. EXPERIMENTS

We show the performance of our mesh reconstruction framework in this section. The experimental point cloud models were selected from Stanford and Shrec Models. We conducted the experiments on a machine equipped with Intel Xeon W 2133 3.6 G Hz, 32 GB RAM, Quadro P620, and with Windows 10 as its running system and Visual Studio 2019 (64 bit) as the development platform. Firstly, we introduce the evaluation metrics for mesh quality measurement. Secondly, we compare several classic mesh reconstruction methods based on the evaluation metrics. Finally, we show a comprehensive analysis of our method with the state of art.

A. Evaluations

To measure the quality of the reconstructed mesh, we introduce some evaluation metrics including triangle quality measurement [61], minimum angle statistics, and visualization. The triangle quality measurement $Q(t) \in [0, 1]$ evaluates the isotropic property of a triangle t from the reconstructed mesh. The minimum Q_{min} and average Q_{ave} are computed from $Q(t)$ to quantify the isotropic property. Another important evaluation metric is the minimum interior angle degree $\theta(t)$ of a triangle. The high-quality triangular mesh should avoid too large or small angles [50]. Similar to Q_{min} and Q_{ave}, θ_{min} and θ_{ave} are used for evaluation. Using histogram and color map to visualize the statistics of $\{\theta(t)\}$, the quality of the reconstructed mesh is shown clearly.

For the histogram visualization, we count all angle values to generate the histogram. For color map visualization, we compute the minimum angle of each vertex and obtain the color value by color transfer. The minimum angle of a vertex is computed by the following steps: 1. collecting the triangles $\{t_p\}$ that include the vertex p; 2. computing the angels $\{\theta_p\}$ which take p as an angle vertex; 3. transferring the angel value to color value. In Figure 15, we show an instance of the color map and histogram from a reconstructed mesh.

B. Comparisons

Based on the evaluation metrics, we compare the performance of different reconstruct methods including Scale Space [18], Screened Poisson [19], Delaunay Strategy [17] [62], Parallel CVT [13], and Gaussian kernel-based optimization [14]. The Scale Space and Delaunay Strategy are implemented with
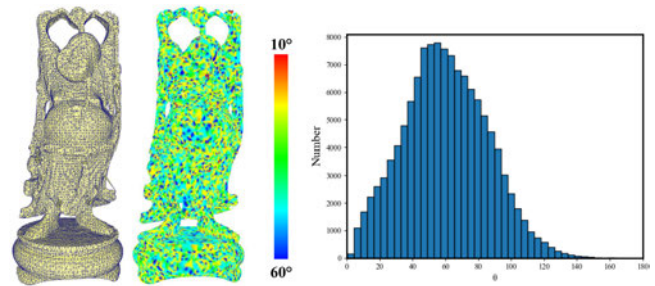
CGAL library (www.cgal.org). The Screened Poisson result is achieved from MeshLab toolbox(www.meshlab.net). The Parallel CVT and Gaussian kernel-based optimization are programmed in our platform with the assistant of CGAL library. The iteration numbers for Parallel CVT is set to 50. For L-BFGS implementation of Gaussian kernel-based optimization, a hybrid L-BFGS tool (HLBFGS) is used (xueyuhanlang.github.io/software/HLBFGS). In our framework, we set the iteration step to 5 for isotropic remeshing.

The test set contains different kinds of point clouds such as dense point clouds, sparse point clouds, and point clouds with external edges. It can be used to evaluate the performance of different reconstruction methods comprehensively. The resampling point number is set to 10 000 or 20 000. As some methods can not control the resampling point number, a point cloud resampling should be added as an assistant. The resampling method is provided by CGAL library (grid-based method). In Figure 16, we show the reconstructed meshes and their color maps obtained with different methods. Our method achieves better isotropic property in reconstructed meshes. The related θ histogram results are shown in Figure 17. In Table I and II, we show the comparisons of θ and Q for the test set. If Q is less than 0.1 or the θ is less than 5°, it means the related triangle t is a low-quality one. We label the value to "Nan"(non-acceptable value). The comparative data supports the conclusion that our method achieves better meshes. To further demonstrate the performance of external edge keeping, we compare the reconstructed meshes with external edges by different methods in Figures 18-20. It shows that our method remains the accurate external edges while keeping the isotropic property. It keeps the 2-manifold property and avoids the wrong truncation (such as the Block model by Parallel CVT).

C. Analysis

In this part, we make a further analysis for different reconstruct methods. The Scale Space is limited by the density of the point cloud. Once the point cloud has some regions with non-uniform point density, the reconstructed mesh is not satisfied (such as the Ant model in Figure 16). Using the Screened Poisson to reconstruct mesh, the 2-manifold property can be kept. The reason is the reconstruction is not limited by the local

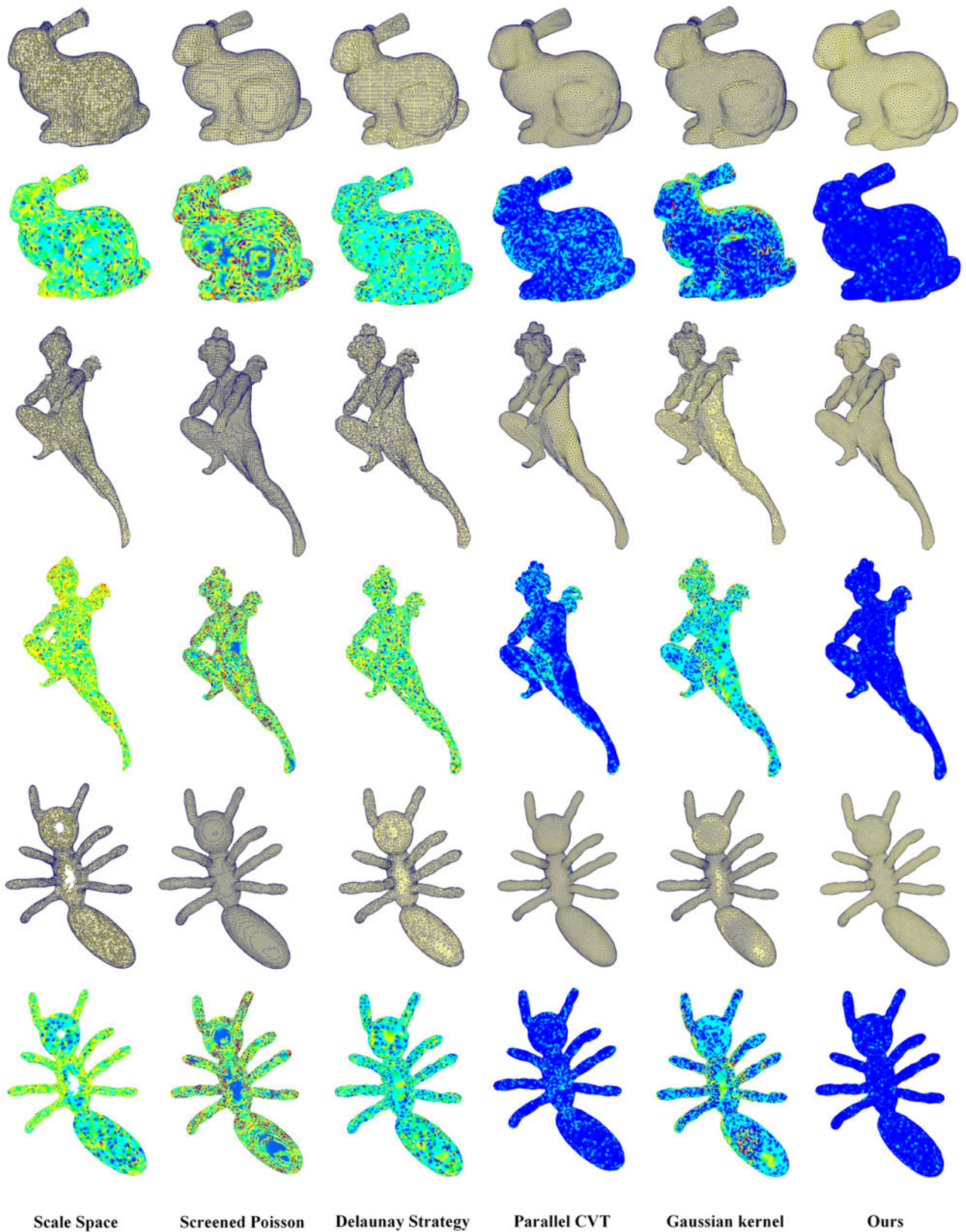| Scale Space | Screened Poisson | Delaunay Strategy | Parallel CVT | Gaussian kernel | Ours |

