

Intrinsic and Isotropic Resampling for 3D Point Clouds

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

Abstract—With rapid development of 3D scanning technology, 3D point cloud based research and applications are becoming more popular. However, major difficulties are still exist which affect the performance of point cloud utilization. Such difficulties include lack of local adjacency information, non-uniform point density, and control of point numbers. In this paper, we propose a two-step intrinsic and isotropic (I&I) resampling framework to address the challenge of these three major difficulties. The efficient intrinsic control provides geodesic measurement for a point cloud to improve local region detection and avoids redundant geodesic calculation. Then the geometrically-optimized resampling uses a geometric update process to optimize a point cloud into an isotropic or adaptively-isotropic one. The point cloud density can be adjusted to global uniform (isotropic) or local uniform with geometric feature keeping (being adaptively isotropic). The point cloud number can be controlled based on application requirement or user-specification. Experiments show that our point cloud resampling framework achieves outstanding performance in different applications: point cloud simplification, mesh reconstruction and shape registration. We provide the implementation codes of our resampling method at <https://github.com/vvwo/I-resampling>.

Index Terms—Isotropic Resampling, Intrinsic Resampling, Point Cloud Simplification, Mesh Reconstruction, Shape Registration.

1 INTRODUCTION

WITH the rapid development of 3D scanning technology [1] [2], 3D point clouds are becoming more popular and widely used in different applications such as indoor modeling [3] [4], geographic remote sensing [5], autopilot [6], localization [7] [8], face recognition [9], etc. Compared to traditional image data, 3D point clouds contains complete geometric information, which improve the performance of computer vision applications [10] [11]. The real world acquisition of point clouds is expected to provide a bridge of computer vision and computer graphics which were two largely separated fields in the past.

Although there are many advantages of 3D point clouds, some difficulties still exist and restrict wide utilization of point cloud data. The difficulties include: 1. lack of local adjacency information; 2. non-uniform point density; 3. control of point numbers. These difficulties are explained as follows. Firstly, the adjacency between points is unknown in a point cloud. Without information on adjacency, local regions of a point cloud are unknown and almost all continuous surface-based applications are not effective [12] [13]. Secondly, using 3D scanning devices with lower resolution in wild scene will inevitably produce non-uniform density regions which reduce the performance in point cloud-based applications. Finally, different point numbers of point clouds cannot be controlled during the scanning. In some cases, point clouds with large volume of redundant points increases the storage and calculation cost. In other cases, point clouds with insufficient points cannot support the accurate geometric feature analysis. For some applications, point clouds should

be uniformed into same point number for feature training such as point cloud registration [14] [15] and deep feature learning framework [16] [17]. Therefore, the point cloud number should be controlled. To solve the difficulties above, point cloud resampling is used.

The point cloud resampling is to achieve a regular alternative from the raw point cloud. It aims to maintain geometric consistency with the raw point cloud and has uniform density with certain point number. According to the moving least squares (MLS) surface theory [18], a continuous MLS surface can be defined based on local regions of a point cloud. In order to obtain the geometric consistency in resampling result, the local region of each point should be detected while the MLS surface from resampling result should fit the raw point cloud. Some resampling methods [19] [20] introduce the local region detection and uniform density adjusting in tangent space. However, most of them do not provide intrinsic control, which means that distances between points may be off the MLS surface. It leads unpredictable errors for local region detection in resampling, especially for areas with sharp curvature changes. Moening provided a solution [21] for intrinsic resampling which is feasible in theory. However, the computation cost is huge for geodesic path searching in a point cloud. To the best of our knowledge, there is no resampling solution to solve all difficulties at the same time.

In this paper, we propose an I&I resampling framework to address the three aforementioned difficulties. As shown in Figure 1, it is a two-step framework that includes efficient intrinsic control and geometrically-optimized resampling. Firstly, efficient intrinsic control is used to provide geodesic measurement for a point cloud. Fundamentally, it inherits the advantages of traditional intrinsic control by geodesic path searching and improves its efficiency. Secondly, the geometrically-optimized resampling is to adjust a

• Chenlei Lv and Weisi Lin were with the School of Computer Science and Engineering, Nanyang Technological University. Baoquan Zhao was with the School of Artificial Intelligent, Sun Yat-sen University. The corresponding author is Weisi Lin, e-mail: (wslin@ntu.edu.sg).

Manuscript received XX Jan. 2021.

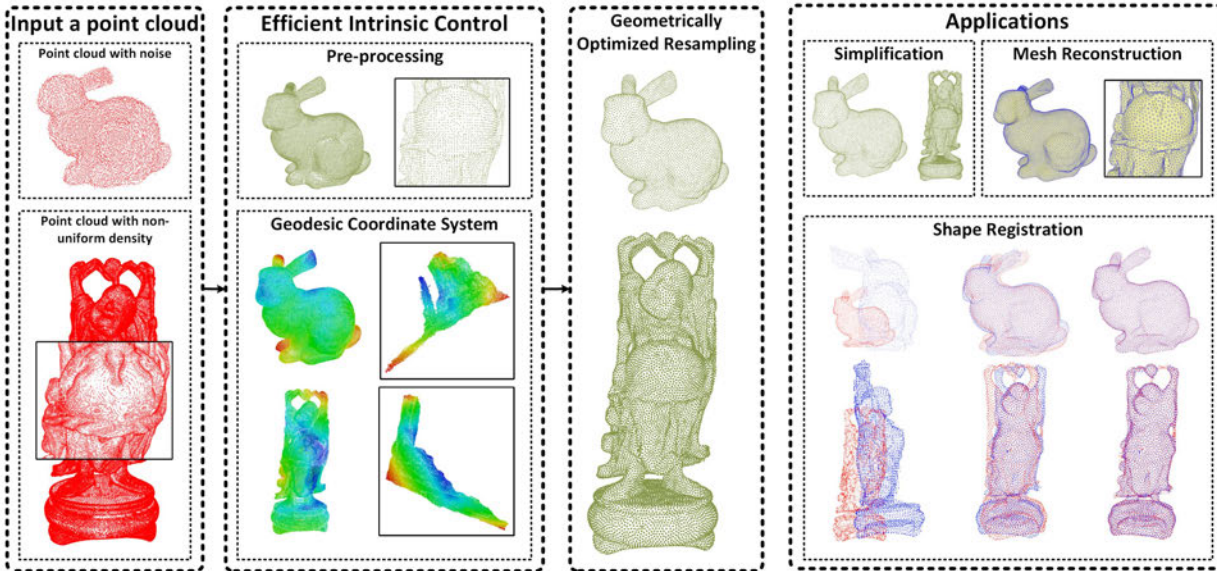


Fig. 1. The pipeline of the proposed point cloud resampling method.

raw point cloud with non-uniform density into an isotropic or adaptively-isotropic one. It solves the problems of traditional isotropic resampling methods with a geometric update process. The resampling also provides adaptively-isotropic resampling which balances the isotropic resampling and geometric feature keeping. Some geometric features like curvature sensitive (point density is proportional to curvature) and sharp edges can be kept in the resampling result. Using the I&I resampling framework, a raw point cloud can be denoised and simplified into an resampling result with accurate local regions, isotropic property and the point number specified by users or applications [15] [16]. The contributions of this paper can be summarized as:

- An efficient intrinsic control scheme is proposed to provide the geodesic measurement for a point cloud. The intrinsic control is based on the proposed geodesic coordinate system which inherits advantages of traditional methods and avoids redundant geodesic distance calculation. Benefited from the improvement of efficiency, our intrinsic control makes the intrinsic resampling more practical.
- A geometrically-optimized resampling method is designed. Following a geometric update process, a point cloud can be optimized into an isotropic or adaptively-isotropic one. It does not require complex optimization for density adjusting and avoids the local optimum resampling. The convergence speed of the method is fast. For geometric feature keeping, the adaptively-isotropic resampling is provided as an addition, which achieves a balance between geometric consistency and isotropic property with a certain point number. The quality of raw point clouds can be greatly improved.
- Applications of our I&I resampling framework are demonstrated, including point cloud simplification, mesh reconstruction and shape registration. Benefited from the I&I resampling, such applications achieve significant improvement in performance

without complex feature analysis or optimization.

The rest of the paper is organized as follows. In Sec. 2, we introduce some classical works for point cloud resampling. In Sec. 3, we describe the implementation of efficient intrinsic control with point cloud pre-processing. The geometrically-optimized resampling and related applications are introduced in Secs. 4 and 5. We demonstrate the effectiveness and efficiency of our method with extensive experimental evidence in Sec. 6, while Sec. 7 concludes the paper.

2 RELATED WORKS

There are many methods in the literature for point cloud resampling, which can be classified into four groups: grid based resampling, feature clustering based resampling, intrinsic resampling and distance field optimization based resampling.

Grid based resampling methods utilize grid structure or similar spatial organization to simplify a point cloud, such as grid structure [22], Poisson disk sampling [23] [24], octree [25], and kd-tree [26]. The advantages of such methods are that they are simple to implement, lower computation cost and have the potential to achieve rough uniform density by radius controlling of grids. Benefited from the efficiency of such methods, some related applications can be implemented with high practicality, including Screened Poisson-based mesh reconstruction [27], Grid-based simplification [22], and point cloud normalization for deep learning frameworks [28] [29]. However, the drawbacks are also significant. Firstly, they are likely to lose important geometric features of the raw point cloud such as sharp edges and curvature sensitive. Secondly, without local region detection, they would be unable to achieve isotropic resampling. Furthermore, without intrinsic control, they may violate the geometric consistency. In most cases, such methods are used to obtain an initial resampling result but not final one.

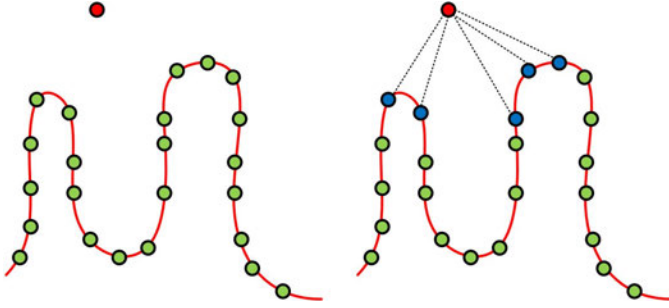


Fig. 2. An instance of neighbor searching error of outlier. The red point is the outlier, blue points represent neighbor points of the outlier by k -nearest searching. It is clear that the blue points can not be used to represent a continuous local region and the outlier can not be mapped on the MLS surface.

Geometric feature based clustering methods attempt to keep the geometric consistency of resampling. The adjacent points with similar geometric features are clustered into a group and one center point is selected to represent the points from the group. These methods differ from the selection of geometric features such as curvature [30] [31], normal vector [32] and locally optimal projection (LOP) operator [33]. They keep the geometric consistency by geometric feature based clustering. Such frameworks are used to improve the performance for some applications, including registration [34] [35] and simplification [31]. However, their performance is limited by the clustering method. Without intrinsic control, resampling errors appear in the areas with sharp curvature changes, which is similar to the grid based resampling methods.

Intrinsic Resampling methods are proposed to add intrinsic control in resampling. These methods measure distances of points on the MLS surface by geodesic path. The geodesic path represents real distance between two points and improves the accuracy of local region detection. It has been discussed that Moening provided intrinsic resampling in previous work [21]. He used the geodesic distance from Fast Marching method to achieve an intrinsic resampling result from an input point cloud. Similar to [21], some methods attempt to use geodesic distance to instead Euclidean distance such as geodesic based Poisson disk resampling [36] and Geodesic Voronoi Diagrams [37] [38]. However the methods achieve the intrinsic control in resampling at the expense of high computational efficiency, which limits their practical application.

Distance field optimization based resampling methods can be regarded as the mainstream solutions for resampling task, which include Laplace graph optimization [39] [40], Centroidal Voronoi Tessellation [41] [42] [43], and particle-based resampling [44] [45] [20]. Such methods optimize a distance field of a point cloud. The distance field is constructed by distances between all points and their neighbors. Once the distance field is optimized which means that distances between points and their neighbors are almost equal, the isotropic resampling is finished. Using the resampling, the local region can be detected in local tangent space and the point density can be adjusted to be isotropic. Such methods have been applied in mesh reconstruction [46] [42]. However, the resampling in local tangent space cannot



Fig. 3. An instance of denoising result. The red point cloud is the input point cloud with noise; the green point cloud is the denoising result.

be regarded as an intrinsic resampling and the resampling errors cannot be avoided in the areas with sharp curvature changes. Another serious problem is that they are likely to trap in the local optimum, especially for the optimization without accurate detection of point neighbors. In summary, such methods cannot satisfy all requirements of resampling as mentioned before.

Our resampling method is proposed to achieve intrinsic and isotropic resampling at the same time. With the efficient intrinsic control, the computation cost of geodesic measurement is reduced while improving the accuracy of local region detection. To avoid local optimum production, we introduce a geometrically-optimized resampling that is inspired by isotropic remeshing. Benefited from the two-step framework, our resampling is more robust and with faster convergence speed.

3 EFFICIENT INTRINSIC CONTROL

As aforementioned, intrinsic control is important for resampling. The difficulty of the intrinsic control algorithm's design is how to balance the efficiency and accuracy of geodesic distance computation. In this section, we introduce the efficient intrinsic control scheme that provides the geodesic measurement for a point cloud in an efficient way. Using three geodesic distance fields to transfer a point cloud into the geodesic coordinate system, distances between points are under the intrinsic control without redundant geodesic calculation like [21].

