

Adaptive simplification of point cloud using k -means clustering

Bao-Quan Shi^{a,*}, Jin Liang^a, Qing Liu^b

^a School of Mechanical Engineering, Xi'an Jiaotong University, PR China

^b School of Medicine, Xi'an Jiaotong University, PR China

ARTICLE INFO

Article history:

Received 22 May 2010

Accepted 2 April 2011

Keywords:

Simplification
Recursive subdivision
3D point cloud
 k -means clustering

ABSTRACT

3D scanning devices usually produce huge amounts of dense points, which require excessively large storage space and long post-processing times. This paper presents a new adaptive simplification method to reduce the number of the scanned dense points. An automatic recursive subdivision scheme is designed to pick out representative points and remove redundant points. It employs the k -means clustering algorithm to gather similar points together in the spatial domain and uses the maximum normal vector deviation as a measure of cluster scatter to partition the gathered point sets into a series of sub-clusters in the feature field. To maintain the integrity of the original boundary, a special boundary detection algorithm is developed, which is run before the recursive subdivision procedure. To avoid the final distribution of the simplified points to become locally greedy and unbalanced, a refinement algorithm is put forward, which is run after the recursive subdivision procedure. The proposed method may generate uniformly distributed sparse sampling points in the flat areas and necessary higher density in the high curvature regions. The effectiveness and performance of the novel simplification method is validated and illustrated through experimental results and comparison with other point sampling methods.

© 2011 Elsevier Ltd. All rights reserved.

1. Introduction

With the development of modern industrial manufacturing, 3D scanning technologies and devices are widely applied in product design, manufacturing [1] and quality assessment [2]. Generally, either contact or non-contact scanning devices are employed in original data acquisition. The contact scanning devices have better accuracy, but the data acquisition process is very slow when they are used for scanning large or complex freeform surfaces. In contrast, the non-contact scanning devices such as 3D-laser scanners [3] and structured light scanners [4] are extremely fast in data acquisition, but they have lower accuracy compared to that of contact scanning devices [5]. Owing to significant improvements in recent years, the accuracy of these devices can partially meet the demands of online inspection [6]. However, these scanning devices usually generate huge amounts of dense points; and then, large storage space and long post-processing times are needed in later applications, such as registration, integration, 3D surface reconstruction and inspection etc. Thus efficient and accurate simplification of point cloud is necessary. Fig. 1 shows the typical process flow of scanned points in the production line. It can be seen that the simplification module is important since the simplification results are directly applied in reverse engineering and/or inspection.

1.1. Previous research

Given a point cloud with or without normal vector information, a large number of algorithms have been developed to prune redundant points. Some reported methods are based on polygonal meshes; some other approaches directly simplify range images or 3D point clouds. According to the topology of point sets to be simplified, the existing simplification algorithms can be classified into the following two main categories or a combination of them: Polygonal mesh-based methods and point cloud-based methods. The former types of simplification methods triangulate each range image or 3D point cloud, and then remove redundant points according to some rules. Informative reviews on these types of algorithms are available in the literature [7,8].

Point cloud-based methods directly simplify range images or point clouds without constructing polygonal meshes. Since there is a lack of topological structures, it is really a difficult task to remove redundant points while keeping small features. Martin et al. [9] reported a data reduction method using a uniform grid in their EU Copernicus Project. Their method adopts a median filtering approach, which has been widely used in image processing. By building a grid structure, all the input points are assigned to a given grid. Then a median point is selected to represent data points belonging to that cell. Lee et al. [5] proposed one-directional or bi-directional non-uniform grid methods. However these methods are only suitable for range images.

Alexa et al. [10] presented moving least squares (MLS) based simplification method. In their work, the distances of each point to the MLS surface are computed, and then the

* Corresponding address: No. 28, Xianning West Road, Xi'an, Shaanxi, 710049, PR China. Tel.: +86 15991646030; fax: +86 029 82669103.

E-mail address: xjtuqb@163.com (B.-Q. Shi).

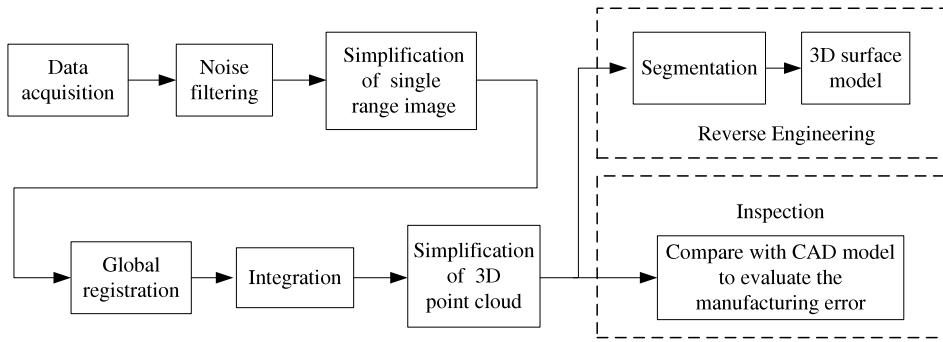


Fig. 1. General procedure of reverse engineering and inspection.