Fig. 16. Comparisons of the reconstructed meshes and their color maps by different methods. The point clouds from top to bottom are Bunny, Angle, and Ant.
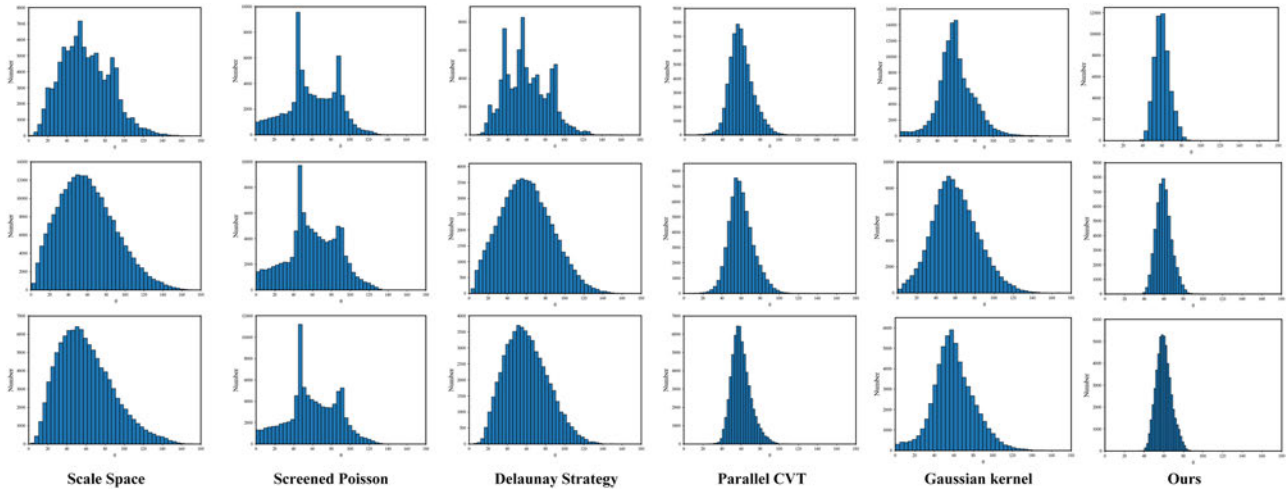
Fig. 17.   Comparisons of the θ histograms of different methods. The point clouds from top to bottom are Bunny, Angle, and Ant.

TABLE I
COMPARISONS OF MINIMUM ANGLE STATISTICS BY THE DIFFERENT METHODS

Mehtod	Scale Space		Screened Poisson		Delaunay Strategy		Parallel CVT		Gaussian kernel		Ours	
Model(Point number)	θ_{min}	θ_{avg}	θ_{min}	θ_{avg}	θ_{min}	θ_{avg}	θ_{min}	θ_{avg}	θ_{min}	θ_{avg}	θ_{min}	θ_{avg}
Bunny(10,000)	Nan	34.73°	Nan	32.75°	Nan	36.84°	11.09°	48.87°	Nan	43.61°	**36.31°**	**52.36°**
Horse(10,000)	Nan	35.29°	Nan	32.94°	Nan	36.62°	8.182°	49.05°	Nan	38.80°	**35.85°**	**52.33°**
Armadillo(10,000)	Nan	31.95°	Nan	32.26°	Nan	33.48°	9.68°	49.29°	Nan	39.66°	**27.03°**	**52.34°**
Dragon(20,000)	Nan	32.06°	Nan	32.50°	Nan	34.27°	Nan	48.84°	Nan	38.81°	**17.96°**	**52.34°**
Asian Dragon(20,000)	Nan	28.61°	Nan	32.49°	Nan	34.40°	Nan	48.25°	Nan	41.25°	**9.23°**	**52.20°**
Angel(10,000)	Nan	31.11°	Nan	32.03°	Nan	32.94°	9.01°	48.03°	Nan	37.04°	**10.16°**	**52.31°**
Buddha(20,000)	Nan	31.82°	Nan	33.21°	Nan	34.01°	Nan	47.09°	Nan	38.95°	**13.14°**	**52.29°**
Lucy(20,000)	Nan	31.44°	Nan	32.77°	Nan	35.46°	6.77°	47.84°	Nan	36.29°	**10.38°**	**52.29°**
Thai Statue(20,000)	Nan	29.81°	Nan	32.63°	Nan	34.33°	6.01°	47.23°	Nan	41.24°	**10.54°**	**52.07°**
Ant(10,000)	Nan	34.28°	Nan	32.53°	Nan	37.96°	23.11°	50.91°	Nan	40.37°	**30.71°**	**52.32°**
Centuar(10,000)	Nan	34.42°	Nan	32.32°	Nan	38.67°	20.35°	50.18°	Nan	37.61°	**34.96°**	**52.35°**
Giraffe(10,000)	Nan	33.52°	Nan	33.27°	Nan	37.09°	14.75°	50.42°	Nan	38.05°	**34.95°**	**52.33°**
Hand(10,000)	Nan	36.07°	Nan	32.59°	Nan	38.44°	18.28°	50.51°	Nan	43.90°	**33.77°**	**52.33°**
Nunchakus(10,000)	Nan	33.57°	Nan	33.10°	7.25°	37.01°	**14.09°**	49.99°	Nan	40.13°	10.48°	**51.88°**
WoodMan(10,000)	Nan	32.46°	Nan	32.58°	Nan	35.56°	10.51°	50.43°	Nan	38.44°	**32.30°**	**52.33°**
Block(10,000)	Nan	36.84°	Nan	36.21°	7.93°	41.33°	7.05°	48.45°	Nan	33.82°	**8.29°**	**51.09°**
Fandisk(10,000)	Nan	37.18°	Nan	37.70°	10.40°	42.02°	**14.81°**	49.79°	Nan	38.56°	8.12°	**51.06°**
Joint(10,000)	Nan	36.12°	Nan	37.44°	8.36°	38.38°	**26.59°**	49.56°	Nan	39.82°	13.96°	**51.28°**