3.1 Pre-processing

Before constructing the efficient intrinsic control, a point cloud pre-processing is needed. The influence of noisy points should be reduced and some areas with non-uniform density should be adjusted. Such situations affect the accuracy of the geodesic coordinate system. For denoising, we use a MLS projection operator to smooth the point cloud [47]. The MLS (moving least squares) surface is used to define a continuous surface from discrete points. The noisy points are mapped onto the MLS surface based on the MLS surface fitting error which is computed by [47]

$$p_i - p_i' = a(n_i)^T (a(p_i) - p_i) a(n_i), \quad (1)$$

$$a(p_i) = \sum_{j=1}^k \frac{\theta(\|p_i - p_j\|) p_j}{\sum_{j=1}^k \theta(\|p_i - p_j\|)}, \quad (2)$$

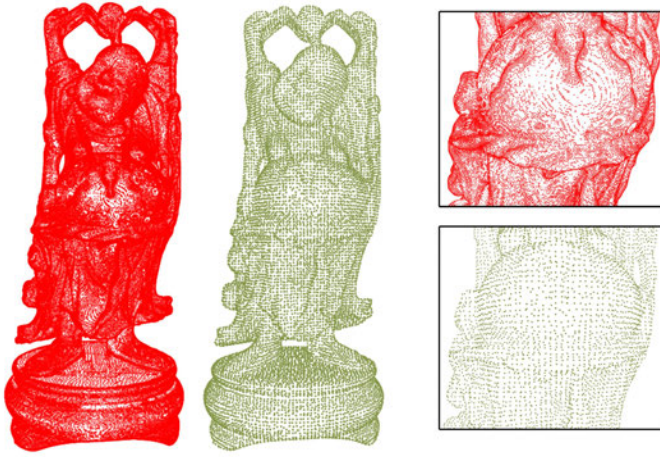


Fig. 4. An instance of density adjustment. The red point cloud is the input point cloud with non-uniform density. The green point cloud is processed by octree with uniform density.

where p_i is a point of a point cloud P , p'_i is the new position of p_i after mapping onto the MLS surface of P . The mapping can be regarded as a “pulling back” process from p_i to p'_i . A set of neighbor points $\{p_j\}$ of p_i is used for computation of p'_i . The weighted average function $a(p_i)$ also can be used to compute weighted average normal $a(n_i)$. For selection of weighted function θ , we use a Gaussian function which is suggested in [48], represented as

$$\theta(d) = e^{-\frac{d^2}{h^2}}, \quad (3)$$

where h represents searching radius in the local region of a point, which is equal to the largest distance of all points in a point cloud to their k -th neighbor (the neighbor point number is set to 8 by default). The d represents the point distance.

During the iterative update for each point by Equation 1, noisy points are smoothed by its neighbors. In most cases, it achieves better denoising performance than classical methods such as bilateral filter [49] and Gaussian filter [50] (results are shown in experiment). A special situation should be processed independently that is to outlier in a point cloud. For k -neighbor searching in a point cloud, the kd-tree is used to increase the neighbor searching speed. However, there is no guarantee that using kd-tree can achieve correct neighbor points for an outlier. An instance is shown in Figure 2. To avoid such error, we add an outlier checking program that is represented as

$$P_{outlier} = \{p_o | p_j \in K(p_o), p_o \notin K(p_j)\}, \quad (4)$$

where K means the k -neighbor searching set of the point ($|K| = 8$ by default). Once a neighbor point p_j of p_o satisfies Equation 4, then p_o is judged as an outlier. The method checks each point in the neighbor set $K(p_o)$ of p_o according to Equation 4. Even there have some points of $K(p_o)$ are outliers, the result is not changed. In Figure 3, we show a denoising instance. If a point is an outlier, the neighborhood relations between the point and its k -neighbor points are

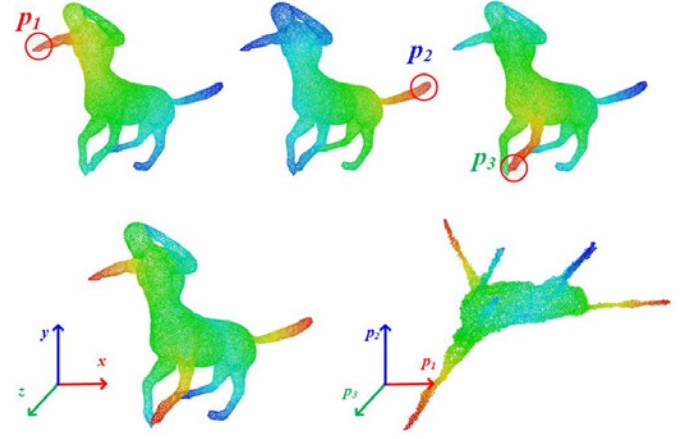


Fig. 5. An instance of the geodesic coordinate system construction. Three pictures in first row represents the satellite detection. Two pictures in second row represents the mapping from a point cloud in Euclidean space into the geodesic coordinate system.

not dual. To “pull back” the outlier onto the MLS surface, the new weighted average function is provided as

$$a(p_o) = \sum_{jn=1}^k \frac{\theta(\|p_o - p_{jn}\|)p_{jn}}{\sum_{j=1}^k \theta(\|p_o - p_{jn}\|)}, \quad (5)$$

where p_o is a outlier that belongs to $P_{outlier}$. Compared to Equation 2, the difference is that the neighbor point set $\{p_{jn}\}$ is selected based on p_n but not the p_o itself. p_n is the nearest point of p_o and $p_n \notin P_{outlier}, p_{jn} \notin P_{outlier}$. Using the new “pulling back” function for $P_{outlier}$, the neighbor searching error of outlier can be avoided.

Another target of the pre-processing is adjusting areas with non-uniform density in a point cloud. It improves the accuracy of the geodesic coordinate system and the computational efficiency in following steps. We use an octree sampling to be the adjustment. The radius of octree is the global average distance between each point to its k -neighbor points ($k = 3$ by default). We leave one point from each voxel box of octree, which is closest to the center. The octree sampling can be exploited as an effective initialization for density adjustment. In Figure 4, we show an instance.

3.2 Intrinsic Control Scheme

After pre-processing, we introduce the details of efficient intrinsic control. It is processed in the geodesic coordinate system inspired by [51]. In the geodesic coordinate system, distances between points reflect geodesic distance and avoid the redundant geodesic path computation. A concise and efficient method for the system construction is computing three geodesic distance fields to achieve a three-satellite positioning. A point can be relocated by three distances between itself and the three satellites. For a point cloud, if we detect three points as three satellites and compute corresponding geodesic distance fields, an intrinsic location is achieved, which can be regarded as the point cloud representation in the geodesic coordinate system. According to the wave propagate property [52] [53], adjacency points should be close in different geodesic distance fields. It means that a local region detection or adjacency points judgment

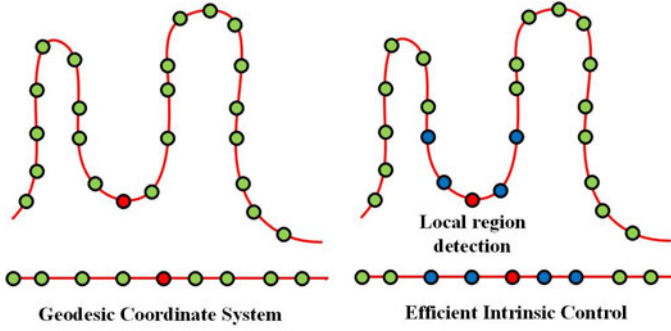


Fig. 6. An instance of local region detection with efficient intrinsic control. The geodesic coordinate system from a 2D curve can be regarded as a 1D distance field. It provides the efficient intrinsic control to detect the neighbor points (blue points) for the red point. It is clear that the efficient intrinsic control is useful for accurate neighbor points judgment.

should be satisfied in Euclidean space and geodesic coordinate system at the same time. If the adjacency between two points is judged as correct in Euclidean space but not in geodesic coordinate system, then the points are not adjacency. It is an efficient implementation of intrinsic control.

To achieve the geodesic coordinate system, two core issues are geodesic distance computation in a point cloud and satellite detection. For geodesic distance computation, the fast marching [52] can be used in point clouds with the balance of efficiency and accuracy. Compared to heat flow [54], it does not require to analysis cotangent weights in the local region of a point. To implement the fast marching in point cloud, the neighbor structure should be defined at first. We construct the kd-tree [55] for the point cloud and use k -nearest neighbor searching [56] to search point neighbors, which can be used to define the neighbor structure and drive the fast marching to compute the geodesic distance iteratively.

Based on the fast marching method, we propose the satellite detection. Theoretically, we can randomly select three points without linear relationships to be satellites. In our framework, we detect different satellites as far as possible. The reason is that if satellites are detected in a small region, geodesic distances of points are more close. It reduces effect of intrinsic control from the geodesic coordinate system. We use farthest point sampling with geodesic distance [21] to detect satellites: 1. select a point randomly as the initial point; 2. use farthest point sampling to detect other three points (p_1, p_2, p_3) by geodesic distance, and achieve three geodesic distance field (g_1, g_2, g_3); 3. a point p_i is mapped into the geodesic coordinate system based on three fields $g_1(p_i)$, $g_2(p_i)$, and $g_3(p_i)$. In Figure 5, we show an instance of the geodesic coordinate system construction. In the geodesic coordinate system, the efficient intrinsic control is implemented. It improves the accuracy of the local region detection. In Figure 6, we show an instance of the improvement.

In most cases, the point cloud has one-to-one relationship to its representation in the geodesic coordinate system. However, there has a special case, that is when the three satellites are distributed in one geodesic path, the one-to-one relationship is broken. Such case will inevitably happen in some point clouds such as sphere models. Using satellite

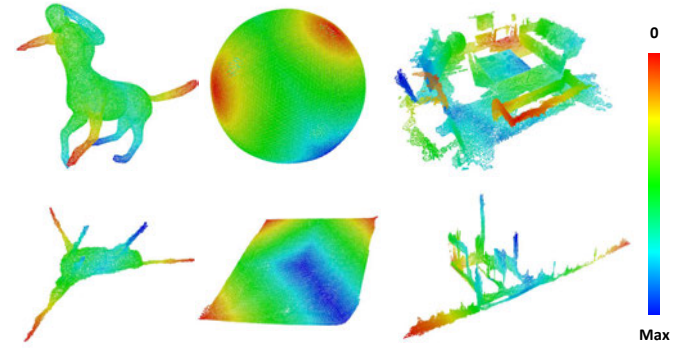


Fig. 7. Some instances of point cloud representation in the geodesic coordinate system. The color maps represent geodesic distance fields constructed by satellites (red).

detection to fix two poles, the third point must be located in the geodesic path (spherical warp) between the poles. If two points are mirror images of each other and the mirror plane is constructed by the three satellites, the two points have same coordinates in the geodesic coordinate system but not the same Euclidean coordinate to share a same neighbor region. Then, the geodesic neighbors of one point are detected as the neighbors of its mirror point at the same time, and this produces wrong geodesic neighbors in the geodesic coordinate system. Therefore, for such a point, we can use the Euclidean adjacency to detect the wrong geodesic neighbors produced by mirror points, so that accurate geodesic neighbor detection can be implemented even for this special case. It can be regarded as the Euclidean searching to avoid the mirror points in neighbor detection. In Figure 7, some representations in the geodesic coordinate system are exhibited from different point cloud models, including sphere and multi-objects models.

4 GEOMETRICALLY-OPTIMIZED RESAMPLING

Based on the efficient intrinsic control, we propose a geometrically-optimized resampling method, which includes an isotropic resampling and an adaptively-isotropic resampling. The core issue for isotropic resampling is to optimize the distance field of a point cloud with geometric consistency and certain point number. Combining the efficient intrinsic control, the geometrically-optimized resampling is convenient to implement, quick converges and avoids trapping in local optimum. For geometric feature keeping, the adaptively-isotropic resampling is proposed as an additional selection. It achieves a better balance between isotropic property and geometric feature keeping. In following parts, we will discuss the implementation details.

4.1 Isotropic Resampling

According to the aforementioned core issue of isotropic resampling, a mathematical model can be used to represent the isotropic resampling energy as

$$E_I = \sum_{i=1}^n ||d(p_i, p_j) - l||, p_j \in r(p_i), M \in r(P), \quad (6)$$

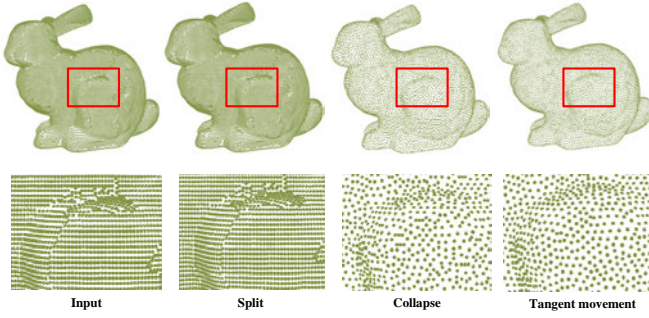


Fig. 8. An instance of isotropic resampling with intermediate outputs.

where E_I represents the isotropic resampling energy, r is the 1-ring local region which is defined by Voronoi Diagram on the MLS surface M and constructed from the neighbor structure. The neighbor structure union set $r(P)$ of point cloud P should cover the M . The average edge length l is computed from the mean value of the distances from p_i to its neighbor set $\{p_j\}$. The n is the certain point number in resampling result. For each point, the neighbor structure has been improved by the efficient intrinsic control.