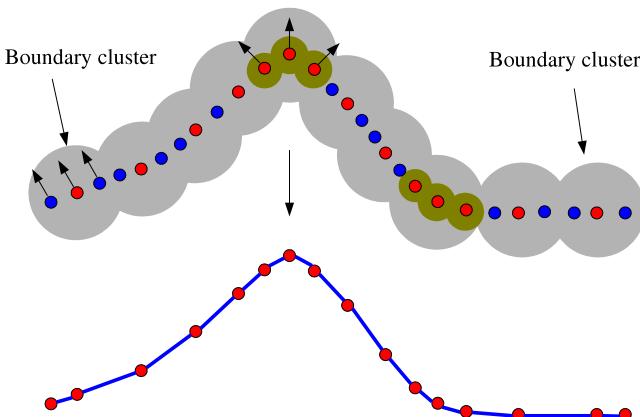


Fig. 2. Principle of the k -means clustering simplification method. The big gray circles represent the initial clusters, the smaller brown circles represent subdivided sub-clusters, the red points represent the cluster centroids, and the blue points are cluster members. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

points with smallest distance values are removed. Scheidegger et al. [11] extended the MLS approach to simplify point set surfaces and constructing a high-quality triangulation. Moenning and Dodgson [12] introduced the fast marching farthest point sampling principle, which is applicable to both images and higher dimensional surfaces in triangulated, point cloud or implicit form. Pauly et al. [13] adapted and generalized some popular methods in mesh simplification to the point cloud case. Several methods, based on the principles of clustering, incremental region growing, and iterative simplification, were tested against the criteria of accuracy, sampling distribution, and computational efficiency. Wu and Kobbelt [14] proposed the optimized elliptical splats. In their work, an approximately minimal set of splats, staying below a globally prescribed maximum error tolerance, are calculated to replace the entire surface. Song and Feng [15] proposed a global clustering approach to simplify point clouds. A global optimal result is obtained by minimizing the geometric deviation between the input and simplified point sets.

Recently, chordal lengths, spatial distances of neighboring points and normal variations have been widely used for point sampling. Kalaiah and Varshney [16] measured the redundancy of each individual point based on the local geometric properties derived from the surrounding points. Linsen [17] defined the information content of a point as a weighted sum of factors including distances to its neighboring points, non-planarity, and change of normal vectors evaluated in the selected neighborhood. Lee et al. [18] presented a simplification method that reduces the number of points using part geometry information. In their work, the points are removed based on their normal vector values using 3D grids. Miao et al. [19] proposed a curvature-aware adaptive

sampling method. An adaptive mean-shift clustering scheme is designed to generate non-uniformly distributed sampling points. Kang et al. [20] presented a balanced feature-sensitive point sampling method. They balance the distribution of sample points by normalizing the number of sample points over the localized regions.

1.2. k -means clustering

The k -means clustering [21–23] is a method of cluster analysis which aims to partition or group n data points into k clusters, in which each point belongs to the cluster with the nearest mean. The k -means clustering can be considered to be the most important unsupervised learning approach, which is widely used in pattern recognition and machine intelligence. It is also extended to integrate multi-view range images [24]. The details of k -means cluster algorithms are described in Refs. [21–23].

1.3. Our work

The goal of simplification is to choose the representative points and remove redundant data points. In this paper, the k -means clustering theory [21–23], which has been widely applied in the pattern recognition and machine learning literatures, is extended to simplify dense points. As shown in Fig. 2, the point cloud is partitioned into k clusters (gray), whose centroids are marked in red. In the relatively flat regions, since the points in the cluster are similar enough to each other in both spatial domain and feature domain, any point, in the cluster, can be chosen to represent the cluster. Considering a uniform distribution is better for post-processing, such as surface reconstruction, the centroids of the clusters are selected. In the high curvature regions, points in the cluster are dissimilar to each other in the feature domain as the normal vector deviations are large. If only the centroids of the clusters are retained, small features may be lost. Therefore, the clusters should be recursively subdivided until the normal vector deviations of points in the sub-clusters are smaller than the user-defined threshold. In each subdivision, one cluster is divided into two sub-clusters. To accelerate convergence of the recursive subdivision, the two points with maximum normal vector deviation in the cluster are selected as new centroids. The user-defined threshold directly determines the number of newly generated sub-clusters. The smaller the user-defined threshold is, the greater is the number of generated sub-clusters, i.e. higher simplification accuracy is obtained if the user-defined threshold is smaller. However, the distribution of simplified points may be locally greedy and unbalanced if the user-defined threshold is small. Hence, a refinement procedure is proposed to balance the density. In addition, the boundary clusters should be noticed. If the boundary clusters are located in the flat regions, the true boundary may be shrunk since the clusters would not be subdivided due to small normal vector deviations. To

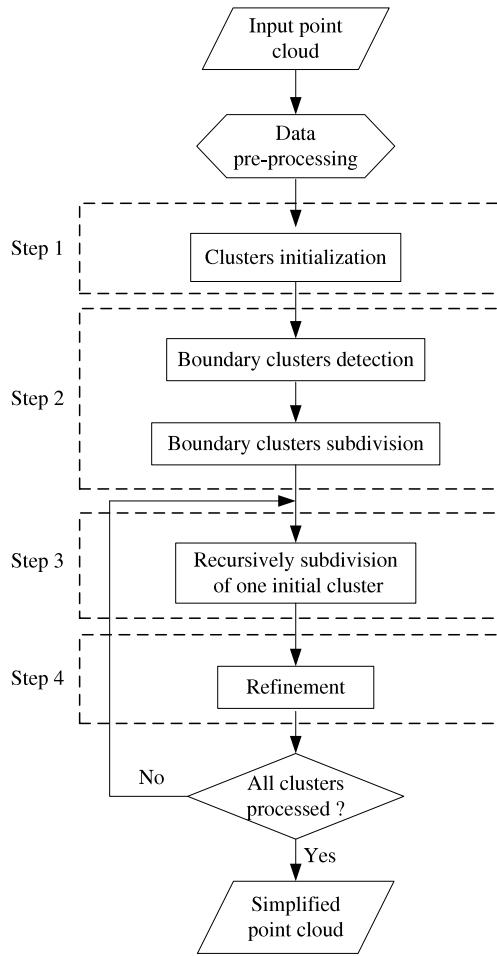


Fig. 3. Flowchart of the proposed k -means clustering simplification algorithm.

maintain the integrity of the original boundary, a special boundary detection algorithm is introduced. Thus, a desirable result can be obtained (Fig. 2). The flowchart of the proposed method is shown in Fig. 3.