TABLE II
COMPARISONS OF TRIANGLE QUALITY MEASUREMENT BY THE DIFFERENT METHODS

| Method	Scale Space		Screened Poisson		Delauny Strategy		Parallel CVT		Gaussian kernel		Ours	
Model(Point number)	Q_{min}	Q_{avg}	Q_{min}	Q_{avg}	Q_{min}	Q_{avg}	Q_{min}	Q_{avg}	Q_{min}	Q_{avg}	Q_{min}	Q_{avg}
Bunny(10,000)	Nan	0.66	Nan	0.61	0.11	0.70	0.19	0.86	Nan	0.77	**0.59**	**0.91**
Horse(10,000)	Nan	0.67	Nan	0.61	0.12	0.69	0.20	0.86	Nan	0.72	**0.62**	**0.91**
Armadillo(10,000)	Nan	0.62	Nan	0.61	0.14	0.65	0.26	0.86	Nan	0.73	**0.47**	**0.91**
Dragon(20,000)	Nan	0.62	Nan	0.61	Nan	0.66	Nan	0.86	Nan	0.72	**0.40**	**0.91**
Asian Dragon(20,000)	Nan	0.56	Nan	0.61	0.11	0.66	Nan	0.85	Nan	0.75	**0.19**	**0.91**
Angel(10,000)	Nan	0.60	Nan	0.60	Nan	0.64	0.19	0.85	Nan	0.69	**0.26**	**0.91**
Buddha(20,000)	Nan	0.61	Nan	0.62	Nan	0.65	Nan	0.84	Nan	0.72	**0.28**	**0.91**
Lucy(20,000)	Nan	0.61	Nan	0.62	Nan	0.68	0.16	0.85	Nan	0.69	**0.22**	**0.91**
Thai Statue(20,000)	Nan	0.58	Nan	0.61	Nan	0.66	0.16	0.84	Nan	0.75	**0.22**	**0.90**
Ant(10,000)	Nan	0.65	Nan	0.61	0.13	0.71	0.45	0.88	Nan	0.74	**0.56**	**0.91**
Centuar(10,000)	Nan	0.65	Nan	0.61	0.11	0.72	0.42	0.87	Nan	0.71	**0.59**	**0.91**
Giraffe(10,000)	Nan	0.64	Nan	0.62	0.13	0.70	0.33	0.88	Nan	0.71	**0.64**	**0.91**
Hand(10,000)	Nan	0.68	Nan	0.61	Nan	0.72	0.38	0.88	Nan	0.78	**0.57**	**0.91**
Nunchakus(10,000)	Nan	0.64	Nan	0.62	0.18	0.70	**0.32**	0.87	Nan	0.74	0.18	**0.90**
WoodMan(10,000)	Nan	0.62	Nan	0.61	Nan	0.68	0.22	0.88	Nan	0.72	**0.54**	**0.91**
Block(10,000)	Nan	0.69	Nan	0.63	0.22	0.75	0.18	0.85	Nan	0.65	0.18	**0.89**
Fandisk(10,000)	Nan	0.68	Nan	0.65	0.27	0.76	**0.38**	0.87	Nan	0.72	0.17	**0.89**
Joint(10,000)	Nan	0.68	Nan	0.64	0.23	0.72	**0.44**	0.86	Nan	0.73	0.28	**0.90**
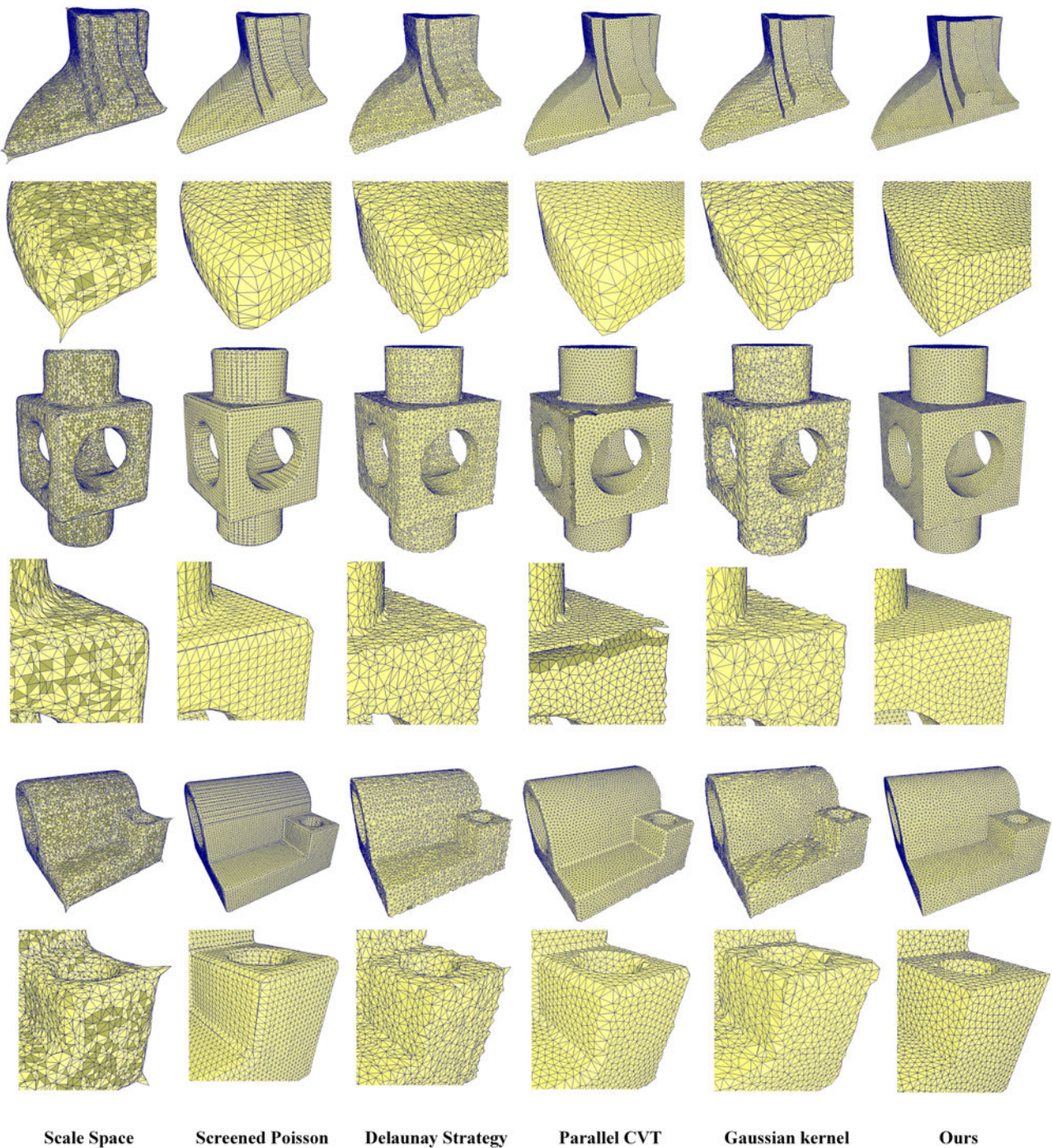
Fig. 18. Comparisons of the reconstructed meshes with external edges by different methods. The point clouds from top to bottom are Fandisk, Block, and Joint.

tangent space. The reconstruction is processed in a global optimization scheme. The advantage of the method is running faster with 2-manifold property. The drawbacks of the method are the quality of the mesh is not favorable and some geometric features are lost. The Delaunay Strategy harvests the reconstructed mesh without point movement. The core of many reconstruction methods is based on the strategy. However, it is limited by the density of point distribution. The above three methods cannot precisely control the point number in the reconstructed mesh.