An isotropic remeshing [57] is proposed to optimize the Equation 6 based on meshes. It can be regarded as a pure geometric update process. However, such parts are designed for meshes but not point clouds which means the isotropic remeshing can not be used for isotropic resampling directly. To solve the problem, we redesign basic steps of isotropic remeshing to fit point clouds. Based on the efficient intrinsic control, a more accurate local region detection can be achieved for each point. Combining all local regions from a point cloud, a half-edge structure is achieved. Based on the structure, the aforementioned four components can be used with a simple modification. Details of our isotropic resampling include the following steps:

- 1) Constructing two kd-trees for the point cloud in Euclidean and the geodesic coordinate system, respectively.
- 2) Searching k neighbors for each point from two kd-trees ($k = 8$ by default). The neighbor should satisfy the k -nearest searching for two kd-trees at the same time. Mapping the k neighbors in the tangent space and constructing Delaunay triangulation for each point. A half-edge structure is achieved from the Delaunay triangulation with the intrinsic control.
- 3) Detecting the average edge length l . If l is not empty, then skip this step; otherwise, compute the average edge length from the half-edge structure to l .
- 4) Running the 3 components (split, collapse, and tangent movement) for the half-edge structure iteratively (assume e_i represents an edge of two adjacent points, running split when $|e_i| > 1.6l$ and collapse when $|e_i| < 0.8l$).
- 5) If the point number is larger or smaller than certain point number from application requirement or user-specified, then increase or decrease the l and loop to step 3.

The main process of resampling is concentrated in Step 4. In Figure 9, we show an instance of the step with inter-

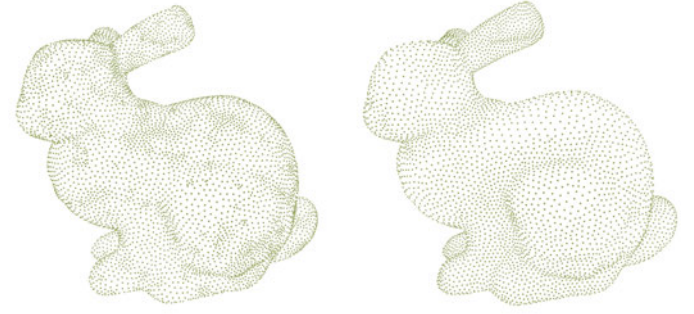


Fig. 9. Comparing the resampling results from the adaptively-isotropic remeshing and our adaptively-isotropic resampling. It is clear that the resampling result of adaptively-isotropic remeshing is discontinuous with abnormal distribution.

mediate outputs. To achieve the isotropic resampling with a certain number, we change the l in each step. The resampling point number is controlled into the certain number. Once the error between two numbers is close enough, we stop to change the l and adjust the number of split and collapse (one split add a point and one collapse minus a point) in Step 4 to achieve final resampling result. Then the point number in resampling result can be controlled.

4.2 Adaptively-Isotropic Resampling

An isotropic resampling is a strictly isotropic point cloud with global uniform density, the distances between each point and its neighbors are almost equal. However, it causes the loss of geometric features to a certain degree. The curvature sensitive is lost. For a fully functional resampling, it should provide an adaptively-isotropic resampling to balance the isotropic property and geometric feature keeping. The mathematical model of adaptively-isotropic resampling is represented as

$$E_H = \sum_{i=1}^n ||d_\eta(p_i, p_j) - l_\eta(p_i)||, \quad (7)$$

where d_η is the adaptive distance, $d_\eta(p_i, p_j) = (\eta(p_{ij}) \cdot d(p_i, p_j))$, η is used to decided the scheme for adaptive distance, which can be regarded as the density function, $\eta(p_{ij}) = (\eta(p_i) + \eta(p_j))/2$, l_η is the adaptive length defined by η . Compared to the Equation 6, the adaptively-isotropic resampling changes the distance d to adaptive distance d_η , which means that distances in different regions are not same. Comparing to average edge length l , values of l_η in different regions are not same. It can be used to adjust the density in different regions. Then some important geometric features such as curvature sensitive and sharp edges can be kept. A resolution has been provided in the adaptively-isotropic remeshing [58]. The basic idea is to change the average length l in different areas to fit η . However, the isotropic property is broken in the area with sharp changes of η value. In Figure 9, we show an instance to introduce the drawback of adaptively-isotropic remeshing.

To achieve the balance, we provide a weighed function field for η . The weighed function field is used to smooth the sharp changes of d_η . The values of d_η are updated to fit con-

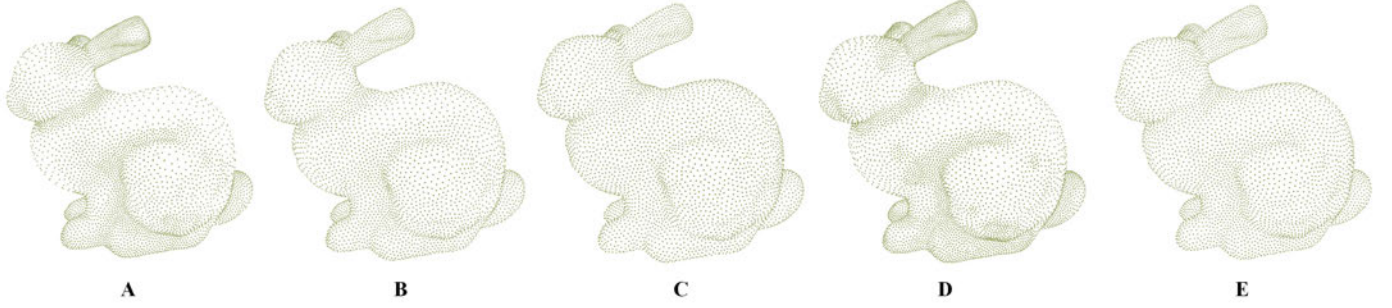


Fig. 10. Some instances of adaptively-isotropic resampling for curvature sensitive keeping. A: $(t = 5, l_q \in \{2.0l, 1.5l, l, 0.8l, 0.6l\})$; B: $(t = 5, l_q \in \{1.5l, 1.3l, l, 0.9l, 0.8l\})$; C: $(t = 5, l_q \in \{1.2l, 1.1l, l, 0.95l, 0.9l\})$; D: $(t = 3, l_q \in \{1.5l, l, 0.6l\})$; E: $(t = 3, l_q \in \{1.2l, l, 0.8l\})$.

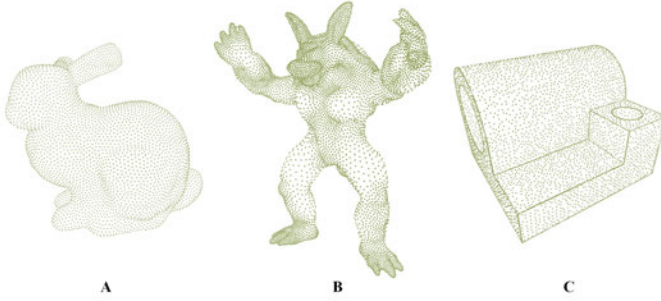


Fig. 11. Some instances of isotropic and adaptively-isotropic resampling results. A: isotropic resampling; B: adaptively-isotropic resampling for curvature sensitive keeping; C: adaptively-isotropic resampling for sharp edge keeping.

tinuous transformation in adaptively-isotropic resampling. The weighted function is represented as

$$\eta'(p_i) = \alpha \sum_{j=1}^k \frac{d(p_i, p_j) \eta(p_j)}{\sum_{j=1}^k d(p_i, p_j)} + \beta \eta(p_i), \quad (8)$$

where $\eta'(p_i)$ is the new value of $\eta(p_i)$ which has been smoothed by $\eta(p_j)$ from neighbors, α and β are update weights, $\alpha = 1 - \beta$ ($\beta = 0.5$ by default). The equation is similar to Equation 1. For curvature sensitive keeping, η represents the curvature of a point cloud. We use the average normal angle between a point and its neighbors to represent the curvature sensitive value. The curvature sensitive function is shown as

$$\eta_c(p_i) = \frac{1}{k} \sum_{j=1}^k \cos^{-1} \langle N(p_i) \cdot N(p_j) \rangle, \quad (9)$$

where η_c is the curvature, N is the unit normal vector. Taking η_c into the Equation 8, we achieve the weighted function for curvature sensitive keeping.

In order to improve the stability of the adaptively-isotropic resampling, the number of adaptive length values $\{l_\eta\}$ is controlled in a small range. Each value corresponds to a range of η_c . The correspondence is shown as

$$l_\eta(p_i) = \begin{cases} l_q, & \eta_c(p_i) \in [\eta_q, \eta_{q+1}], \\ l_q & l_q \in [l_1, \dots, l_t], q \in [1, \dots, t], \end{cases} \quad (10)$$

where $l_\eta(p_i)$ is the adaptive distance which is computed by range judgment of $\eta_c(p_i)$. Values of η_c are classified into

t groups. If $\eta_c(p_i)$ belongs to a range of group q ($\eta_c(p_i) \in [\eta_i, \eta_{i+1})$), then the corresponding value l_q of group q is assigned to $l_\eta(p_i)$. Using the same method of isotropic resampling and adaptive length values $\{l_q\}$, we achieve the adaptively-isotropic resampling. For $\{l_q\}$ selection, we choose different multiples of original average length l , which are convenient to change. In Figure 10, we show some instances of adaptively-isotropic resampling with different t and l_q .

For sharp edge keeping, edge points are detected by the edge detection method [59]. Similar to curvature sensitive keeping, density of edge points can be increased by smaller length l_e . During the isotropic resampling, we do not change edge point's positions which maintains sharp edges in the resampling result. Some details of the implementation have been discussed in our previous work [60]. Following different kinds of η , we can achieve various adaptively isotropic resampling results to fit certain requirement. In Figure 11, some resampling instances are shown.

5 APPLICATIONS

Based on the I&I resampling framework, we implement several applications including point cloud simplification, mesh reconstruction, and shape registration. For point cloud simplification, the I&I resampling can be used without any adjustments. The simplification result of I&I resampling keeps the geometric consistency and global uniform density with less point number. For mesh reconstruction, the I&I resampling improves the accuracy for local region detection. Using the classical Delaunay triangulation method [61] on resampling result, the accurate reconstructed mesh can be achieved directly.

For shape registration, our resampling method improves the efficiency of transform matrix computation. In [9], a point cloud normalization method can be used to be an effective initialization for registration. However, it requires the global uniform density and the same point numbers for both input point clouds to be registered. Our resampling makes any input point clouds to meet the two requirements. Combining with the normalization [9], it can be applied for registration and the influence of different scales and locations can be removed. The registration energy is represented as

$$E_d = H(K_s(P_i), O.K_s(P_j)), \quad (11)$$

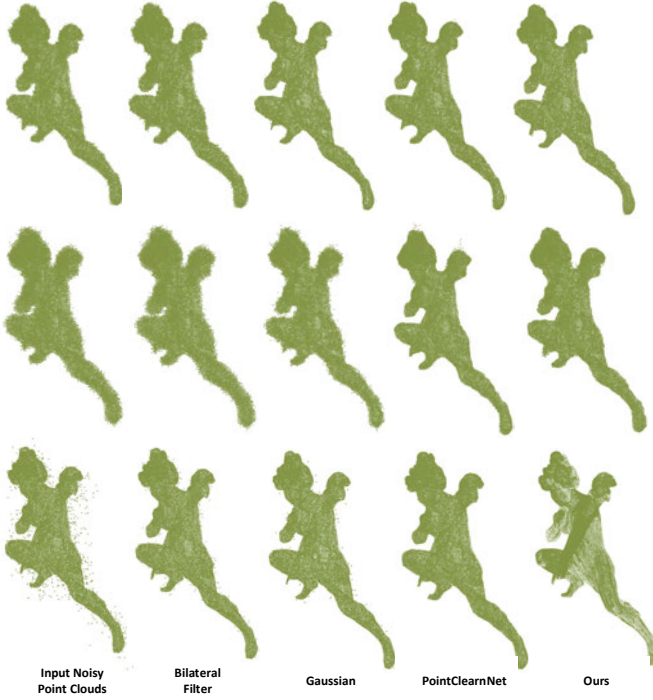


Fig. 12. Comparisons of denoising results. From top to bottom: 0.33 Gaussian noisy models and related denoising results; 0.66 Gaussian noisy models and related denoising results; random outliers and related denoising results.

where E_d represents the registration energy, computed by Hausdorff distance H , $K_s(P_i)$ is the normalization of point cloud P_i , O is the rotation matrix. The target of registration can be regarded as the optimization of E_d by searching O . To achieve the solution of O , we build a candidate set $\{O\}$ and this is achieved from the rotation with certain angle θ . Selecting the O_i with minimum value of E_d to be the initial O_{init} . The optimization based on $\{O\}$ is represented as

$$O_{init} = \{O_i | \min E_d(O_i)\}, O_i \in \{O\}, \quad (12)$$

$$\{O\} = \{(\theta.x, \theta.y, \theta.z) | x, y, z \in [0, 2\pi/\theta]\},$$

where $\theta.x$ means to rotate around the X axis with $\theta.x$ degrees, x is an integer, $x \in [0, 2\pi/\theta]$ ($\theta = \pi/12$ be default). $\theta.y$ and $\theta.z$ share the similar definition to $\theta.x$. Based on the O_{init} , we use the ICP [62] to achieve more accurate O . Finally, the registration result is achieved. The most significant advantage of our shape registration is that the method does not require any feature analysis such as normal matching, 3D keypoint extraction, different shape operators, etc. In experiments, we evaluate the performance of our resampling method for the applications.