The advantages of the novel algorithm, compared to other feature-sensitive simplification algorithms developed by Lee et al. [18], Miao et al. [19] and Kang et al. [20], are analyzed from several factors. Firstly, the novel algorithm takes into account the boundary of point sets thus maintaining the integrity of original boundary. Secondly, the novel algorithm balances the distribution of sampling points. Though the method presented by Kang et al. [20] produces well-balanced sample points, their method is complex and time-consuming since complex data structures such as Voronoi diagrams, fuzzy c -means clustering, octree and k - d tree are employed. In addition, the simplification result of Miao et al. [19] is not a true subset of the original point set.

The contributions are summarized as follows:

1. The k -means clustering [21–23], which has been widely applied in the pattern recognition and machine learning literatures, is extended to simplify point clouds.
2. A special boundary detection algorithm is developed to maintain the integrity of the original boundary and a refinement procedure is proposed to balance the density.
3. The novel algorithm can produce any level-of-detail balanced sampling points with two user specified parameters, while maintaining high simplification accuracy. The effectiveness and performance of the novel method is validated and illustrated through experimental results and comparison with other point sampling methods.

1.4. Evaluation of simplification error

To evaluate the accuracy of the simplified point set, the geometric error between the original and simplified point set should be measured. Cignoni et al. [25] developed the Metro tool to compare the difference between a pair of surfaces. Pauly et al. [13] and Miao et al. [19] adopted similar methods to evaluate simplification errors. In this paper, the maximum error and the average error between the original point set S and the simplified point set S' are measured:

$$\Delta_{\max}(S, S') = \max_{q \in S} |d(q, S')| \quad (1)$$

$$\Delta_{\text{avg}}(S, S') = \frac{1}{\|S\|} \sum_{q \in S} |d(q, S')|. \quad (2)$$

For each point $q \in S$, the geometric error $d(q, S')$ can then be defined as the Euclidean distance between the sampled point q and its projection point \bar{q} on the simplified surface S' . Assuming that N_p is the normal vector of point q , then the sign of $d(q, S')$ is the sign of $N_p * (\bar{q} - q)$. The projection point \bar{q} is calculated according to the procedure presented in Ref. [2]. Since mesh surface is required as a prerequisite [2], the simplified surface S' is triangulated in the reverse software geomagic studio [26].

1.5. Evaluation of unbalance

To quantify the unbalance, the simplified point set is triangulated in the reverse software geomagic studio [26] firstly. For a well balanced point set, the sizes of the generated meshes should be spread out over a small range of values i.e. the areas or side lengths of these meshes should be spread out over a small range of values. For an unbalanced point set, just the opposite is true. Thus the standard deviation of the triangle areas or side lengths can be employed to evaluate the unbalance of the simplified point surface.

In this paper, the standard deviation of the triangle side lengths is calculated to quantify the unbalance of the simplified point set:

$$\sigma = \sqrt{\frac{1}{N-1} \sum_{i=1}^N (L_i - \bar{L})^2} \quad (3)$$

where L_i is the i th triangle side length and \bar{L} is the average triangle side length. A low standard deviation σ indicates that the distribution of the simplified point set is globally balanced, whereas high standard deviation σ indicates that the distribution of the simplified point set is locally greedy and unbalanced.

2. Pre-processing

The aim of data pre-processing is to extract some useful information such as normal vector as well as to filter scanning noise of the input data. Since normal vector deviation is used as a measure of cluster scatter, it is necessary to estimate normal vectors before the clustering simplification process.

Hoppe et al. [27] presented an algorithm where the normal vector at each point is estimated as the normal vector of the fitting plane obtained by applying a least squares method to the k nearest neighbors of the point in the point cloud. Pauly et al. [13] extended the above method to estimate local surface properties such as normal vector and surface variation with a MLS kernel function. Mitra et al. [28] analyzed how the neighborhood size, curvature, sampling density, and noise affect the estimation of vertex normal vector by adopting a local least squares fitting approach. Hu et al. [29] propose a bilateral vertex normal vector estimation algorithm. OuYang et al. [30] introduced a 3D Voronoi diagram

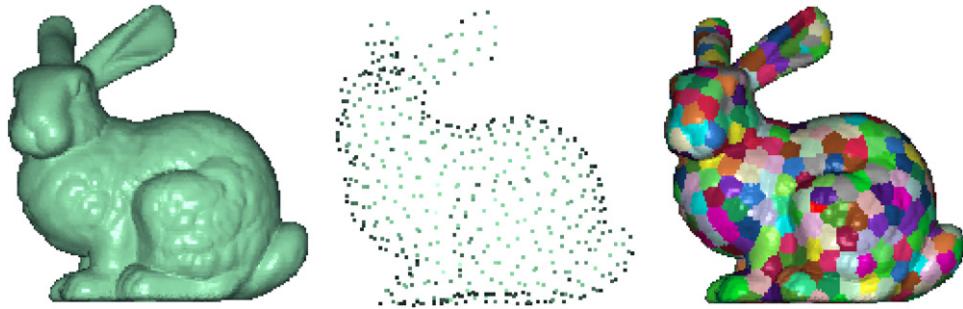


Fig. 4. Cluster initialization of the Stanford bunny. Left: input data. Middle: initialization of the cluster centroids. Right: initial clusters are formed, and one cluster is shown in one color.

based vertex normal vector estimation algorithm. Thurmer and Wuthrich [31] proposed a mean weighted by angle algorithm. Max [32] introduced a mean weighted by areas of adjacent triangles algorithm, a mean weighted by edge length reciprocals algorithm, and a mean weighted by square root of edge length reciprocals algorithm. A good comparison of these methods can be found in [33].

There are also many reported works that focus on denoising scanning noise. Taubin [34] proposed a signal processing approach to the problem of fairing arbitrary topology surface triangulations. Vollmer et al. [35] presented an HC algorithm. Fleishman et al. [36] introduced a bilateral mesh smoothing algorithm. Desbrun et al. [37] presented a mean curvature flow denoising algorithm. Hu et al. [38] proposed a trilateral point filtering algorithm.