As the state of the art, the Parallel CVT and Gaussian kernel-based optimization are capable of controlling the point number in the reconstructed mesh while keeping the isotropic property. However, as aforementioned, such methods lost some geometric features after points' smoothing in the local tangent space. If the user-specified resampling point number is too small, the point density is reduced. Then the distance between two adjacent points in tangent space can not be used to represent the real point distance on the mesh, which affects the accuracy
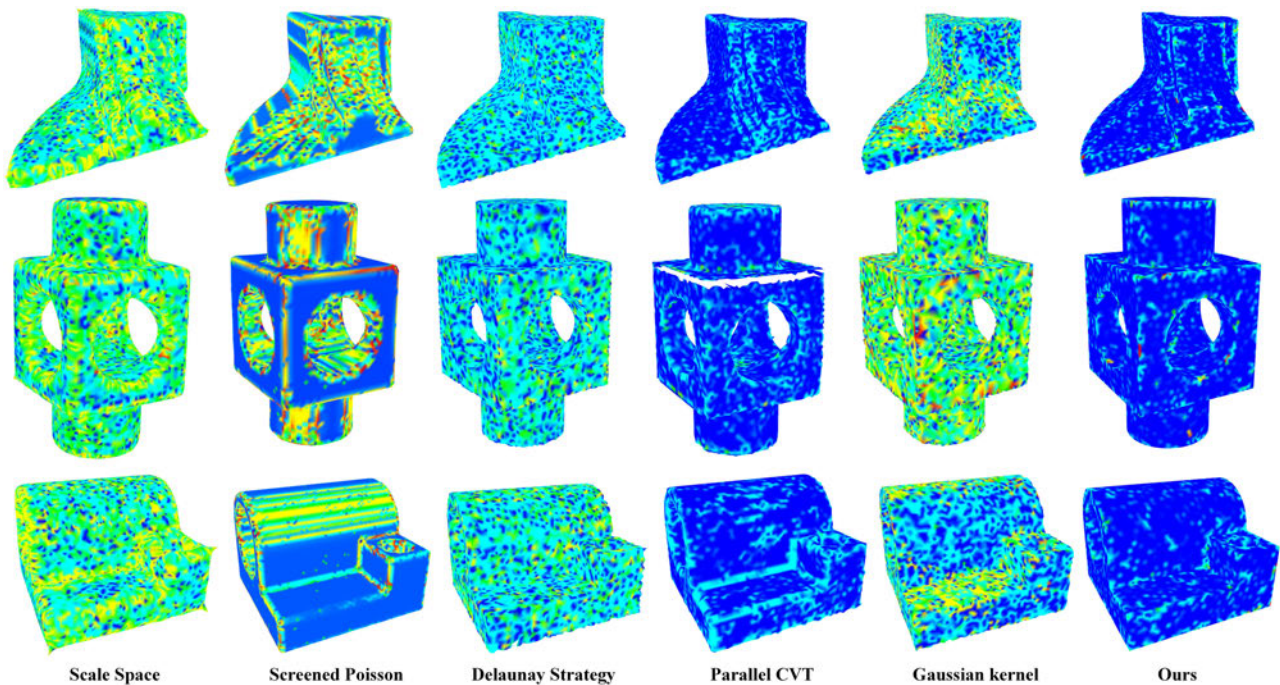
Fig. 19. Comparisons of the color maps by different methods. The point clouds from top to bottom are Fandisk, Block, and Joint.
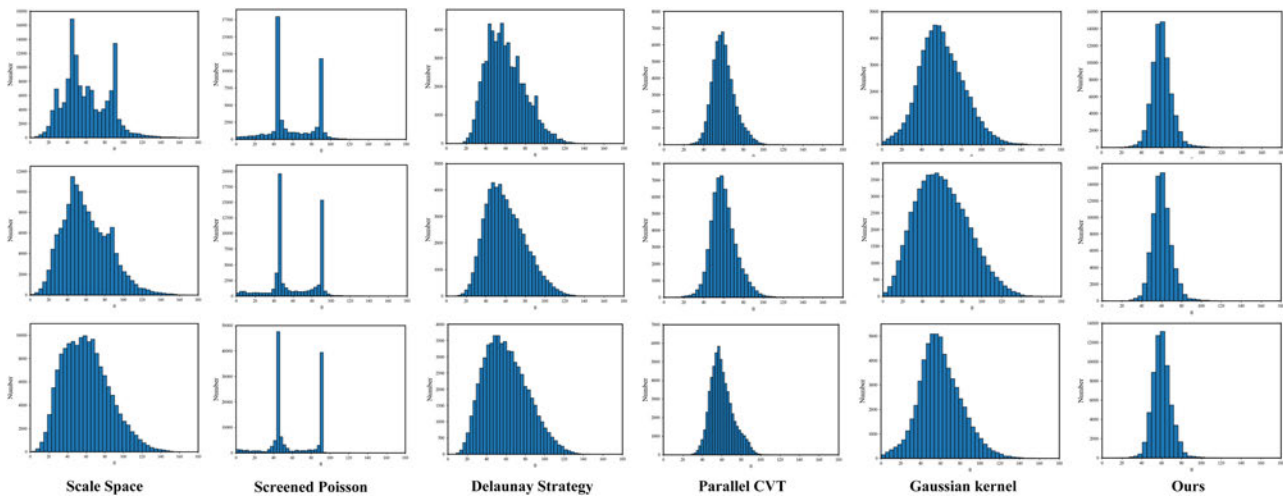

Fig. 20. Comparisons of the θ histograms of different methods. The point clouds from top to bottom are Fandisk, Block, and Joint.

of local region detection, especially for the regions with sharp curvature changes. Some instances are shown in Figure 21. In our framework, such errors are avoided by the voxel structure. In Figure 22, we show the improvement of our framework.

For time cost analysis, we compare two kinds of mesh reconstruction tasks: 1. input point clouds with different point number and output the certain resampling point number (10 000) in the reconstructed meshes; 2. fix the input point number (200 000) and output different resampling point numbers in the reconstructed meshes. The time overhead of the three methods is shown in Figure 23. For Parallel CVT, it is not sensitive to the input point number. However, it is sensitive to the output resampling point number. The time cost of the Voronoi diagram

is larger than other methods even the computation can be accelerated by a multi-core structure. Thanks to the acceleration of HLBFGS, the Gaussian kernel-based optimization speed is faster than Parallel CVT. It is not sensitive to the output or input point number. Our method achieves similar performance to the Gaussian kernel-based optimization.