6 EXPERIMENTS

We evaluate the performance of our resampling method in different applications which include point cloud simplification, mesh reconstruction, and shape registration. The experimental point cloud models are collected from Stanford [63], SHREC [64], and RGB-D scene [65] models. We implement the framework in Visual Studio 2019, X64 platform. The hardware configuration of the experimental machine is equipped with Intel Xeon W-2133 3.6GHz, 32G

TABLE 1
Geometric average error-based evaluation for different denoising methods.

Noisy Point Clouds/Points	Bilateral Filter	Gaussian Filter	Point-CleanNet	Ours
Angel(0.33)/237,018	0.001121	0.002105	0.001766	0.001451
Angel(0.66)/237,018	0.001895	0.003851	0.009576	0.00168
Angel(Outlier)/237,018	0.007151	0.004891	0.002337	0.000562
Armadillo(0.33)/172,974	0.003801	0.002524	0.003126	0.002257
Armadillo(0.66)/172,974	0.004604	0.005169	0.003948	0.002798
Armadillo (Outlier)/172,974	0.01557	0.01391	0.005732	0.001436
Bunny(0.33)/35,947	0.010906	0.008218	0.008257	0.004693
Bunny(0.66)/35,947	0.01526	0.024622	0.012222	0.006793
Bunny(Outlier)/35,947	0.06163	0.06241	0.07866	0.00284
Cat(0.33)/15,017	0.002899	0.004032	0.003409	0.004805
Cat(0.66)/15,017	0.003238	0.007807	0.006693	0.005637
Cat(Outlier)/15,017	0.01183	0.01246	0.02821	0.00225
Hand(0.33)/20,470	0.006672	0.0108	0.004436	0.004558
Hand(0.66)/20,470	0.007432	0.011328	0.006722	0.00821
Hand(Outlier)/20,470	0.01853	0.01855	0.009078	0.003062

TABLE 2
Time report for different denoising methods (average time in seconds for different kinds of noisy point clouds).

Noisy Point Clouds/Points	Bilateral Filter	Gaussian Filter	Point-CleanNet	Ours
Angel/237,018	2832.61s	117.06s	1213.83s	326.97s
Armadillo/172,974	304.47s	37.69s	400.14s	237.17s
Bunny/35,947	46.79s	7.71s	51.27s	48.62s
Cat/15,017	18.65s	3.36s	30.51s	20.55s
Hand/20,470	20.89s	4.56s	37.02s	28.22s

RAM, and Quadro P620. The rest of the experimental part is organized as follows: 1. results of denoising; 2. evaluation of simplification; 3. evaluation of mesh reconstruction; 4. evaluation of shape registration; 5. efficiency analysis of our resampling result which includes time complexity and algorithm convergence.

6.1 Results of denoising

In Subsec. 3.1, we have introduced our denoising method in point cloud pre-processing. To evaluate the performance of our denoising method, we provide a denoising measurement between our denoising method and peer ones, including Bilateral filter [49], Gaussian filter [50], and PointCleanNet [66]. We generate a noisy point cloud set from Stanford and SHREC models. The noisy point cloud set includes three kinds of noisy models: 0.33 Gaussian noise, 0.66 Gaussian noise, and random outliers. Let l_i to represent the average distance between a point p_i to its k ($k = 6$) neighbors. We

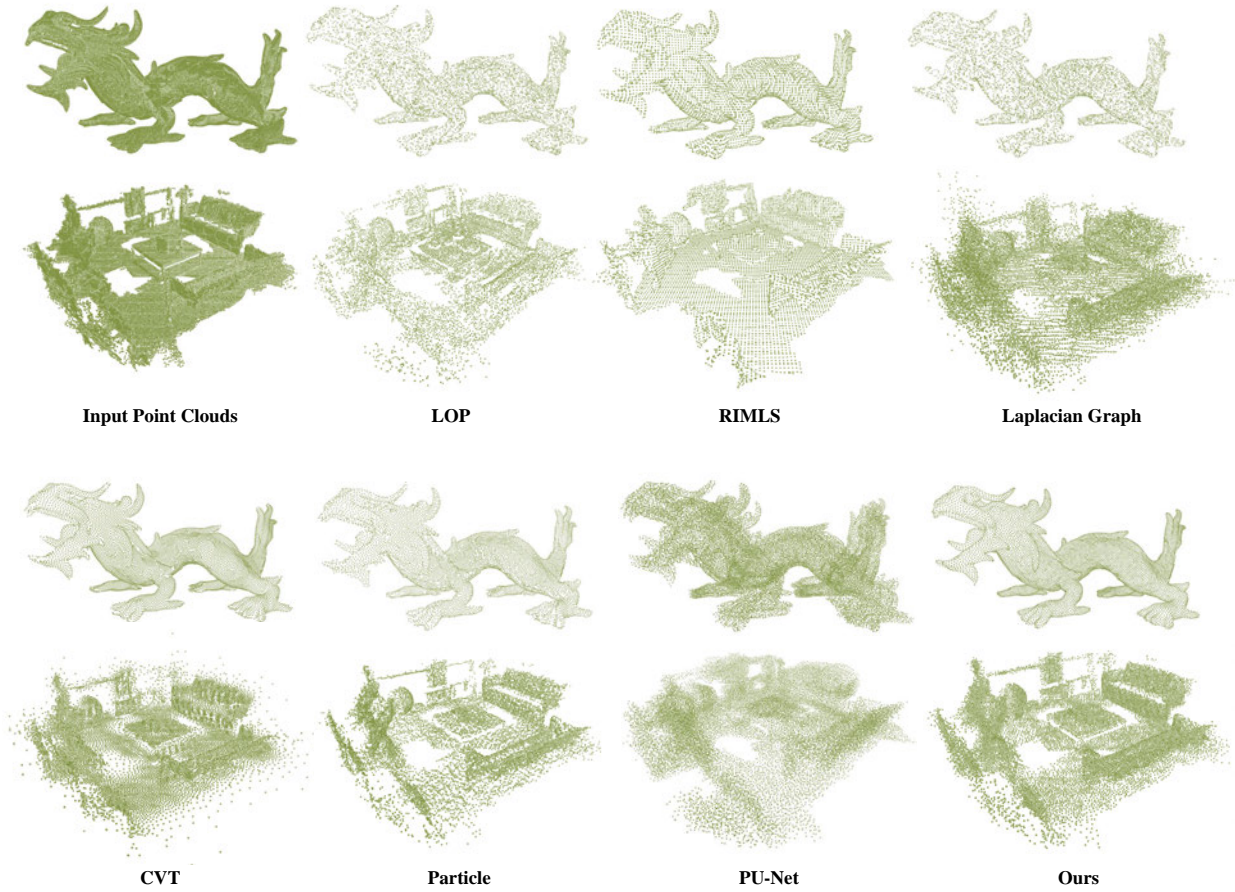


Fig. 13. Comparisons of simplification results by different methods. Model: Asian Dragon and 03scene; simplification number: 20,000.

TABLE 3
Geometric error analysis of different simplification methods.

Method	LOP		RIMLS		Laplacian		CVT		Particle-based		PU-Net		Ours	
Model/Points	Max	Avg	Max	Avg	Max	Avg	Max	Avg	Max	Avg	Max	Avg	Max	Avg
Bunny/1,000	0.0881	0.0137	0.1304	0.0233	0.0744	0.0156	0.1206	0.0194	0.0935	0.0155	0.0253	0.0052	0.0735	0.0158
Armadillo/1,000	0.0789	0.0135	0.2245	0.0181	0.0634	0.0134	0.2067	0.0155	0.0955	0.0138	Nan	Nan	0.2504	0.0214
Angel/5,000	0.0391	0.0038	0.1103	0.0215	0.0412	0.0039	0.0234	0.0042	0.0503	0.0039	Nan	Nan	0.0226	0.0039
Dragon/5,000	0.0541	0.0079	0.3008	0.0453	0.0331	0.0057	0.0394	0.0064	0.0458	0.0057	0.0268	0.0044	0.0369	0.0056
Buddha/20,000	0.0194	0.0027	0.0259	0.0028	0.0546	0.0089	0.0204	0.0032	0.0518	0.0062	0.0239	0.0045	0.0189	0.0026
Thai Statue/20,000	0.0249	0.0034	0.2054	0.0046	0.0602	0.0035	Nan	Nan	0.0633	0.0092	Nan	Nan	0.0186	0.0031
03scene/20,000	0.0734	0.0057	0.2171	0.0217	Nan	0.0232	0.0435	0.0019	0.0587	0.0079	0.1696	0.0095	0.0552	0.0015
04scene/20,000	0.0913	0.0053	0.1271	0.0121	0.0547	0.0102	0.0349	0.0019	0.0471	0.0053	0.1702	0.0107	0.0406	0.0017
03scene/50,000	0.0671	0.0029	0.1241	0.0143	0.0611	0.0012	0.0363	0.0012	0.0648	0.0011	0.1449	0.0083	0.0662	0.0009
04scene/50,000	0.0844	0.0041	0.1122	0.0112	Nan	0.0029	0.0327	0.0012	Nan	0.0032	0.1639	0.0097	0.0264	0.0011

add a random movement for each point according to normal vector to generate Gaussian noise. With different ranges ($[0, 0.33l_i]$ and $[0, 0.66l_i]$) of the movement, the related noisy models can be achieved. To generate random outliers, we randomly select some points and move them according to normal vectors with a range ($[3l_i, 6l_i]$). Based on the three kinds of noisy models, we compare the performance of different denoising methods by geometric average errors [31] [47]. The geometric average errors represent the geometric consistency between denoising results and ground truth, which are implemented by [47]. In Figure 12, we show some instances of denoising results by different methods.

In Table 1 and 2, we compare the geometric average errors of different methods with time report. It proves that our method achieves better performance of denoising, especially for noisy point clouds with random outliers. Our method is faster than PointCleanNet (the number of iteration steps in PointCleanNet is set to three by default).

6.2 Evaluation of Simplification

The target of the simplification is to simplify a point cloud while keeping the geometric consistency. In a quantitative view, it should minimize the MLS surface fitting error from the input point cloud to the simplification. The geometric error analysis also can be used to measure the

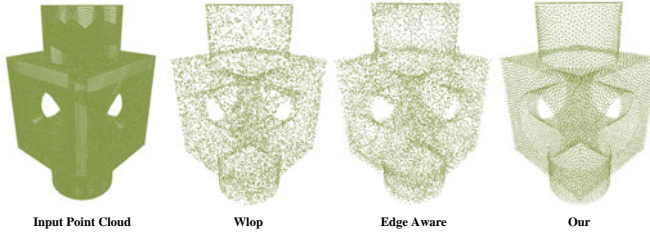


Fig. 14. Comparisons of simplification results by different methods. Model: Block; simplification number: 10,000.

fitting error, including geometric maximum and average errors. The geometric maximum error represents the upper MLS fitting error. To show the performance of our resampling method, we compare several mainstream simplification methods, including Locally Optimal Projection-based simplification (LOP) [67], Robust Implicit MLS-based simplification (RIMLS) [68], Laplacian graph-based uniform resampling [39], Centroidal Voronoi Tessellation (CVT) [19], Particle-based resampling [20], and PU-Net [69]. The LOP-based simplification and Centroidal Voronoi Tessellation are implemented by the CGAL library, which is a computer graphic toolbox. The Robust Implicit MLS-based simplification is implemented in MeshLab. The Laplacian graph-based uniform resampling is realized with the same platform of our resampling method. The optimization of particle-based resampling is processed by a hybrid L-BFGS tool: [HLBFGS](#). The code of PU-Net is provided by the author.

Test point clouds are collected from Stanford and RGB-D scene models. Each point cloud from the two datasets is a dense one which is convenient to be simplified by different simplification methods. The number of points of these point clouds ranges from about 30,000 to 3,000,000. We specify the simplification point number to 10,000 or 50,000 for different point clouds. In Figures 13, we show the visualization of the simplification results by different methods. It is clear that our method achieves similarly isotropic resampling results than the state-of-the-art. In Table 3, we compare the geometric error analysis results of different methods. The results show that our resampling method achieves a better balance between isotropic resampling and geometric consistency keeping. Some methods based on tangent space optimization may fail to achieve accurate simplification result in the region with sharp curvature changes (labeled by "Nan" in Tabel 3, means "not a reasonable number"). Benefited from the intrinsic control, such failure can be avoided in our resampling method.