In this paper, the mean weighted by areas of adjacent triangles algorithm [32] is used to estimate the normal vectors according to the summary in [33]. The bilateral smoothing algorithm [36] is employed to filter scanning noise. The other normal vector estimation methods and denoising methods can also be used instead.

3. Clustering simplification process

The proposed method can handle uniformly and non-uniformly distributed single 3D point clouds. It can also handle multiple point sets. However, the multiple point sets must be registered under a single coordinate system. The clustering simplification process includes four steps: cluster initialization, boundary clusters detection, recursive subdivision and refinement.

3.1. Cluster initialization

In the k -means clustering algorithm [21–23], initialization of the cluster centroids is vital, since different cluster initializations yield different results [39]. In the section, uniformly distributed k points are selected as initial centroids, details of the whole procedure are described as following:

1. Build a k - d tree [40] using the N input points in R^3 .
2. For $i = 1$ to N do
3. If P_i is non-marked, search the fixed radius neighbors of P_i , here the fixed radius (space interval— DT) should be manually specified by users.
4. Mark those fixed radius neighbors.
5. End if.
6. End for
7. Select non-marked points as initial cluster centroids.

The initial cluster centroids, generated by the above procedure, may be influenced by the order of the input points. To avoid this shortcoming, other uniform sampling methods can also be used instead, such as dividing the bounding box of the input point

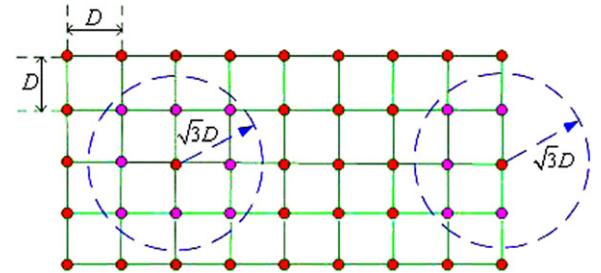


Fig. 5. Principle of boundary clusters detection in 2D space.

set into a series of sub-grids according to the user-defined space interval, and then picking out the points closest to the centers of these sub-grids as the initial cluster centroids. After initializing the cluster centroids, the standard k -means cluster procedure [21–23] is conducted to assign the other points to these centroids to form the initial clusters. Fig. 4 shows an example.

3.2. Boundary clusters detection

In the high curvature regions, the boundary clusters will be recursively subdivided into small sub-clusters. However, in the flat areas, if the centroids of the boundary clusters are far away from the boundary of the model (Fig. 2), the boundary will shrink and the accuracy would be lost after simplification. To avoid this phenomenon, the boundary clusters are detected and divided to required level. Since the initial cluster centroids are uniformly distributed, it is easy to detect the boundary clusters. Fig. 5 shows the principle of boundary clusters detection in 2D space. For each cluster centroid, a search circle is defined and then all of its 8 neighboring centroids are included into the search circle with a radius of $\sqrt{2}DT$ according to the Pythagorean Theorem. For security, the fixed radius is set to $\sqrt{3}DT$. If the number of neighboring centroids contained in the search circle is less than 6, the cluster must be a boundary cluster. In 3D space, a search sphere with radius of $\sqrt{3}DT$ is constructed and the above rule is extended to detect boundary clusters. Experimental results show that this simple rule could detect the boundary clusters efficiently in most situations, and though a few boundary clusters may be lost, the simplification accuracy is hardly affected.

After the boundary clusters are detected, whether a boundary cluster should be subdivided is judged. If the centroid of the detected boundary cluster is far away from the boundary of the model, it needs to be subdivided, and vice versa. However, it is a very difficult task to explicitly define “far” or “close”. In this section, an explicitly judgment criterion is developed. As shown in Fig. 6, Assuming c_i is a boundary cluster centroid and m_j is one member of the centroid c_i , and then the following relationships exist:

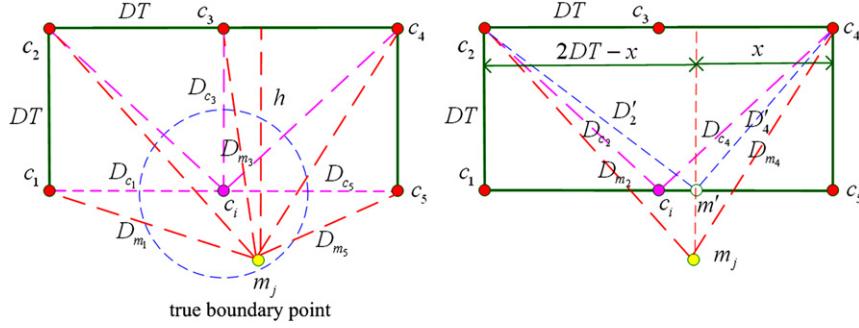


Fig. 6. The centroid of the boundary cluster far away from the true boundary.

$$D_{m_1} + D_{m_5} > D_{c_1} + D_{c_2} \quad (4)$$

$$D_{m_3} > h > D_{c_3} \quad (5)$$

$$D_{m_2} + D_{m_4} > D_{c_2} + D_{c_4} \quad (6)$$

where D_{ci} ($i = 1, 2, \dots, 5$) denotes the distances of c_i to its neighboring centroids c_j , D_{mj} ($j = 1, 2, \dots, 5$) denotes the distances of m_j to c_j , and h is the vertical distance of m_j to the edge (c_2, c_4) . The formulas (4) and (5) are obvious. The formula (6) can be proved as well.