In convergence analysis, we compare the Q_{ave} and θ_{ave} in different iterations of three methods. The details are shown in Figure 24. The Gaussian kernel-based optimization achieves faster convergence speed than Parallel CVT. However, it is sensitive to the initial resampling points' positions and is easy to fall into the local optimum. The values of Q_{ave} and θ_{ave} are worse. For Parallel CVT, the convergence speed is slower. The values of Q_{ave}
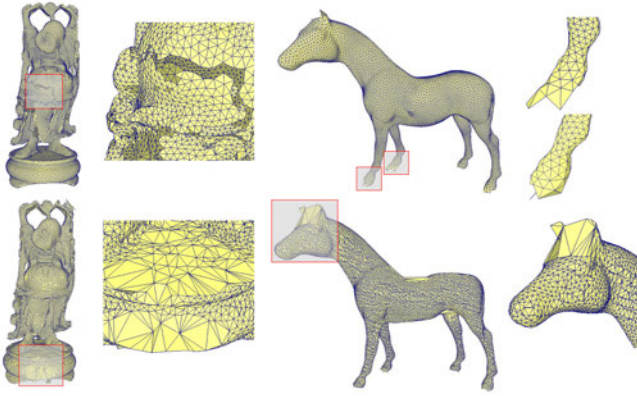
Fig. 21. Instances of the incorrect triangles in reconstruct meshes (Buddha and Horse). The top row: reconstruct meshes by Parallel CVT; the bottom row: reconstruct meshes by Gaussian kernel-based optimization.
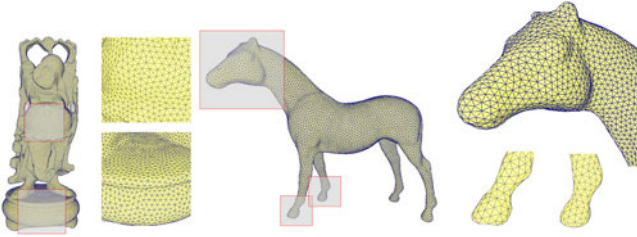

Fig. 22. Instances of our reconstructed meshes (Buddha and Horse). Based on the intrinsic metric, the incorrect triangles are removed.
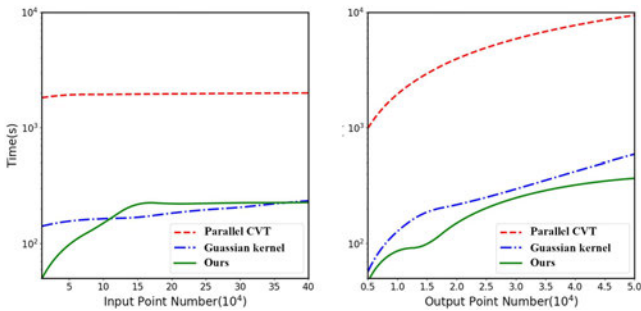

Fig. 23. The time cost of three methods.

and θ_{ave} are not converged even after 50 iterations. In contrast, our method achieves faster convergence speed with better values of Q_{ave} and θ_{ave}.

For geometric consistency evaluation, we compute the MLS error [63] based on the state of the art and our method. To achieve the isotropic reconstructed mesh, some point positions are changed inevitably. If a reconstructed mesh maintains better geometric consistency to a point cloud, then the MLS error should be approach to zero. It means that the reconstructed mesh should fit the MLS surface well. We provide MLS error reports of Parallel CVT, Gaussian kernel-based optimization, and our framework in Table III. MLS error results show that our mesh reconstruction achieves better performance for geometric consistency keeping.
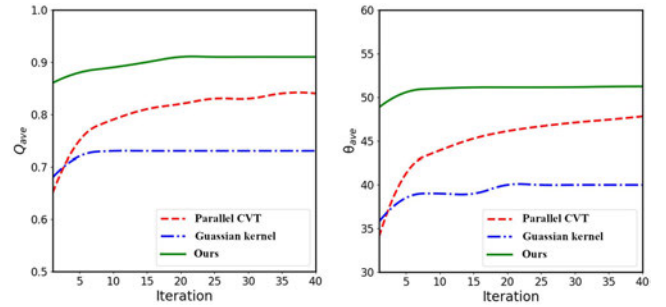

Fig. 24. The convergence reports ($Q(t)$ and $\theta(t)$) of the three methods.

TABLE III
COMPARISONS OF MLS ERROR REPORTS BY THE DIFFERENT METHODS

Methods	Parallel CVT		Particle-based		Ours	
Model	$\triangle$Max	$\triangle$Avg	$\triangle$Max	$\triangle$Avg	$\triangle$Max	$\triangle$Avg
Bunny	2.65E-2	4.12E-3	4.12E-1	1.46E-2	**2.02E-2**	**2.96E-3**
Horse	Nan	Nan	1.08E-1	5.24E-3	**1.87E-2**	**1.75E-3**
Armadillo	2.54E-2	**3.42E-3**	2.51E-2	3.54E-3	**2.27E-2**	4.68E-3
Dragon	3.24E-2	**2.74E-3**	9.33E-2	3.27E-2	**2.82E-2**	5.04E-3
Asian Dragon	**1.94E-2**	**2.33E-3**	2.10E-2	3.62E-3	2.02E-2	4.13E-3
Angel	1.95E-2	2.94E-3	4.06E-2	**2.76E-3**	**1.94E-2**	2.99E-3
Buddha	2.04E-2	3.20E-3	5.18E-2	6.24E-3	**1.81E-2**	**3.10E-3**
Lucy	2.39E-2	3.54E-3	**1.49E-2**	2.62E-3	1.79E-2	3.18E-3
Thai Statue	Nan	Nan	6.33E-2	9.28E-3	**2.01E-2**	**3.77E-3**
Ant	8.83E-3	2.35E-3	1.50E-2	2.28E-3	**8.01E-3**	**1.61E-3**
Centaur	**1.51E-2**	**2.96E-3**	2.08E-2	4.25E-3	2.06E-2	2.98E-3
Giraffe	**1.52E-2**	2.12E-3	2.25E-2	4.07E-3	2.19E-2	**1.56E-3**
Hand	**1.18E-2**	2.36E-3	1.25E-2	2.22E-3	1.67E-2	**2.14E-3**
Nunchakus	1.44E-2	2.19E-3	3.35E-2	9.27E-3	**1.05E-2**	**1.58E-3**
WoodMan	**9.98E-3**	2.53E-3	1.46E-2	3.29E-3	1.13E-2	**2.23E-3**
Block	4.47E-2	6.96E-3	**3.39E-2**	3.96E-3	3.41E-2	**3.95E-3**
Fandisk	3.06E-2	2.82E-3	2.90E-2	3.03E-3	**3.06E-2**	**2.35E-3**
Joint	5.48E-2	4.12E-3	5.86E-2	8.39E-3	**4.99E-2**	**3.07E-3**

More mesh reconstruction results based on our framework are shown in Figures 25-28. The additional test point clouds are collected from RGB-D Scenes Dataset and FRCG2.0. In Figures 25 and 26, some reconstructed meshes from RGB-D Scenes Dataset and FRGC2.0 are shown. It proves that our framework has good performance for different datasets. However, to some point clouds, our framework can not achieve accurate reconstructed meshes. Such point clouds can be divided into two categories. One kind of point cloud has complex geometric features such as delicate texture, complicated topology structure, or extremely sharp edges (both external and internal, included angle is less than $30°$). Using our framework, such geometric details may be broken to a certain degree. In Figure 27, some inaccurate reconstructed meshes are shown. Another kind of point cloud is not single object model. It contains a larger scene with multiple objects. For this kind of point cloud, our framework can not reconstruct the scene with accurate mesh. In Figure 28, we show two instances. The reason is that our framework is designed for single object. For scene reconstruction, our framework does not consider the relationships between different objects.