As mentioned in Subsec. 4.2, our method can keep sharp edges in simplification result by adaptively-isotropic resampling and edge detection method [59]. To evaluate the performance of sharp edge keeping, we collect some point clouds with sharp edges to be the test set. We compare some simplification methods, including WLOP simplification [33] and edge-aware resampling [70]. In Figures 14, we show the visualization of the simplification results with sharp edges by different methods. In Table 4, we compare the geometric error analysis results of different methods. The results show that our resampling method achieves a better

TABLE 4
Geometric error analysis of different simplification methods.

Method		WLOP		EdgeAware		Ours	
Models	Points	Max	Avg	Max	Avg	Max	Avg
Fandisk	10,000	0.0311	0.0018	0.0507	0.0059	0.0159	0.0014
Block	10,000	0.0273	0.0025	0.0345	0.0035	0.0177	0.0017
Joint	10,000	0.0448	0.0027	0.0398	0.0083	0.0307	0.0022
Fandisk	20,000	0.0194	0.0009	0.0215	0.0018	0.0123	0.0008
Block	20,000	0.0192	0.0013	0.0229	0.0025	0.0122	0.0010
Joint	20,000	0.0289	0.0015	0.0338	0.0026	0.0207	0.0010

balance between sharp edge keeping and geometric consistency keeping.

6.3 Evaluation of Mesh Reconstruction

To evaluate the performance of our resampling method for mesh reconstruction, we introduce two metrics for reconstruction quality measurement: triangle quality measurement [71] [72] and Hausdorff distance [73]. The triangle quality measurement is used to measure the isotropic property of the reconstructed mesh, which is computed by $Q_t = \frac{6}{\sqrt{3}} \cdot \frac{S_t}{p_t h_t}$, t represents a triangle in the reconstructed mesh. S_t is the area of t . p_t is the in-radius of t . h_t is the longest edge length of t . The measurement includes the minimum and average values of Q_t from the reconstructed mesh. The Hausdorff distance can be used to measure the geometric consistency between reconstructed mesh and original point cloud.

To show the advantages of our method for reconstruction, we compare several popular mesh reconstruction methods, including Scale Space [74], Screened Poisson [27], Advancing Delaunay Reconstruction [61] [75], CVT [19], and Particle-based reconstruction [20]. The CVT and Particle-based reconstruction methods have been discussed before. The two methods resample an input point cloud into an isotropic one. Based on an isotropic point cloud, the reconstructed mesh can be achieved conveniently. The Scale Space-based reconstruction is to detect the local region from a ball space. The Screened Poisson reconstruction is a global optimization strategy that is used to define different sides of the 2-manifold. The Advancing Delaunay triangulation is a classical method for mesh reconstruction. It detects the local regions from a point cloud by Voronoi Diagram. The Scale Space and advancing Delaunay triangulation are implemented by the CGAL library. The Screened Poisson reconstruction is processed by the related project: [Poisson Version8.0](#). Some methods (Scale Space and Advancing Delaunay) cannot control the output point number. For an unbiased comparison, we resample the point cloud by Poisson disk resampling [76] to uniform the point number before reconstruction of the two methods.

Test point clouds are selected from SHREC and RGB-D scene models. Most models from SHREC are non-uniform and sparse point clouds, which increases the difficulty for mesh reconstruction. Each point cloud in RGB-D scene dataset belongs to multi-objects model. We set the target point number to 10,000 or 20,000 in the reconstructed mesh. In Figure 15, we compare the reconstructed meshes by different methods. The results show that our method achieves better isotropic property in the reconstructed meshes. In

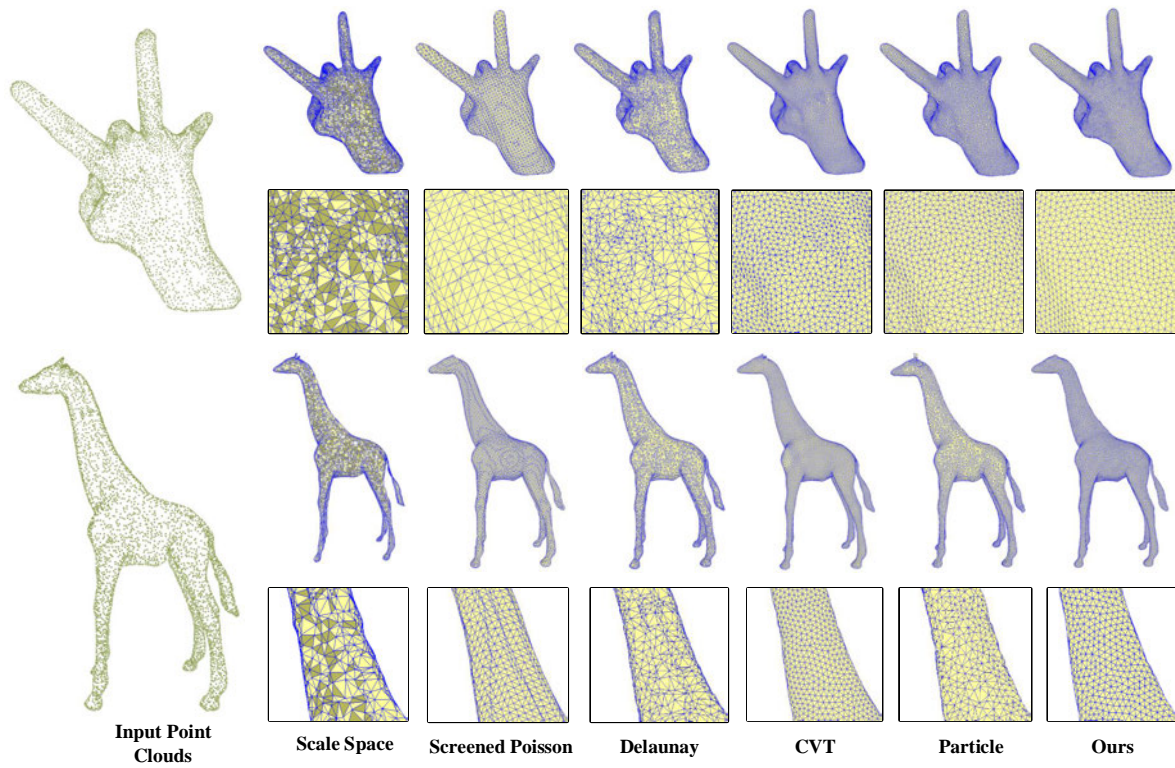


Fig. 15. Comparisons of mesh reconstruction results. Model: Hand and Giraffe; point number: 10,000.

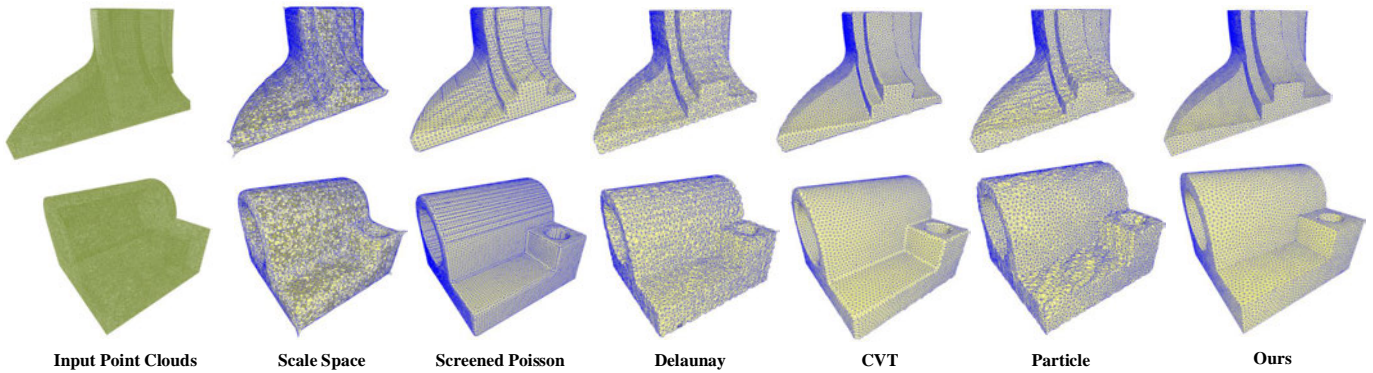


Fig. 16. Comparisons of mesh reconstruction results with sharp edges by different methods. Model: Fandisk and Joint; point number: 10,000.

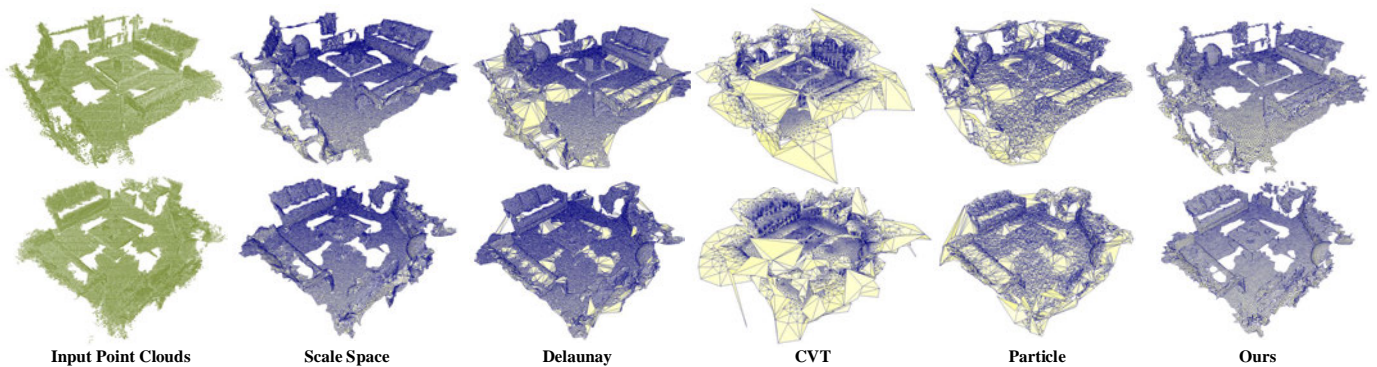


Fig. 17. Comparisons of mesh reconstruction results with multi-objects by different methods. Model: 02scene and 03scene; point number: 20,000.

TABLE 5

Triangle quality measurement by different reconstruction methods in SHREC (10,000 points) and RGB-D scene (20,000 points) models.

Method	ScaleSpace		ScreenPoisson		Advancing Delaunay		CVT		Particle-based		Ours	
Model	Q_{min}	Q_{avg}	Q_{min}	Q_{avg}	Q_{min}	Q_{avg}	Q_{min}	Q_{avg}	Q_{min}	Q_{avg}	Q_{min}	Q_{avg}
Ant	Nan	0.65	Nan	0.61	0.13	0.71	0.45	0.88	Nan	0.74	0.56	0.91
Centuar	Nan	0.65	Nan	0.61	0.11	0.72	0.42	0.87	Nan	0.71	0.59	0.91
Giraffe	Nan	0.64	Nan	0.62	0.13	0.70	0.33	0.88	Nan	0.71	0.64	0.91
Hand	Nan	0.68	Nan	0.61	Nan	0.72	0.38	0.88	Nan	0.78	0.57	0.91
WoodMan	Nan	0.62	Nan	0.61	Nan	0.68	0.22	0.88	Nan	0.72	0.54	0.91
02scene	Nan	0.74	—	—	Nan	0.77	Nan	0.76	Nan	0.60	Nan	0.81
03scene	Nan	0.75	—	—	Nan	0.77	Nan	0.76	Nan	0.60	0.16	0.83
04scene	Nan	0.74	—	—	Nan	0.77	Nan	0.77	Nan	0.60	0.12	0.83

TABLE 6

Hausdorff distances by different reconstruction methods in SHREC (10,000 points) and RGB-D scene (20,000 points) models.

Method Model	Scale	Screen	Advancing Delaunay	CVT	Particle	Ours
Ant	0.0290	0.1053	0.0176	0.0183	0.0561	0.0172
Centuar	0.0352	0.3983	0.0281	0.0208	0.0762	0.0251
Giraffe	0.02031	0.2498	0.0110	0.0157	0.0202	0.0809
Hand	0.0347	0.2938	0.0216	0.0215	0.0265	0.0779
WoodMan	0.0244	0.1479	0.0110	0.0160	0.0253	0.0171
02scene	0.3211	—	0.3214	0.3739	0.0957	0.0610
03scene	0.1398	—	0.1403	0.2346	0.0905	0.0808
04scene	0.0275	—	0.0173	0.2015	0.0543	0.0310

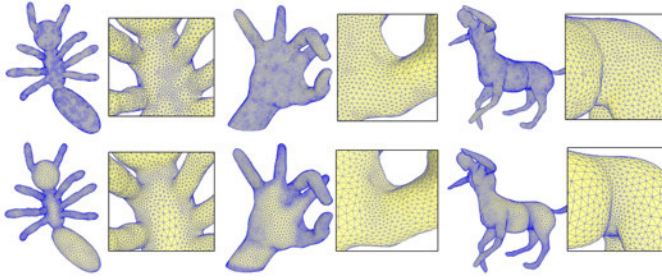


Fig. 18. Comparisons of curvature sensitive mesh reconstruction results. First row: results by adaptive remeshing [58]; second row: results by our resampling method ($t = 5, l_q \in \{2.0l, 1.5l, l, 0.8l, 0.6l\}$). Model: Ant, Hand, and Centuar.