The projection point of m_j on the edge (c_1, c_3) is calculated and marked as m' , the distances of m' to c_2 and c_4 are denoted as D'_2 and D'_4 respectively, then it is obvious that:

$$D_{m_2} + D_{m_4} > D'_2 + D'_4. \quad (7)$$

Supposing the distance of m' to c_5 is x , then the distance of m' to c_2 is $2DT - x$. According to the Pythagorean theorem, the addition of the distances of m' to c_2 and c_4 can be expressed as a function of x

$$F(x) = \sqrt{DT^2 + x^2} + \sqrt{(2DT - x)^2 + DT^2}. \quad (8)$$

The value of $F(x)$ is minimum when

$$F'(x) = 0. \quad (9)$$

The Eq. (9) has the only solution that $x = DT$, which means

$$D'_2 + D'_4 > D_{c_2} + D_{c_4}. \quad (10)$$

The formula (10) combined with formula (7) can derive the formula (6).

Finally, if one boundary cluster centroid is far away from the true boundary of the model, we can obtain the following basic relationship:

$$D_M = \frac{1}{N} \sum_{j=1}^N D_{mj} > D_C = \frac{1}{N} \sum_{j=1}^N D_{cj} \quad (11)$$

where D_C and D_M denote the average distances of c_i and m_j to c_j ($j = 1, 2, \dots, N$).

In this paper, the average distances of all the members in the boundary cluster are calculated and the maximum value is recorded:

$$\text{Max}D_M = \text{Max}(D_M). \quad (12)$$

If $\text{Max}D_M$ is greater than D_C , as described in Eq. (11), the boundary cluster centroid is considered as too far away from the border. For security, a scale factor, greater than 1.0, is multiplied by D_C :

$$\text{Max}D_M > \alpha D_C. \quad (13)$$

Thus, the boundary cluster, whose centroid is far away from the boundary of the model, is subdivided into two new sub-clusters in which one centroid is located at the original centroid point and the other in the centroid point of the cluster member whose average distance is $\text{Max}D_M$. According to experiments, if $\alpha = 7.0/6.0$ is better.

Fig. 7 shows an example. It can be seen that the initial centroids are distributed uniformly in space (Fig. 7(a)), and the sizes of the initial clusters are almost equal to each other (Fig. 7(b)). By observing the enlargement figure of the boundary clusters (Fig. 7(d)), we can see that some of the boundary cluster centroids are located at the true boundary of the model while some others are far away from the boundary, which indicate that the boundary detection step is necessary. The comparison of the simplification results with and without conducting the boundary detection procedure (Fig. 7(g) and (h)) also demonstrates the advantage of the proposed algorithm.

3.3. Cluster subdivision

Generally, in the flat areas, the members of the cluster are similar enough to each other in both the spatial domain and feature fields, thus the cluster centroid would be employed to represent the cluster. However, in the high curvature regions, the members of the cluster may be dissimilar to each other in the feature field since there may exist highly detailed features. And these features may be blurred if only the cluster centroid is preserved to represent the whole cluster. For accuracy, in the section, the initial clusters are recursively decomposed into smaller sub-clusters until the members are similar enough in both the spatial domain and feature fields.

Pauly et al. [13] introduced a hierarchical clustering method that splits the point cloud into smaller sub-sets in a top-down manner [41]. The cluster cell is split if either the size is larger than the user specified maximum cluster size or the variation above a maximum threshold. However, the splitting procedure starts from the original input data, which observes the point set as a big cluster, leading to an increase in computational time. Lee et al. [18] employed an octree based spatial decomposition method to recursively subdivide a 3D grid cube into 8 octants, and the criterion used for subdividing a cube is the standard deviation of the point normal vector values. Since a cube is divided into 8 smaller octants each time, there may be too many points preserved in high curvature region. In this section, the max normal vector deviation is applied to evaluate the similarity in feature field. The benefit is that it can not only determine whether the cluster members are similar enough, but can also locate the position of the dissimilarity. For obvious reasons, the members with the maximum deviations will be the ones with the most dissimilarity in the cluster. So these highly dissimilar members can be used as the centroids of the sub-clusters, which can accelerate the subdivision procedure.

In the cluster, firstly the normal vector deviations of the cluster members to each other are calculated. And then the maximum normal vector deviation is recorded. If the maximum normal vector deviation is larger than the user-defined normal vector deviation threshold NT (expressed as angles of the normal vector in our