VI. CONCLUSION

We have introduced a voxel structure-based framework for mesh reconstruction. The proposed voxel structure provides the intrinsic metric which improves the accuracy of the initial reconstructed mesh. The point-based distance field is optimized by the voxel structure. It provides the correct point connections
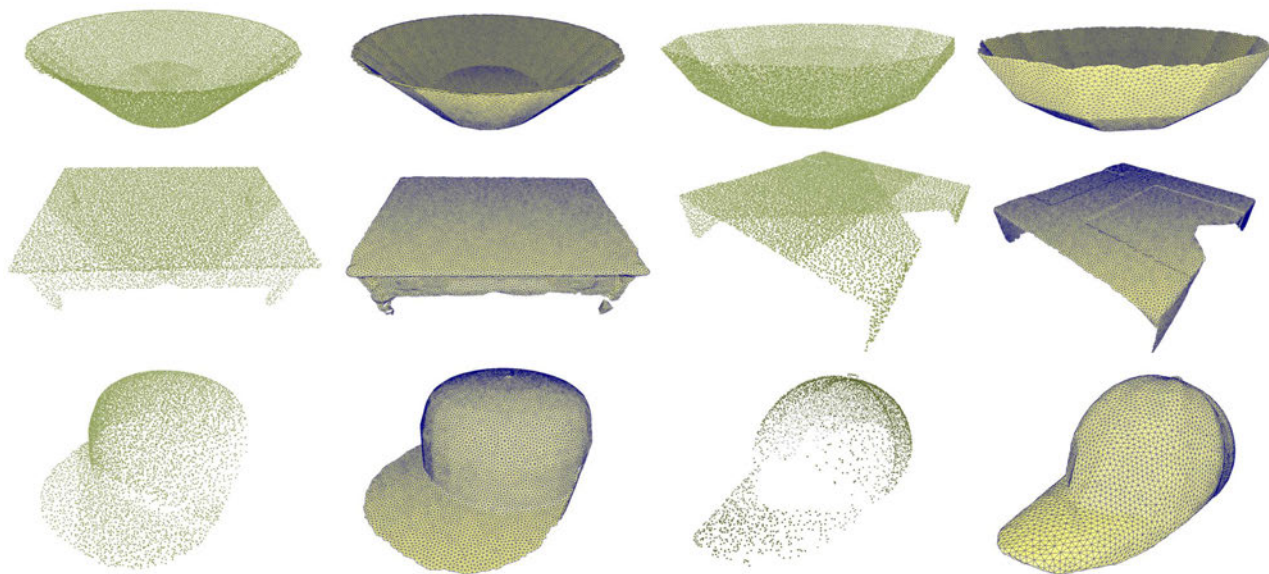
Fig. 25. Instances of our mesh reconstruction in RGB-D Scenes Dataset.

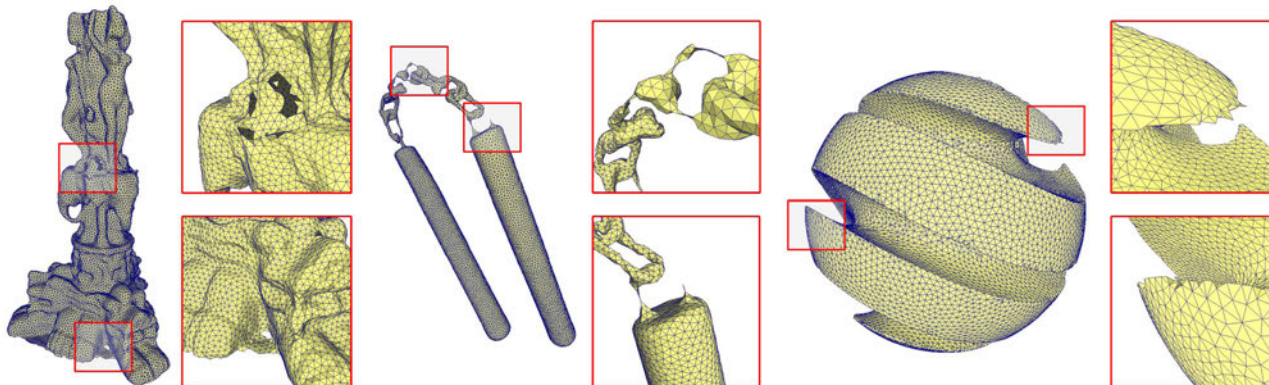Fig. 26. Instances of our mesh reconstruction in FRGC2.0.

Fig. 27. Some inaccurate instances of our mesh reconstruction in Stanford and SHREC.
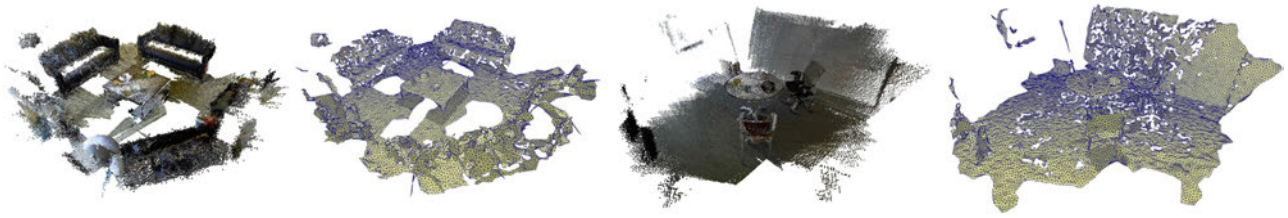
Fig. 28.   Some inaccurate instances of our mesh reconstruction in RGB-D Scenes Dataset.

and maintains the user-specified point number. With different resampling rates for different point subsets, geometric features such as external edges and curvature sensitive property can be kept in the initial reconstructed mesh. Based on the initial reconstructed mesh, a mesh optimization method is proposed to further improve the mesh quality. The experimental results show that our framework achieves better reconstructed meshes. It obtains better isotropic property and keeps important geometric features. It is also shown that our framework achieves better convergence and faster speed.

REFERENCES

[1] A. Agudo and F. Moreno-Noguer, "Shape basis interpretation for monocular deformable 3-d reconstruction," *IEEE Trans. Multimedia*, vol. 21, no. 4, pp. 821–834, Apr. 2018.

[2] I. Cheng and P. Boulanger, "Adaptive online transmission of 3-d texmesh using scale-space and visual perception analysis," *IEEE Trans. Multimedia*, vol. 8, no. 3, pp. 550–563, Jun. 2006.

[3] J.-K. Ahn, Y. J. Koh, and C.-S. Kim, "Efficient fine-granular scalable coding of 3 d mesh sequences," *IEEE Trans. Multimedia*, vol. 15, no. 3, pp. 485–497, Apr. 2012.

[4] M. Corsini, E. D. Gelasca, T. Ebrahimi, and M. Barni, "Watermarked 3-d mesh quality assessment," *IEEE Trans. Multimedia*, vol. 9, no. 2, pp. 247–256, Feb. 2007.

[5] R. Jiang, H. Zhou, W. Zhang, and N. Yu, "Reversible data hiding in encrypted three-dimensional mesh models," *IEEE Trans. Multimedia*, vol. 20, no. 1, pp. 55–67, Jan. 2017.