Tables 5 and 6, we show the triangle quality measurement and Hausdorff distance results of different methods. Based on the experimental results, our methods achieves similar Hausdorff distances and better isotropic property with other methods. The results prove that our method obtains better balance between geometric consistency and mesh quality. In Figure 16 and 17, we compare different reconstructed meshes with sharp edges and multi-objects. Our reconstructed mesh keeps more accurate details of geometric features and better isotropic property. More details of sharp edge keeping in mesh reconstruction are discussed in another of our work [60].

For curvature sensitive keeping, our adaptively-isotropic resampling achieves a balance between isotropic property and adaptive density based on curvature. The adaptive remeshing [58] is used to compare with our method. In Figure 18, we show the reconstructed meshes with curvature sensitive keeping. The results show that our method keeps better balance. For further quantitative analysis, we calculate mean values of triangle quality measurement and minimum inner angle statistics from test models. In Fig-

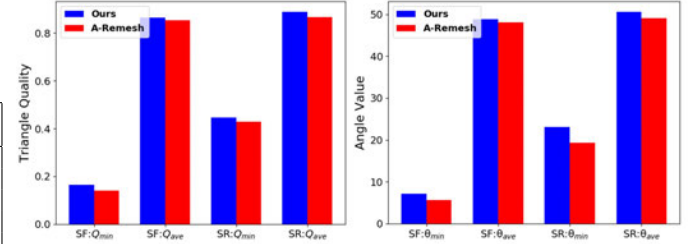


Fig. 19. Comparisons of triangle quality measurement and minimum inner angle statistics results by adaptive remeshing (A-Remesh) [58] and our method. SF: Stanford Models; SR: SHREC Models.

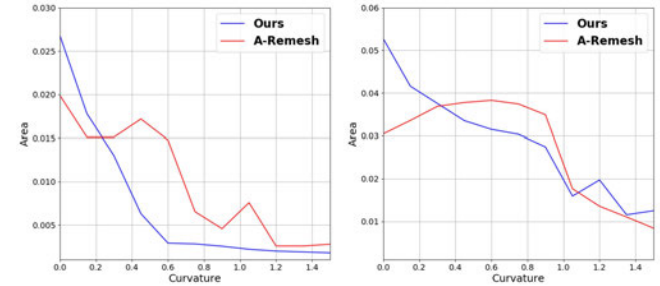


Fig. 20. Comparisons of related area change curves from adaptive remeshing (A-Remesh) [58] and our method (left: curves in Stanford models; right: curves in SHREC models).

ure 19, we compare the mean values between adaptive remeshing [58] and our method. The results show that our method achieves better isotropic property in reconstructed meshes with curvature sensitive keeping. Benefited from the weighted function field, our reconstructed meshes have smooth density changes based on curvature. The smooth density changes means that the related areas of different points should be inversely proportional to the curvature values (the related area is the area of Voronoi cell for a point). To evaluate the smoothness, we calculate the related area change curve based on the relationship between curvature value and related area for each point. In Figure 20, the curves in two test sets are shown. Our curves are smoother while keeping better monotonicity. It proves that our method achieves better curvature sensitive property with more stable performance. In Figure 21, we shown some resampling and mesh reconstruction results for point clouds with more complex geometric structures. Some more results are provided for download: [test models](#).



Fig. 21. Some resampling and mesh reconstruction instances by our method. The third column: the mesh reconstruction is based on isotropic resampling; the fifth column: the mesh reconstruction is based on adaptively-isotropic resampling.

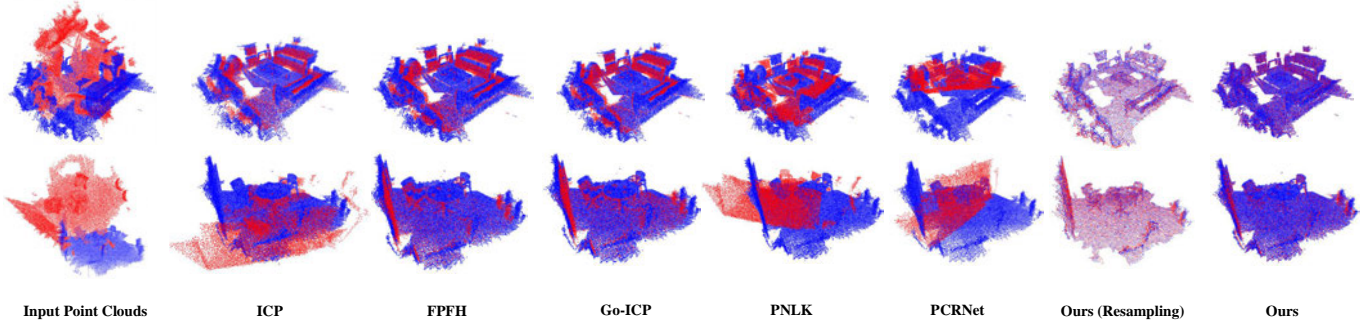


Fig. 22. Comparisons of shape registration results for multi-object point clouds from RGB-D scene models. Model: 01scene and 05scene.

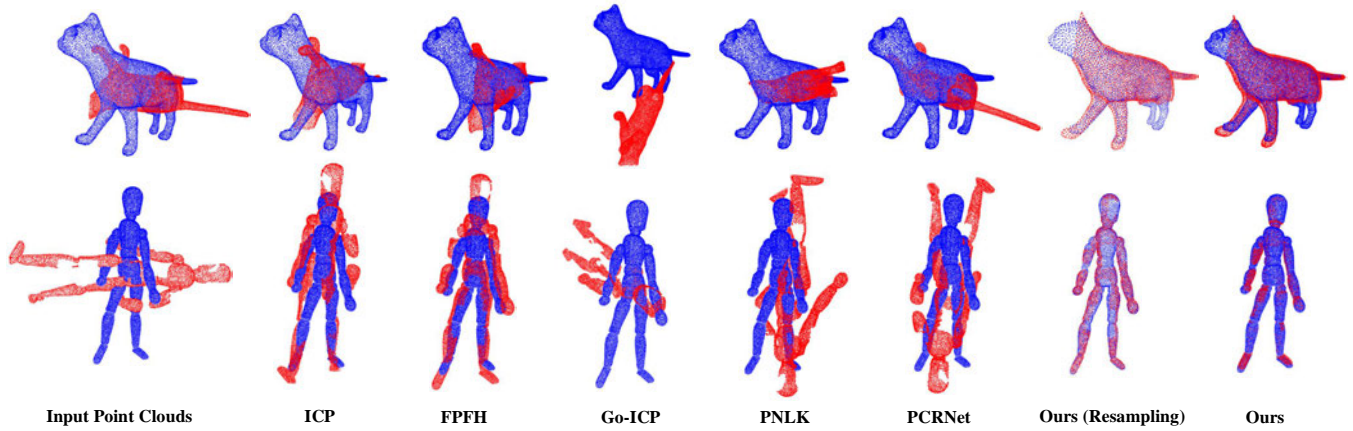


Fig. 23. Comparisons of shape registration results for defective point clouds from SHREC models. Model: Cat and Woodman.

TABLE 7

Mean squared errors (MSE) of point distance by different registration methods in RGB-D scene models.

Method Model	ICP	FPFH	Go-ICP	PNLK	PCR-Net	Ours
01scene	4.75E-3	4.76E-3	4.86E-3	1.84E-2	2.16E-1	2.63E-4
02scene	3.21E-2	2.29E-2	3.52E-6	7.95E-1	1.99E-1	3.44E-4
03scene	Nan	1.86E-2	3.11E-2	7.93E-1	4.69E-1	3.02E-4
04scene	1.49E-2	1.14E-2	1.70E-2	3.29E-1	7.23E-2	3.61E-4
05scene	3.32E-6	1.01E-6	2.62E-6	1.57E-5	1.16E-1	3.10E-4
06scene	8.08E-3	8.06E-3	8.78E-3	5.26E-2	2.76E-2	7.11E-1
07scene	8.54E-5	8.54E-5	8.62E-5	8.64E-5	1.03E-1	2.00E-4
08scene	5.97E-2	3.06E-3	3.15E-3	1.58E-1	1.17E-1	3.97E-4
09scene	1.14E-2	1.15E-2	1.18E-2	1.57E-1	1.64E-1	6.43E-4
10scene	7.91E-3	9.20E-2	7.95E-3	4.29E-2	5.93E-1	5.83E-4
11scene	6.53E-3	6.53E-3	6.75E-3	4.65E-2	2.38E-1	8.07E-4
Time	159.32s	225.36s	326.23s	69.32s	5.49s	56.32s

TABLE 8

Mean squared errors (MSE) of point distance by different registration methods in SHREC models.

Method Model	ICP	FPFH	Go-ICP	PNLK	PCR-Net	Ours
Ant	6.81E-3	1.07E-6	1.53E-2	1.15E-1	1.69E-2	4.43E-3
Cat	1.35E-3	1.37E-3	8.36E-2	9.65E-3	2.01E-2	2.23E-4
Centuar	1.33E-2	9.70E-7	8.87E-2	2.50E-2	5.06E-2	6.61E-4
Dog	1.21E-2	1.10E-3	8.70E-2	7.10E-2	5.81E-2	1.85E-4
Giraffe	7.41E-4	7.39E-4	2.45E-1	8.73E-3	1.30E-2	4.58E-5
Girl	2.19E-3	1.33E-3	2.28E-2	6.01E-2	4.92E-3	3.42E-4
Hand	3.34E-3	2.95E-3	1.37E-1	1.22E-2	6.83E-3	6.09E-5
Wood-Man	7.74E-3	3.55E-3	4.62E-1	3.29E-2	1.90E-2	6.35E-5
Time	15.65s	75.26s	52.36s	4.78s	3.26s	37.16s

6.4 Evaluation of Registration

As mentioned before, we provide a shape registration application based on our resampling method and this improves the efficiency of transform matrix computation in the registration. We compare different classical shape registration methods to show the improvement, including ICP [62], FPFH [34], Go-ICP [77], PointNetLK [28], and PCRNet [29]. The ICP and FPFH are implemented by the PCL library. The codes of Go-ICP, PointNetLK and PCRNet are provided by related projects on Github.

Test point clouds are collected from RGB-D scene and SHREC models. The source point clouds and target point

clouds are resampled with non-uniform density distribution, different scales, random locations, and random rotations. The point clouds collected from RGB-D scene models are used to evaluate the performance of multi-object models' registration. In Figure 22, we show some shape registration results by different methods. It is clear that our method achieves better results for the multi-objects models with different scales, locations, and rotations. To quantify the quality of registration, we compute mean squared errors of point distance between the registered point clouds (red) and target point clouds (blue). In Tables 7, the distances are shown from different registration methods. Our registration method achieves better results.

We also build a subset with defective point clouds from

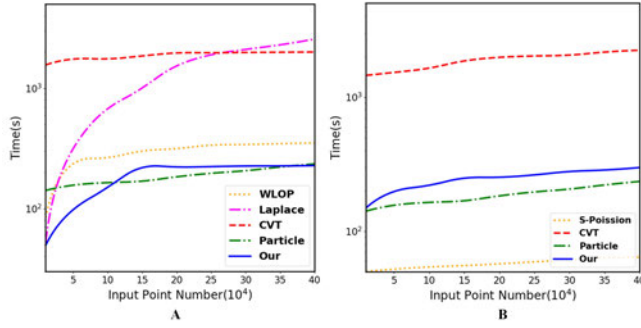


Fig. 24. Comparisons of time cost reports of different methods. A: simplification methods; B: mesh reconstruction methods(S-Poisson: Screened Poisson).

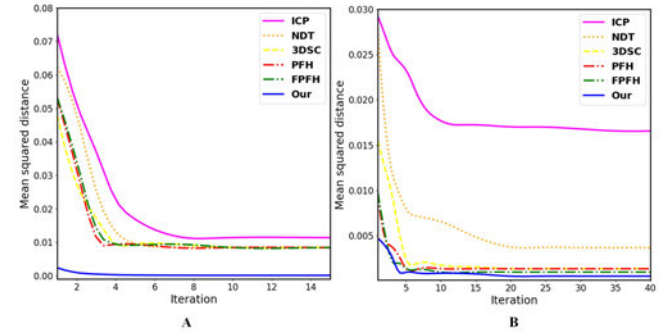


Fig. 26. The convergence reports (A: point clouds with random scales, rotations and translations and B: defective point cloud subset) of different shape registration methods.

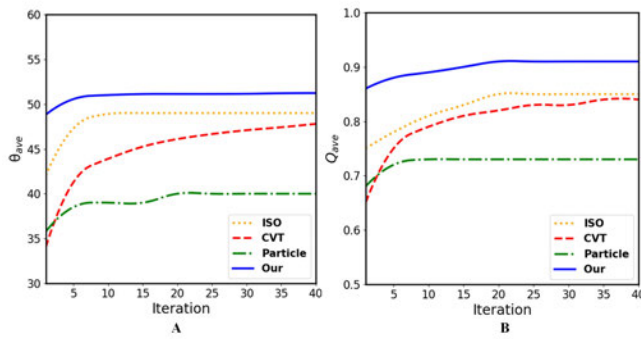


Fig. 25. The convergence reports (A: $\theta(t)$ and B: $Q(t)$) of different mesh reconstruction methods (ISO: isotropic remeshing [57]).