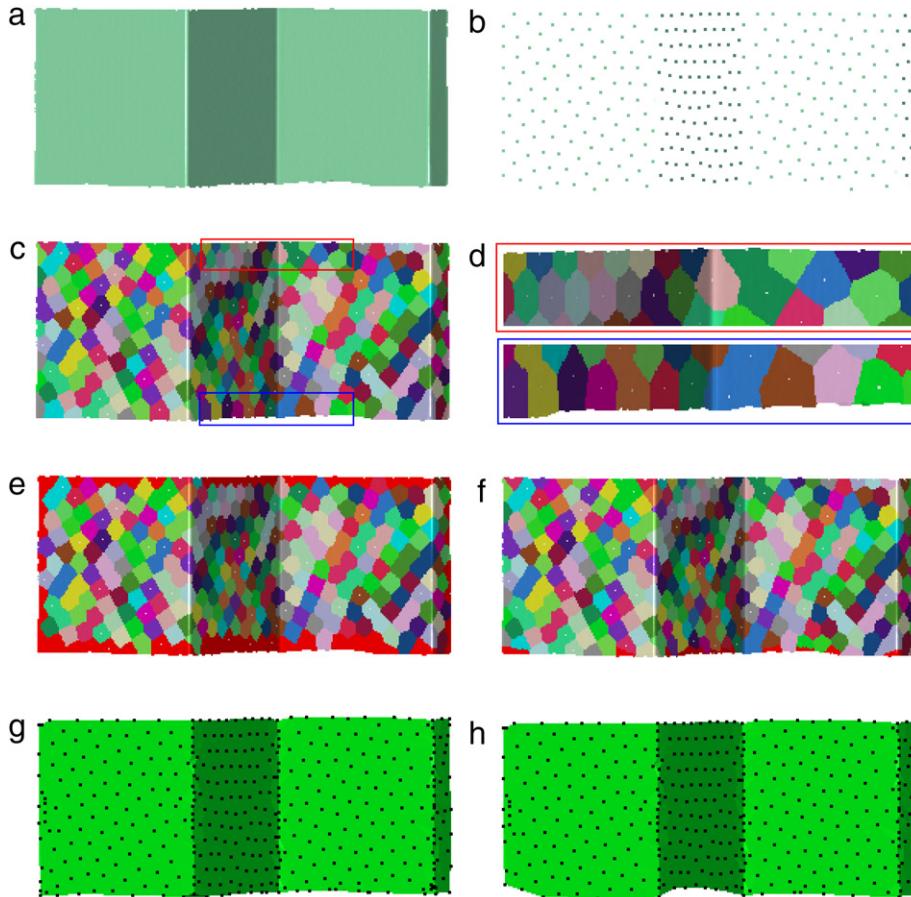


Fig. 7. Simplification of the T-block model using the proposed method with parameters of the space interval $DT = 10.0$ mm and the normal vector deviation threshold $NT = 0.25$. (a) The original date (194 663). (b) Initial Cluster Centroids. (c) Initial clusters formed by using standard k -means cluster procedure, one cluster is displayed in one color. (d) Enlargement of the boundary clusters. (e) Detected boundary clusters, shown in red. (f) Boundary cluster being divided, newly generated clusters are shown in red. (g) The final result (414). (h) The simplified result without conducting the boundary detection procedure (393). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Table 1
Number of the car model under different user-defined normal vector tolerance.

Normal vector tolerance	Number of centroids before subdivision	Number of centroids after subdivision	Number of points in the final results
0.2	1134	24503	2013
0.4	1134	9864	1659
0.6	1134	5427	1513
0.8	1134	3250	1394

implement), the cluster is subdivided into two sub-clusters. The splitting is done by the process using a binary space partition method as following:

1. In the cluster, the two members, whose normal vector deviations are maximum, are recorded and set as two new centroids.
2. The standard k -means cluster procedure [21–23] is conducted to assign the other members of the cluster to the new two centroids. Then two sub-clusters are generated.
3. Steps 1 and 2 are repeated to the newly generated sub-clusters until the termination conditions are met. The termination conditions are met when the maximum normal vector deviation is smaller than the user-defined tolerance NT or only one point is contained in the cluster.

Fig. 8(a)–(d) is an example of the recursive subdivision process, it can be seen that the initial clusters in the flat areas are kept as their original size, the initial clusters in the less flat regions are subdivided into a few larger sub-clusters, and the initial clusters in the high curvature regions are subdivided into lots of sub-clusters.

The simplification result (Fig. 8(e)) shows that the distribution is uniform in the flat areas and denser in the high curvature regions. This example illustrates the effectiveness of the proposed subdivision procedure.

The method of choosing the normal vector tolerance NT also imposes a considerable effect on the final simplification results. As the normal vector tolerance increases, fewer clusters will be subdivided, and even though some are subdivided, fewer sub-clusters will be generated, as shown in Table 1. Meanwhile, high computational efficiency and high reduction ratio are achieved. However, the small features of the model may be blurred, and the simplification error is increased. These phenomenon are shown in Fig. 8(e)–(j), from which we can see that with the increase of the normal vector tolerance NT (0.2–0.8), fewer points are preserved in the high curvature regions and the details of the car model become vague. Fig. 9 gives the simplification errors, from which we can see that the larger the user-defined normal vector tolerance is, the larger is the simplification error. According to the experimental results, if the normal vector tolerance is set between 0.2 and 0.4 it is better.