[6] S.-W. Jeong and J.-Y. Sim, "Saliency detection for 3 d surface geometry using semi-regular meshes," *IEEE Trans. Multimedia*, vol. 19, no. 12, pp. 2692–2705, Dec. 2017.

[7] V. Surazhsky, T. Surazhsky, D. Kirsanov, S. J. Gortler, and H. Hoppe, "Fast exact and approximate geodesics on meshes," *ACM Trans. Graph.*, vol. 24, no. 3, pp. 553–560, 2005.

[8] C. Lv, Z. Wu, X. Wang, M. Zhou, and K.-A. Toh, "Nasal similarity measure of 3 d faces based on curve shape space," *Pattern Recognit.*, vol. 88, pp. 458–469, 2019.

[9] L. Kumar, M. Shuaib, Q. Tanveer, V. Kumar, M. Javaid, and A. Haleem, "3d scanner integration with product development," *Int. J. Eng. Technol.*, vol. 7, no. 2, pp. 220–225, 2018.

[10] M. Alexa, J. Behr, D. Cohen, S. Fleishman, and D. Levin, "Point set surfaces," in *Proc. IEEE Visualization.*, 2001, pp. 21–29.

[11] S. Bouaziz, A. Tagliasacchi, H. Li, and M. Pauly, "Modern techniques and applications for real-time non-rigid registration," in *Proc. SIGGRAPH ASIA 2016 Courses*, 2016, pp. 1–25.

[12] Y. Sahillioğlu and L. Kavan, "Detail-preserving mesh unfolding for non-rigid shape retrieval," *ACM Trans. Graph.*, vol. 35, no. 3, pp. 1–11, 2016.

[13] Z. Chen, T. Zhang, J. Cao, Y. J. Zhang, and C. Wang, "Point cloud resampling using centroidal voronoi tessellation methods," *Comput.-Aided Des.*, vol. 102, pp. 12–21, 2018.

[14] S. Zhong, Z. Zhong, and J. Hua, "Surface reconstruction by parallel and unified particle-based resampling from point clouds," *Comput. Aided Geometric Des.*, vol. 71, pp. 43–62, 2019.

[15] S. K. Yadav, U. Reitebuch, and K. Polthier, "Robust and high fidelity mesh denoising," *IEEE Trans. Visualization Comput. Graph.*, vol. 25, no. 6, pp. 2304–2310, Jun. 2018.

[16] A. Samir, S. Patrick, L. Fabian, G. Stefan, and G. Michael, "Visibility-consistent thin surface reconstruction using multi-scale kernels," *ACM Trans. Graph.*, vol. 36, no. 6, pp. 187:1–187:13, 2017.

[17] D. Cohen-Steiner and F. Da, "A greedy delaunay-based surface reconstruction algorithm," *Vis. Comput.*, vol. 20, no. 1, pp. 4–16, 2004.

[18] J. Digne, J. M. Morel, C. M. Souzani, and C. Lartigue, "Scale space meshing of raw data point sets," *Comput. Graph. Forum*, vol. 30, no. 6, pp. 1630–1642, 2011.

[19] M. Kazhdan and H. Hoppe, "Screened poisson surface reconstruction," *ACM Trans. Graph.*, vol. 32, no. 3, pp. 1–13, 2013.

[20] R. Grossmann, N. Kiryati, and R. Kimmel, "Computational surface flattening: A voxel-based approach," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 4, pp. 433–441, Apr. 2002.

[21] J. R. Shewchuk, "Delaunay refinement algorithms for triangular mesh generation," *Comput. Geometry*, vol. 22, no. 1-3, pp. 21–74, 2002.

[22] T. D. Blacker and M. B. Stephenson, "Paving: A new approach to automated quadrilateral mesh generation," *Int. J. Numer. Methods Eng.*, vol. 32, no. 4, pp. 811–847, 1991.

[23] L. De Floriani, P. Magillo, E. Puppo, and D. Sobrero, "A multi-resolution topological representation for non-manifold meshes," *Comput.-Aided Des.*, vol. 36, no. 2, pp. 141–159, 2004.

[24] F. Lafarge and C. Mallet, "Creating large-scale city models from 3d-point clouds: A robust approach with hybrid representation," *Int. J. Comput. Vis.*, vol. 99, no. 1, pp. 69–85, 2012.

[25] K. Michael, B. Matthew, and H. Hugues, "Poisson surface reconstruction," in *Proc. Eurographics Symp. Geometry Process.*, vol. 7, 2006, pp. 61–70.

[26] O. Yutaka, A. Belyaev, and M. Alexa, "Sparse low-degree implicit surfaces with applications to high quality rendering, feature extraction, and smoothing," in *Proc. Eurographics Symp. Geometry Process.*, 2005, pp. 149–159.

[27] N. Yukie, O. Yutaka, and S. Hiromasa, "Smoothing of partition of unity implicit surfaces for noise robust surface reconstruction," *Comput. Graph. Forum*, vol. 28, no. 5, pp. 1339–1348, 2009.

[28] F. Simon and G. Michael, "Floating scale surface reconstruction," *ACM Trans. Graph.*, vol. 33, no. 4, pp. 1–11, 2014.

[29] J. Wang, M. M. Oliveira, and A. E. Kaufman, "Reconstructing manifold and non-manifold surfaces from point clouds," in *VIS 05. IEEE Visual.*, 2005, pp. 415–422.

[30] B. Guan *et al.*, "Voxel-based quadrilateral mesh generation from point cloud," *Multimedia Tools Appl.*, vol. 79, no. 29, pp. 20 561–20 578, 2020.

[31] N. Amenta, M. Bern, and M. Kamvysselis, "A new voronoi-based surface reconstruction algorithm," in *Proc. Comput. Graph. Interactive Techn.*, 1998, pp. 415–421.

[32] N. Amenta, S. Choi, T. K. Dey, and N. Leekha, "A simple algorithm for homeomorphic surface reconstruction," in *Proc. Int. Symp. Comput. Geometry*, 2000, pp. 213–222.

[33] N. Amenta, S. Choi, and R. Kolluri, "The power crust," in *Proc. ACM Symp. Solid Model. Appl.*, 2001, p. 249–266.

[34] C. Kuo and H. Yau, "A delaunay-based region-growing approach to surface reconstruction from unorganized points," *Comput.-Aided Des.*, vol. 37, no. 8, pp. 825–835, 2005.

[35] J. Peethambaran and R. Muthuganapathy, "Reconstruction of watertight surfaces through delaunay sculpting," *Comput.-Aided Des.*, vol. 58, pp. 62–72, 2015.

[36] W. Wang *et al.*, "Surface reconstruction from unoriented point clouds by a new triangle selection strategy," *Comput. Graph.*, vol. 84, pp. 144–159, 2019.

[37] S. Liu and Q. Chai, "Shape-optimizing and illumination-smoothing image stitching," *IEEE Trans. Multimedia*, vol. 21, no. 3, pp. 690–703, Mar. 2018.

[38] Y. Liu *et al.*, "On centroidal voronoi tessellation-energy smoothness and fast computation," *ACM Trans. Graph.*, vol. 28, no. 4, pp. 1–17, 2009.

[39] D. Yan, B. Lévy, Y. Liu, F. Sun, and W. Wang, "Isotropic remeshing with fast and exact computation of restricted voronoi diagram," in *Proc. Eurographics Symp. Geometry Process.*, 2009, pp. 1445–1454.