SHREC. It is used to evaluate the robustness of different registration methods. In Figure 23, we compare the registration results by different methods. In Table 8, mean squared errors of point distance are shown from different registration methods. The results prove that our method achieves better performance for registration of defective point clouds (red). In summary, the registration can be improved based on our resampling method without any complex framework.

6.5 Efficiency Analysis

We provide the efficiency analysis for our resampling method, which is constructed by time complexity and algorithm convergence analysis. For our resampling framework, the computation includes efficient intrinsic control and geometrically-optimized resampling. The time complexity of the efficient intrinsic control implementation is about $O(n \log n)$ that is decided by the construction of the geodesic coordinate system. For geometrically-optimized resampling, the time complexity is approximately equal to $O(sm k)$, s is the iterative steps of isotropic resampling ($s \in [5, 10]$), m is resampling point number by application requirement or user-specified, and k is the neighbor point number for neighbor structure detection.

For simplification, our method achieves better performance than other mainstream resampling methods. In Figure 24A, we show a graph of time cost reports by different methods for simplification (10,000 points in simplification

result). We also provide the time cost reports (Tables 9 and 10) in extreme cases (input point number: 5k~5,000k, output point number: 1k~1,000k). It shows that our resampling method has good performance when the output point cloud number is lower than 50k. However, the performance of our method is reduced when the specified number of output points is greater than 1,000k. The reason is that the time cost of Delaunay triangulation for all points is huge. For mesh reconstruction, some methods that can not control the point number in the reconstructed mesh are removed. Comparing different methods, the Screened Poisson achieves better performance than other methods. However, most geometric features are lost in reconstructed mesh and the isotropic property is not good. The point number can be controlled into a rough range but not an accurate number. In Figure 24B, we compare time cost reports from different reconstruct methods (with 10,000 points). Although our method is not faster than some methods, it achieves a better balance between effectiveness and isotropic property. The storage cost of our method depends on the volume of input point cloud (related to peak value of storage cost) and output point number specified by user or application (related to median value of storage cost). A rough estimation is provided: input point clouds 5k(8MB)~5,000k(400MB); output point clouds 1k(6MB)~1,000k(120MB).

For convergence analysis, we compare the θ_{ave} (average inner angle of reconstructed mesh), Q_{ave} , and mean squared distance in different iterations of mesh reconstruction and registration. θ_{ave} and Q_{ave} represent the quality of the isotropic property and mean squared distance evaluates the accuracy of registration. In Figures 25 and 26, convergence reports are shown from different reconstruction and registration methods. Benefited from the geometrically-optimized resampling method, the I&I resampling achieves faster convergence speed. It avoids complex optimization while keeping accuracy in the resampling process. For shape registration, our method achieves better performance for point clouds with defective parts, different scales and rotations. Above all, it proves that our resampling framework can be used to improve different point cloud-based applications without any other complicated algorithms.

Our method achieves better performance for resampling and improvement for related applications, but some limita-

TABLE 9

Time cost report of our resampling method (previous steps) in extreme cases.

Mapping into the geodesic coordinate system					
Output\Input	50K	100K	500K	1,000K	5,000K
1k	1.46s	1.48s	1.56s	1.67s	2.27s
5k	6.49s	7.81s	7.96s	8.91s	12.25s
10k	11.86s	15.56s	16.14s	19.45s	23.06s
50k	—	61.66s	103.28s	105.36s	123.06s
100k	—	—	201.75s	203.02s	220.36s
500k	—	—	—	736.23s	784.36s
1,000k	—	—	—	—	1365.84s
Delaunay triangulation with the intrinsic control					
Output\Input	50K	100K	500K	1,000K	5,000K
1k	0.84s	0.87s	1.99s	1.19s	1.21s
5k	4.63s	4.78s	6.51s	6.71s	7.88s
10k	8.04s	9.87s	16.81s	17.22s	16.31s
50k	—	69.17s	251.92s	268.24s	283.56s
100k	—	—	744.26s	916.16s	923.56s
500k	—	—	—	>1h	>1h
1,000k	—	—	—	—	>1h

TABLE 10

Time cost report of our resampling method in extreme cases.

Output\Input	50k	100k	500k	1,000k	5,000k
1k	15.44s	15.61s	16.51s	16.74s	18.95s
5k	45.41s	47.21s	51.82s	56.45s	57.54s
10k	81.53s	85.59s	92.32s	98.36s	106.28s
50k	—	405.24s	683.31s	742.29s	753.69s
100k	—	—	2004.27s	2157.35s	2192.02s
500k	—	—	—	>1h	>1h
1,000k	—	—	—	—	>1h

tions are still exist, which can be concluded into two parts: efficiency of computation and accuracy for city-scale point clouds. According to the experiments, the efficiency of our resampling is reduced when the output point number is larger than 50k. The time cost of the geodesic coordinate system mapping and Delaunay triangulation with intrinsic control is proportional to the point number. Based on the current version, our method focuses on single object or indoor scene resampling. For city-scale point clouds, the performance of our method may be reduced without structural optimization or segmentation. The reason is that such point clouds have multiple objects stick to each other. The neighbor structure cannot be accurately detected even in the geodesic coordinate system. An instance is shown in Figure 27. In future work, we will design a sub-division strategy to improve the efficiency and accuracy for large-scale scenarios.

7 CONCLUSION

We have proposed an I&I resampling framework which is constructed by efficient intrinsic control and geometrically-optimized resampling. The efficient intrinsic control improves the accuracy of local region detection for a point cloud while avoiding the redundant geodesic computation. The geometrically-optimized resampling method can be used to achieve an isotropic or adaptively-isotropic resampling result without complex optimization. The framework avoids drawbacks of tangent space-based resampling, including loss of geometric feature and inaccurate local region detection. Using our framework, the intrinsic and isotropic

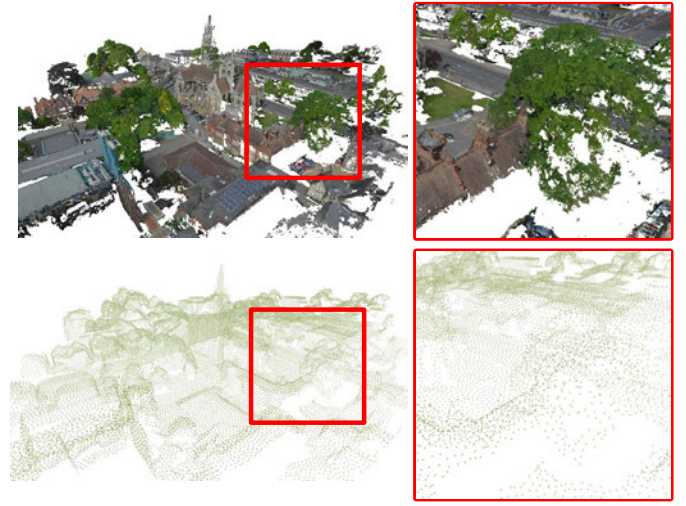


Fig. 27. An instance of city-scale point cloud resampling.

resampling can be processed at the same time while keeping all advantages of both. The I&I resampling provides a new technical route for point cloud processing. Experiments have proved that the I&I resampling framework can improve the accuracy, robustness, and convergence speed in different applications.

ACKNOWLEDGMENT

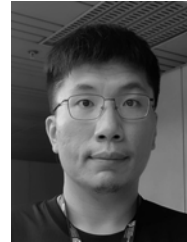
This research is supported by the Ministry of Education, Singapore, under its Tier-1 Fund MOE2021, RG14/21. The authors would like to thank the code providers for CGAL Library, Point Cloud Library, and Screened Poisson Project. We also thank the data providers for Stanford, SHREC, and RGB-D scene datasets.

REFERENCES

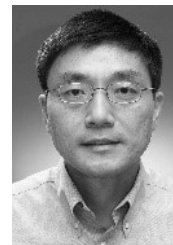
- [1] Y. Cui, S. Schuon, S. Thrun, D. Stricker, and C. Theobalt, "Algorithms for 3d shape scanning with a depth camera," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 5, pp. 1039–1050, 2012.
- [2] Q. Legros, J. Tachella, R. Tobin, A. McCarthy, S. Meignen, G. S. Buller, Y. Altmann, S. McLaughlin, and M. E. Davies, "Robust 3d reconstruction of dynamic scenes from single-photon lidar using beta-divergences," *IEEE Transactions on Image Processing*, 2020.
- [3] W. Wu, Z. Qi, and L. Fuxin, "Pointconv: Deep convolutional networks on 3d point clouds," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.
- [4] T. Schöps, T. Sattler, and M. Pollefeys, "Surfelmeshing: Online surfel-based mesh reconstruction," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019.
- [5] X. Guan and H. Wu, "Leveraging the power of multi-core platforms for large-scale geospatial data processing: Exemplified by generating dem from massive lidar point clouds," *Computers & Geosciences*, vol. 36, no. 10, pp. 1276–1282, 2010.
- [6] G. J. Brostow, J. Shotton, J. Fauqueur, and R. Cipolla, "Segmentation and recognition using structure from motion point clouds," in *European conference on computer vision*. Springer, 2008, pp. 44–57.
- [7] W. Cheng, W. Lin, X. Zhang, M. Goesele, and M.-T. Sun, "A data-driven point cloud simplification framework for city-scale image-based localization," *IEEE Transactions on Image Processing*, vol. 26, no. 1, pp. 262–275, 2016.