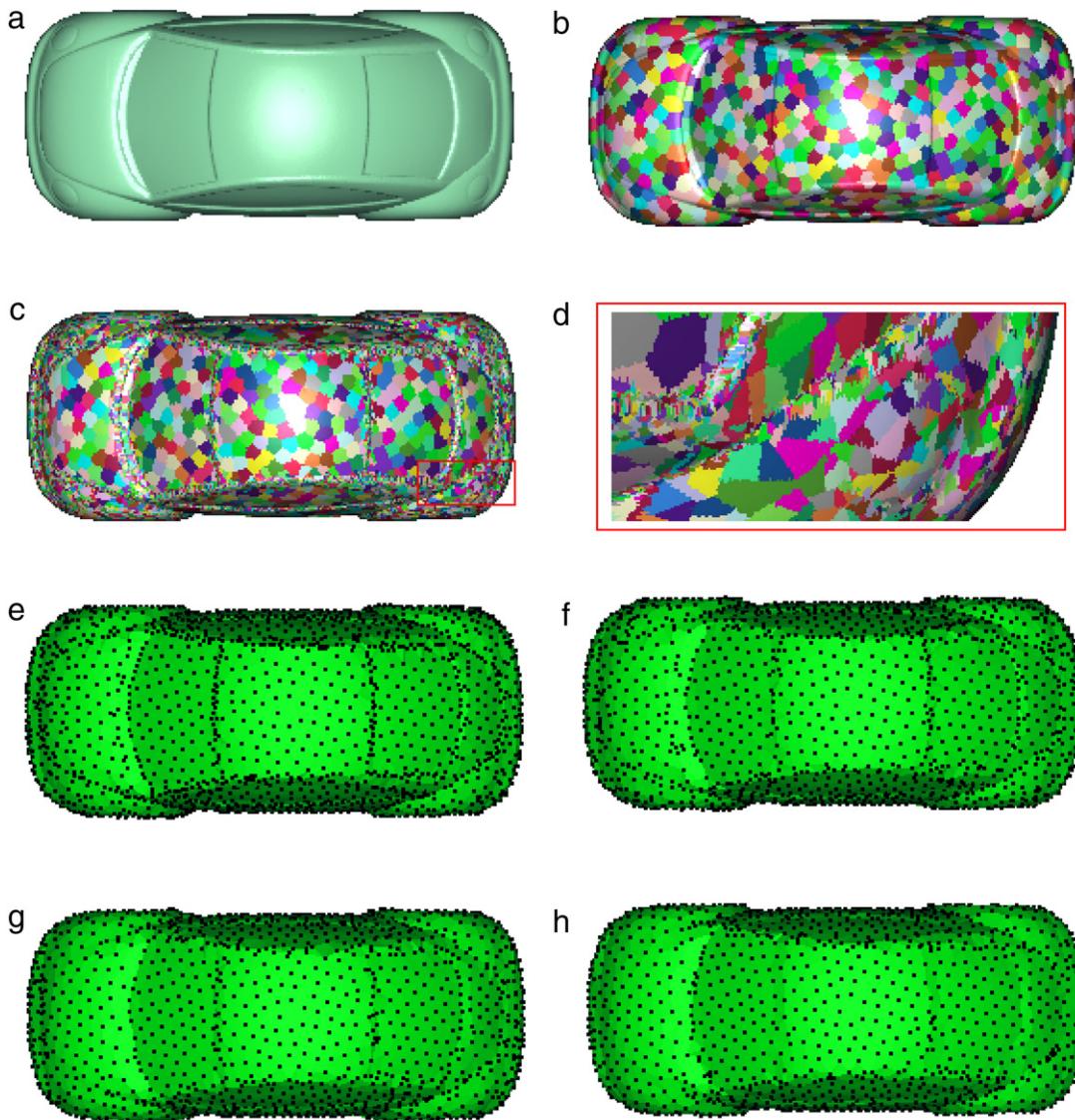


Fig. 8. Simplification of a car model uses the new method with space interval $DT = 5.0$ mm. (a) Input point cloud with normal vector (318 992). (b) Initial clusters. (c) Subdivision of the initial clusters. (d) Enlargement of a corner of (c). (e), (f), (g), (h): Simplification results with different normal vector deviation thresholds ($NT = 0.2, 0.4, 0.6, 0.8$).

3.4. Cluster refinement

As a result of the cluster subdivision, many small clusters will be generated where the object geometry changes dramatically. This makes the density distribution unbalanced globally. This phenomenon is shown in Table 1, from which we can see that there are only 1134 centroids before subdivision, however 24503 centroids are generated after conducting the subdivision procedure ($NT = 0.2$). In Fig. 10, we can see that there are too many centroids gathered in the vicinity of the sharpened edges after subdivision. Kang et al. [31] presented a balancing program to reduce the density in the high curvature regions. However, their balancing procedure is difficult to implement and time consuming since it is heavily based on the CGAL library. In this paper, a simple and easy to implement refinement (balancing) algorithm is developed to balance the density.

As mentioned above, the subdivision procedure is like establishment of a binary tree [42]. According to the characteristics of a binary tree [42], a combination of the subdivision criterion, we can draw that greater the depth value of the sub-cluster, sharper is the feature. So the depth value of the sub-cluster can be used as

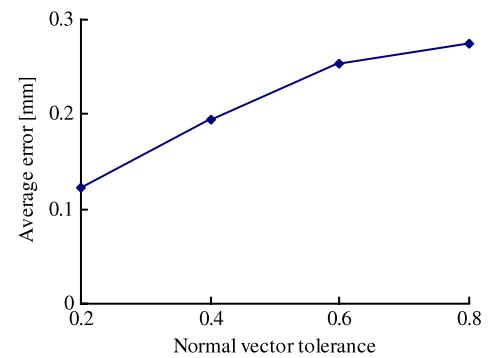


Fig. 9. Simplification errors of the car model under different normal vector tolerance.

a factor to evaluate the importance. In addition, if the depth values of the sub-clusters are identical, the sizes of these sub-clusters can be used as an additional factor. Because if the normal vector in the sub-cluster varies dramatically, it must be recursively subdivided into smaller sub-clusters so that the sizes become much

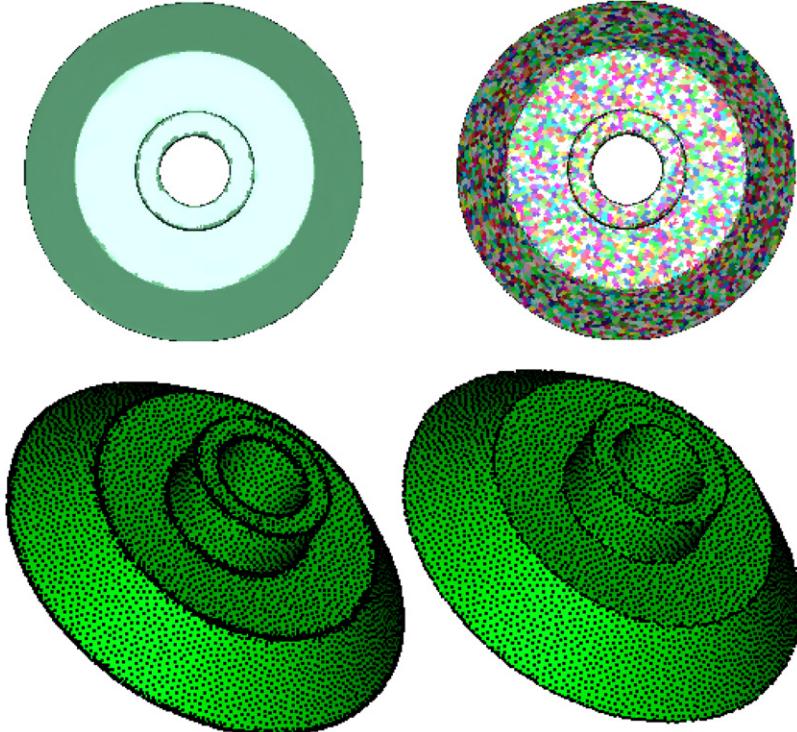


Fig. 10. Simplification of a Pulley model uses the new method with $DT = 3.0$ mm, $NT = 0.2$. Top left: input data (150 166). Top right: subdivision. Bottom left: centroids of clusters after subdivision (17 119). Bottom right: after refinement (11 348).