[40] Y. Fei, G. Rong, B. Wang, and W. Wang, "Parallel L-BFGS-b algorithm on GPU," *Comput. Graph.*, vol. 40, pp. 1–9, 2014.

[41] D. Boltcheva and B. Lévy, "Surface reconstruction by computing restricted voronoi cells in parallel," *Comput.-Aided Des.*, vol. 90, pp. 123–134, 2017.

[42] N. Ray, D. Sokolov, S. Lefebvre, and B. Lévy, "Meshless voronoi on the gpu," *ACM Trans. Graph.*, vol. 37, no. 6, pp. 1–12, 2018.

[43] C. Luo, X. Ge, and Y. Wang, "Uniformization and density adaptation for point cloud data via graph laplacian," in *Comput. Graph. Forum*, vol. 37, no. 1, 2018, pp. 325–337.

[44] H. Fan, Y. Yu, and Q. Peng, "Robust feature-preserving mesh denoising based on consistent subneighborhoods," *IEEE Trans. Vis. Comput. Graph.*, vol. 16, no. 2, pp. 312–324, Mar.–Apr. 2009.

[45] P.-S. Wang, Y. Liu, and X. Tong, "Mesh denoising via cascaded normal regression.," *ACM Trans. Graph.*, vol. 35, no. 6, pp. 232:1–232:12, 2016.

[46] J. Wang, J. Huang, F. L. Wang, M. Wei, H. Xie, and J. Qin, "Data-driven geometry-recovering mesh denoising," *Comput.-Aided Des.*, vol. 114, pp. 133–142, 2019.

[47] M. Botsch and L. Kobbelt, "A remeshing approach to multiresolution modeling," in *Proc. Eurographics/ACM SIGGRAPH Symp. Geometry Process.*, 2004, pp. 185–192.

[48] B. Lévy and Y. Liu, "l_p centroidal voronoi tessellation and its applications," *ACM Trans. Graph.*, vol. 29, no. 4, pp. 1–11, 2010.

[49] M. Dunyach, D. Vanderhaeghe, and M. Botsch, "Adaptive remeshing for real-time mesh deformation," in *Proc. Eurographics Short Papers*, 2013, pp. 29–32.

[50] Y. Wang *et al.*, "Isotropic surface remeshing without large and small angles," *IEEE Trans. Visualization Comput. Graph.*, vol. 25, no. 7, pp. 2430–2442, Jul. 2018.

[51] X. Du *et al.*, "Field-aligned isotropic surface remeshing," in *Proc. Comput. Graph. Forum*, vol. 37, no. 6, 2018, pp. 343–357.

[52] T. Ju, "Robust repair of polygonal models," *ACM Trans. Graph.*, vol. 23, no. 3, pp. 888–895, 2004.

[53] F. Hétroy, S. Rey, C. Andújar, P. Brunet, and A. Vinacua, "Mesh repair with user-friendly topology control," *Comput.-Aided Des.*, vol. 43, no. 1, pp. 101–113, 2011.

[54] Q. Zhou, E. Grinspun, D. Zorin, and A. Jacobson, "Mesh arrangements for solid geometry," *ACM Trans. Graph.*, vol. 35, no. 4, pp. 1–15, 2016.

[55] L. Chu, H. Pan, Y. Liu, and W. Wang, "Repairing man-made meshes via visual driven global optimization with minimum intrusion," *ACM Trans. Graph.*, vol. 38, no. 6, pp. 1–18, 2019.

[56] G. Lavoué and M. Corsini, "A comparison of perceptually-based metrics for objective evaluation of geometry processing," *IEEE Trans. Multimedia*, vol. 12, no. 7, pp. 636–649, Nov. 2010.

[57] D. Levin, "Mesh-independent surface interpolation," in *Proc. Geometric Model. Sci. Visualization*, 2004, pp. 37–49.

[58] R. Schnabel and R. Klein, "Octree-based point-cloud compression," in *Proc. Eurographics Symp. Point-Based Graph.*, vol. 6, 2006, pp. 111–120.

[59] C. Moenning and A. N. Dodgson, "Fast marching farthest point sampling for implicit surfaces and point clouds," *Comput. Lab. Tech. Rep.*, vol. 565, pp. 1–12, 2003.

[60] Q. Mérigot, M. Ovsjanikov, and L. J. Guibas, "Voronoi-based curvature and feature estimation from point clouds," *IEEE Trans. Vis. Comput. Graph.*, vol. 17, no. 6, pp. 743–756, Jun. 2010.

[61] P. J. Frey and H. Borouchaki, "Surface mesh quality evaluation," *Int. J. Numer. Methods Eng.*, vol. 45, no. 1, pp. 101–118, 1999.

[62] F. Lafarge and P. Alliez, "Surface reconstruction through point set structuring," *Comput. Graph. Forum*, vol. 32, pp. 225–234, 2013.

[63] M. Alexa and A. Adamson, "On normals and projection operators for surfaces defined by point sets," in *Proc. Eurographics Symp. Point-Based Graph.*, vol. 4, 2004, pp. 149–155.

Chenlei Lv received the Ph.D. degree from the College of Information Science and Technology, Beijing Normal University. He is currently a Postdoctor with School of Computer Science and Engineering, Nanyang Technology University. He has authored or coauthored several papers in *Pattern Recognition, ACM Transactions on Multimedia Computing Communications and Applications, Pattern Recogniton Letter*, etc. His research interests include computer vision, 3D biometrics, computer graphics, discrete differential geometry and conformal geometric.
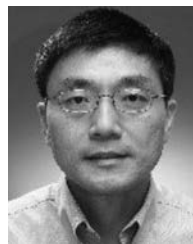
Weisi Lin (Fellow, IEEE) received the Ph.D. degree from King's College London, U.K. He is currently a Professor with the School of Computer Science and Engineering, Nanyang Technological University. His research interests include image processing, perceptual signal modeling, video compression, and multimedia communication, in which he has authored or coauthored more than 200 journal papers, more than 230 conference papers, filed seven patents, and authored two books. He is a fellow of the IET and an Honorary Fellow of the Singapore Institute of Engineering Technologists. He was the Technical Program Chair of the IEEE ICME 2013, PCM 2012, QoMEX 2014, and the IEEE VCIP 2017. He has been an Invited/Panelist/Keynote/Tutorial Speaker at more than 20 international conferences and was a Distinguished Lecturer of the IEEE Circuits and Systems Society from 2016 to 2017 and the Asia-Pacific Signal and Information Processing Association (APSIPA) from 2012 to 2013. He has been an Associate Editor of the IEEE TRANSACTIONS ON IMAGE PROCESSING, IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, IEEE TRANSACTIONS ON MULTIMEDIA, and IEEE SIGNAL PROCESSING LETTERS.
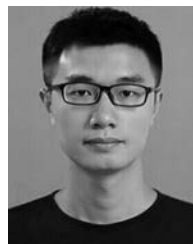
Baoquan Zhao received the Ph.D. degree in computer science from Sun Yat-sen University, Guangzhou, China, in 2017. He is currently a Research Fellow with the School of Computer Science and Engineering, Nanyang Technological University, Singapore. He was a Reviewer and Technical Program Committee Member of several journals and international conferences. His research interests include point cloud processing and compression, visual information analysis, multimedia systems and applications, etc. He was the recipient of the Outstanding Reviewer Award of 2020 IEEE ICME.