- [8] W. Cheng, K. Chen, W. Lin, M. Goesele, X. Zhang, and Y. Zhang, "A two-stage outlier filtering framework for city-scale localization using 3d sfm point clouds," *IEEE Transactions on Image Processing*, vol. 28, no. 10, pp. 4857–4869, 2019.
- [9] C. Lv, Z. Wu, X. Wang, and M. Zhou, "3d facial similarity measurement and its application in facial organization," *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, vol. 16, no. 3, pp. 1–20, 2020.
- [10] F. Tsalakanidou, F. Forster, S. Malassiotis, and M. G. Strintzis, "Real-time acquisition of depth and color images using structured light and its application to 3d face recognition," *Real-Time Imaging*, vol. 11, no. 5-6, pp. 358–369, 2005.
- [11] Z. Song and R. Chung, "Determining both surface position and orientation in structured-light-based sensing," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 10, pp. 1770–1780, 2009.
- [12] Y.-J. Liu, C.-X. Xu, D. Fan, and Y. He, "Efficient construction and simplification of delaunay meshes," *ACM Transactions on Graphics (TOG)*, vol. 34, no. 6, pp. 1–13, 2015.
- [13] G. Shamai and R. Kimmel, "Geodesic distance descriptors," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 6410–6418.
- [14] L. Maier-Hein, A. M. Franz, T. R. Dos Santos, M. Schmidt, M. Fangerau, H.-P. Meinzer, and J. M. Fitzpatrick, "Convergent iterative closest-point algorithm to accommodate anisotropic and inhomogeneous localization error," *IEEE transactions on pattern analysis and machine intelligence*, vol. 34, no. 8, pp. 1520–1532, 2011.
- [15] Y. Aoki, H. Goforth, R. A. Srivatsan, and S. Lucey, "Pointnetlk: Robust & efficient point cloud registration using pointnet," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.
- [16] Y. Li, R. Bu, M. Sun, W. Wu, X. Di, and B. Chen, "Pointcnn: Convolution on x-transformed points," *Advances in neural information processing systems*, vol. 31, pp. 820–830, 2018.
- [17] H. Lei, N. Akhtar, and A. Mian, "Spherical kernel for efficient graph convolution on 3d point clouds," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.
- [18] M. Alexa, J. Behr, and D. Cohen, "Point set surfaces," in *Visualization*, 2001, pp. 21–28.
- [19] Z. Chen, T. Zhang, J. Cao, Y. J. Zhang, and C. Wang, "Point cloud resampling using centroidal voronoi tessellation methods," *Computer-Aided Design*, vol. 102, pp. 12–21, 2018.
- [20] S. Zhong, Z. Zhong, and J. Hua, "Surface reconstruction by parallel and unified particle-based resampling from point clouds," *Computer Aided Geometric Design*, vol. 71, pp. 43–62, 2019.
- [21] C. Moening and N. Dodgson, "Intrinsic point cloud simplification," in *International conference on computer graphics and vision*, 2004, pp. 1147–1154.
- [22] K. Lee, H. Woo, and T. Suk, "Point data reduction using 3d grids," *The International Journal of Advanced Manufacturing Technology*, vol. 18, no. 3, pp. 201–210, 2001.
- [23] R. Bridson, "Fast poisson disk sampling in arbitrary dimensions," *SIGGRAPH sketches*, vol. 10, p. 1, 2007.
- [24] L.-Y. Wei, "Parallel poisson disk sampling," *Acm Transactions On Graphics (tog)*, vol. 27, no. 3, pp. 1–9, 2008.
- [25] J.-Y. Sim, S.-U. Lee, and C.-S. Kim, "Construction of regular 3d point clouds using octree partitioning and resampling," in *2005 IEEE International Symposium on Circuits and Systems*. IEEE, 2005, pp. 956–959.
- [26] Z. Xiao and W. Huang, "Kd-tree based nonuniform simplification of 3d point cloud," in *International Conference on Genetic and Evolutionary Computing*, 2009, pp. 339–342.
- [27] M. Kazhdan and H. Hoppe, "Screened poisson surface reconstruction," *ACM Transactions on Graphics (ToG)*, vol. 32, no. 3, pp. 1–13, 2013.
- [28] Y. Aoki, H. Goforth, R. A. Srivatsan, and S. Lucey, "Pointnetlk: Robust & efficient point cloud registration using pointnet," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 7163–7172.
- [29] S. Vinit, X. Li, H. Goforth, Y. Aoki, R. Srivatsan, S. Lucey, and H. Choset, "Pcnet: Point cloud registration network using pointnet encoding," in *Proceedings of the International Conference on Computer Vision (ICCV)*, Seoul, Korea, vol. 2, 2019, p. 3.
- [30] M. Pauly, M. Gross, and L. P. Kobbelt, "Efficient simplification of point-sampled surfaces," in *IEEE Visualization*, 2002, pp. 163–170.
- [31] Y. Miao, R. Pajarola, and J. Feng, "Curvature-aware adaptive resampling for point-sampled geometry," *Computer-Aided Design*, vol. 41, no. 6, pp. 395–403, 2009.
- [32] B.-Q. Shi, J. Liang, and Q. Liu, "Adaptive simplification of point cloud using k-means clustering," *Computer-Aided Design*, vol. 43, no. 8, pp. 910–922, 2011.
- [33] H. Huang, D. Li, H. Zhang, U. Ascher, and D. Cohen-Or, "Consolidation of unorganized point clouds for surface reconstruction," *ACM transactions on graphics (TOG)*, vol. 28, no. 5, pp. 1–7, 2009.
- [34] R. B. Rusu, N. Blodow, and M. Beetz, "Fast point feature histograms (fpfh) for 3d registration," in *2009 IEEE international conference on robotics and automation*. IEEE, 2009, pp. 3212–3217.
- [35] P. Biber and W. Straßer, "The normal distributions transform: A new approach to laser scan matching," in *Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003)(Cat. No. 03CH37453)*, vol. 3. IEEE, 2003, pp. 2743–2748.
- [36] X. Ying, S.-Q. Xin, Q. Sun, and Y. He, "An intrinsic algorithm for parallel poisson disk sampling on arbitrary surfaces," *IEEE transactions on visualization and computer graphics*, vol. 19, no. 9, pp. 1425–1437, 2013.
- [37] Y.-J. Liu, D. Fan, C.-X. Xu, and Y. He, "Constructing intrinsic delaunay triangulations from the dual of geodesic voronoi diagrams," *ACM Transactions on Graphics (TOG)*, vol. 36, no. 2, pp. 1–15, 2017.
- [38] X. Wang, X. Ying, Y.-J. Liu, S.-Q. Xin, W. Wang, X. Gu, W. Mueller-Wittig, and Y. He, "Intrinsic computation of centroidal voronoi tessellation (cvt) on meshes," *Computer-Aided Design*, vol. 58, pp. 51–61, 2015.
- [39] C. Luo, X. Ge, and Y. Wang, "Uniformization and density adaptation for point cloud data via graph laplacian," *Computer Graphic Fourm*, vol. 37, no. 1, pp. 325–337, 2018.
- [40] S. Chen, D. Tian, C. Feng, A. Vetro, and J. Kovačević, "Fast resampling of three-dimensional point clouds via graphs," *IEEE Transactions on Signal Processing*, vol. 66, no. 3, pp. 666–681, 2017.
- [41] D. Boltcheva and B. Lévy, "Surface reconstruction by computing restricted voronoi cells in parallel," *Computer-Aided Design*, vol. 90, pp. 123–134, 2017.
- [42] Z. Chen, T. Zhang, and J. C. et al, "Point cloud resampling using centroidal voronoi tessellation methods," *Computer-Aided Design*, vol. 102, p. 12–21, 2018.
- [43] N. Ray, D. Sokolov, S. Lefebvre, and B. Lévy, "Meshless voronoi on the gpu," *ACM Transactions on Graphics (TOG)*, vol. 37, no. 6, pp. 1–12, 2018.
- [44] Z. Zhong, X. Guo, W. Wang, B. Lévy, F. Sun, Y. Liu, W. Mao et al., "Particle-based anisotropic surface meshing," *ACM Trans. Graph.*, vol. 32, no. 4, pp. 99–1, 2013.
- [45] S. Ni, Z. Zhong, J. Huang, W. Wang, and X. Guo, "Field-aligned and lattice-guided tetrahedral meshing," in *Computer Graphics Forum*, vol. 37, no. 5. Wiley Online Library, 2018, pp. 161–172.
- [46] Z. Chen, M. Qi, and T.-S. Tan, "Computing delaunay refinement using the gpu," in *Proceedings of the 21st ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*, 2017, pp. 1–9.
- [47] M. Alexa and A. Adamson, "On normals and projection operators for surfaces defined by point sets," *SPBG*, vol. 4, pp. 149–155, 2004.
- [48] D. Levin, "Mesh-independent surface interpolation," in *Geometric modeling for scientific visualization*. Springer, 2004, pp. 37–49.
- [49] J. Digne and C. De Franchis, "The bilateral filter for point clouds," *Image Processing On Line*, vol. 7, pp. 278–287, 2017.
- [50] G. Rosman, A. Dubrovina, and R. Kimmel, "Patch-collaborative spectral point-cloud denoising," in *Computer Graphics Forum*, vol. 32, no. 8. Wiley Online Library, 2013, pp. 1–12.
- [51] T. Jia, K. Wang, Z. Wu, J. Zhao, P. Xu, C. Liu, and M. Zhou, "Isometric shape matching based on the geodesic structure and minimum cost flow," in *2014 International Conference on Cyber-worlds*. IEEE, 2014, pp. 122–129.
- [52] R. Kimmel and J. A. Sethian, "Computing geodesic paths on manifolds," *Proceedings of the national academy of Sciences*, vol. 95, no. 15, pp. 8431–8435, 1998.
- [53] Y. Qin, X. Han, H. Yu, Y. Yu, and J. Zhang, "Fast and exact discrete geodesic computation based on triangle-oriented wavefront propagation," *ACM Transactions on Graphics (TOG)*, vol. 35, no. 4, pp. 1–13, 2016.
- [54] K. Crane, C. Weischedel, and M. Wardetzky, "Geodesics in heat: A new approach to computing distance based on heat flow," *ACM Transactions on Graphics (TOG)*, vol. 32, no. 5, pp. 1–11, 2013.

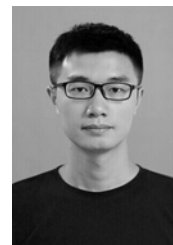
- [55] K. Zhou, Q. Hou, R. Wang, and B. Guo, "Real-time kd-tree construction on graphics hardware," *ACM Transactions on Graphics (TOG)*, vol. 27, no. 5, pp. 1–11, 2008.
- [56] S. Arya, D. M. Mount, N. S. Netanyahu, R. Silverman, and A. Y. Wu, "An optimal algorithm for approximate nearest neighbor searching fixed dimensions," *Journal of the ACM (JACM)*, vol. 45, no. 6, pp. 891–923, 1998.
- [57] M. Botsch and L. Kobbelt, "A remeshing approach to multiresolution modeling," in *Eurographics symposium on Geometry processing*, 2004, pp. 185–192.
- [58] M. Duniach, D. Vanderhaeghe, L. Barthe, and M. Botsch, "Adaptive remeshing for real-time mesh deformation," in *EUROGRAPHICS-ICS*, 2013, pp. 029–032.
- [59] Q. Mérigot, M. Ovsjanikov, and L. J. Guibas, "Voronoi-based curvature and feature estimation from point clouds," *IEEE Transactions on Visualization and Computer Graphics*, vol. 17, no. 6, pp. 743–756, 2010.
- [60] C. Lv, W. Lin, and B. Zhao, "Voxel structure-based mesh reconstruction from a 3d point cloud," *IEEE Transactions on Multimedia*, 2021.
- [61] D. Cohen-Steiner and F. Da, "A greedy delaunay-based surface reconstruction algorithm," *The visual computer*, vol. 20, no. 1, pp. 4–16, 2004.
- [62] P. J. Besl and N. D. McKay, "Method for registration of 3-d shapes," in *Sensor fusion IV: control paradigms and data structures*, vol. 1611. International Society for Optics and Photonics, 1992, pp. 586–606.
- [63] M. Levoy, K. Pulli, B. Curless, S. Rusinkiewicz, D. Koller, L. Pereira, M. Ginzton, S. Anderson, J. Davis, J. Ginsberg *et al.*, "The digital michelangelo project: 3d scanning of large statues," in *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, 2000, pp. 131–144.
- [64] A. Bronstein, M. Bronstein, U. Castellani, B. Falcidieno, A. Fusiello, A. Godil, L. Guibas, I. Kokkinos, Z. Lian, M. Ovsjanikov *et al.*, "Shrec 2010: robust large-scale shape retrieval benchmark," *Proc. 3DOR*, vol. 5, no. 4, 2010.
- [65] K. Lai, L. Bo, X. Ren, and D. Fox, "Rgb-d object recognition: Features, algorithms, and a large scale benchmark," in *Consumer Depth Cameras for Computer Vision*. Springer, 2013, pp. 167–192.
- [66] M.-J. Rakotosaona, V. La Barbera, P. Guerrero, N. J. Mitra, and M. Ovsjanikov, "Pointcleannet: Learning to denoise and remove outliers from dense point clouds," in *Computer Graphics Forum*, vol. 39, no. 1. Wiley Online Library, 2020, pp. 185–203.
- [67] Y. Lipman, D. Cohen-Or, D. Levin, and H. Tal-Ezer, "Parameterization-free projection for geometry reconstruction," *ACM Transactions on Graphics (TOG)*, vol. 26, no. 3, pp. 22–es, 2007.
- [68] A. C. Öztireli, G. Guennebaud, and M. Gross, "Feature preserving point set surfaces based on non-linear kernel regression," in *Computer graphics forum*, vol. 28, no. 2. Wiley Online Library, 2009, pp. 493–501.
- [69] L. Yu, X. Li, C.-W. Fu, D. Cohen-Or, and P.-A. Heng, "Pu-net: Point cloud upsampling network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 2790–2799.
- [70] H. Huang, S. Wu, M. Gong, D. Cohen-Or, U. Ascher, and H. Zhang, "Edge-aware point set resampling," *ACM transactions on graphics (TOG)*, vol. 32, no. 1, pp. 1–12, 2013.
- [71] P. J. Frey and H. Borouchaki, "Surface mesh quality evaluation," *International journal for numerical methods in engineering*, vol. 45, no. 1, pp. 101–118, 1999.
- [72] D.-M. Yan, B. Lévy, Y. Liu, F. Sun, and W. Wang, "Isotropic remeshing with fast and exact computation of restricted voronoi diagram," in *Computer graphics forum*, vol. 28, no. 5. Wiley Online Library, 2009, pp. 1445–1454.
- [73] M. Berger, J. A. Levine, L. G. Nonato, G. Taubin, and C. T. Silva, "A benchmark for surface reconstruction," *ACM Transactions on Graphics (TOG)*, vol. 32, no. 2, pp. 1–17, 2013.
- [74] J. Digne, J.-M. Morel, C.-M. Souzani, and C. Lartigue, "Scale space meshing of raw data point sets," in *Computer Graphics Forum*, vol. 30, no. 6. Wiley Online Library, 2011, pp. 1630–1642.
- [75] F. Lafarge and P. Alliez, "Surface reconstruction through point set structuring," in *Computer Graphics Forum*, vol. 32, no. 2pt2. Wiley Online Library, 2013, pp. 225–234.
- [76] M. Corsini, P. Cignoni, and R. Scopigno, "Efficient and flexible sampling with blue noise properties of triangular meshes," *IEEE transactions on visualization and computer graphics*, vol. 18, no. 6, pp. 914–924, 2012.
- [77] J. Yang, H. Li, and Y. Jia, "Go-icp: Solving 3d registration efficiently and globally optimally," in *Proceedings of the IEEE International Conference on Computer Vision*, 2013, pp. 1457–1464.



Chenlei Lv received PhD degree in College of information science and technology, Beijing Normal University (BNU). He is currently a Post-doctor in School of Computer Science and Engineering, Nanyang Technological University (NTU). His research interests include Computer Vision, 3D Biometrics, Computer Graphics, Discrete Differential Geometry and Conformal Geometric. He has published several papers in IEEE Transactions on Image Processing, IEEE Transactions on Multimedia, Pattern Recognition, et al. He also be a reviewer for Pattern Recognition and NEUROCOMPUTING. Recently, he joins in a point cloud-based research project. The personal page of the link is: <https://aliexken.github.io/>.



Weisi Lin (M'92–SM'98–F'16) 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 published over 200 journal papers, over 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 over 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, the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, the IEEE TRANSACTIONS ON MULTIMEDIA, and the IEEE SIGNAL PROCESSING LETTERS.



Baoquan Zhao is currently an associate professor at the School of Artificial Intelligent, Sun Yat-sen University, China. Prior to his current position, he was a Research Fellow at the School of Computer Science and Engineering (SCSE), Nanyang Technological University. He received his Ph.D. degree in computer science from Sun Yat-sen University. He has served as a reviewer and technical program committee member of several journals and international conferences. He is a recipient of the Outstanding Reviewer Award of 2020 IEEE ICME. His research interests include point cloud processing and compression, visual information analysis, multimedia systems and applications, etc.