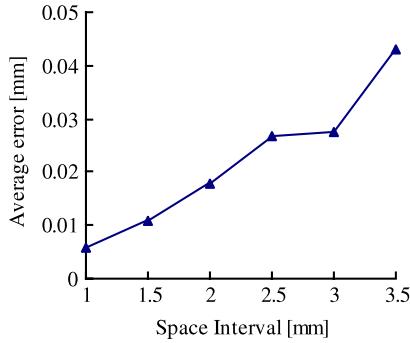


Fig. 11. Simplification errors of the Pulley model under different space intervals.

smaller (Fig. 8(d)). The importance of the sub-cluster is implicitly expressed as weight value:

$$W_i = Dep_i/nCount_i \quad (14)$$

where W_i is the weight (importance value) of the i th sub-cluster, Dep_i is the depth value of the i th sub-cluster, and $nCount_i$ is the size of the i th sub-cluster (the number of sub-cluster members).

For an initial cluster, if the height value is smaller than 3, we just consider that the refinement is needless. Otherwise, the sub-clusters are sorted according to their weight values. In order to get well-balanced sampling points, one initial cluster should contain no more than 4 sub-clusters after refinement. In addition, the preserved sub-cluster centroids should not be too close to the neighboring centroids of the initial cluster and also should not be too close to each other. Assuming the sub-cluster $subc_i$ and $subc_k$ are two members of the initial cluster c_i i.e. $subc_j \in c_i$, $subc_k \in c_i$, $subc_j \neq subc_k$, and c_j is the neighboring centroid of c_i . Then the above conditions can be described as:

$$d(subc_i, c_j) > \delta DT \quad (15)$$

$$d(subc_j, subc_k) > \delta DT \quad (16)$$

where δ is a scale factor and $\delta < 1.0$. In our implementation, the scale factor δ is set to $1/3$ so that the minimum distances of the preserved points should not be smaller than $DT/3$. Finally, to ensure that no gaps are generated among the preserved sub-clusters, three sub-clusters with high weight values and one sub-cluster with low weight values are retained. Fig. 10 shows an example, which illustrates the effectiveness of the refinement procedure.

Details of the refinement algorithm are described as follows:

1. If $Dep_{ci} > 2$, where Dep_{ci} is the depth value of the initial cluster c_i .
2. Sort the sub-clusters in descending order according to their weights.
3. If $d(subc_j, c_j) < \delta DT$ // in our implementation $\delta = 1/3$.
4. Remove the sub-cluster $subc_j$.
5. End if.
6. If $W_{subc_j} == W_{subc_1}, j \in [2, N]$, where N is the number of the sub-clusters.
7. Push $subc_j$ into the queue Q .
8. End if.
9. Iteratively compute $d(subc_j, subc_k), j \in [1, Q.size()], k \in [1, Q.size()], j \neq k$.
10. If $Maximum(d(subc_j, subc_k)) > \delta DT$
11. Preserve $subc_j$ and $subc_k$. //Keep two sub-clusters.
12. If $d(subc_j, subc_l) > \delta DT \&& d(subc_k, subc_l) > \delta DT$
13. Preserve $subc_l$. // $l \in [1, Q.size()]$
14. End if. // Keep another important sub-cluster.
15. If $d(subc_j, subc_m) > \delta DT \&& d(subc_k, subc_m) > \delta DT$
16. Preserve $subc_m$. // $m \in [Q.size(), N]$
17. End if. // Keep a sub-cluster in the flat area.
18. Else
19. Preserve $subc_1$
20. End if.
21. End if.

The level of simplification is mainly determined by two factors: the user-defined space interval DT and normal vector tolerance

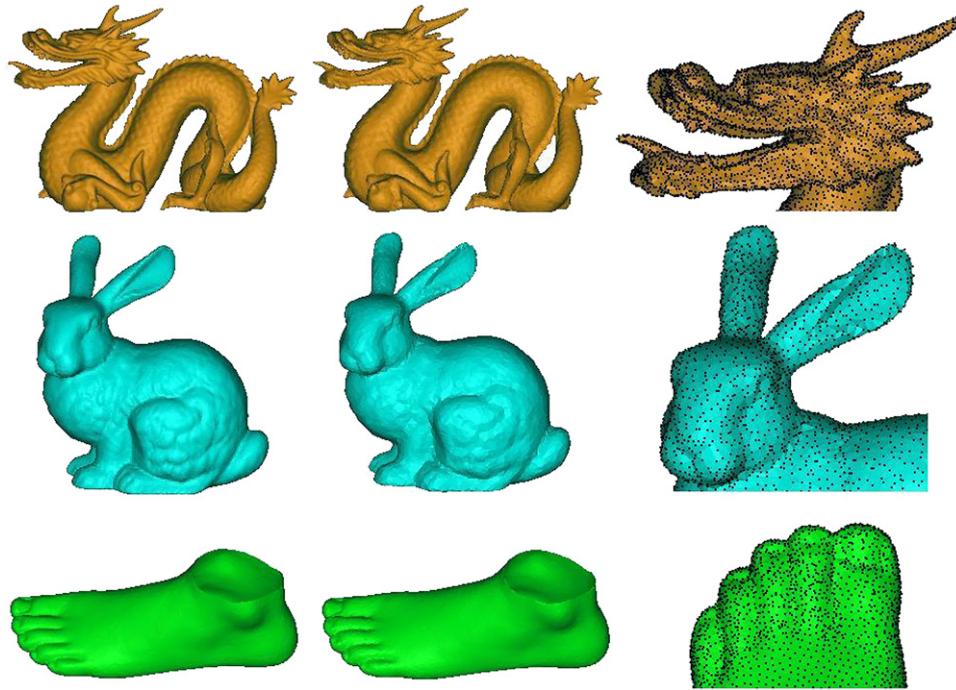


Fig. 12. Well-distributed sampling points simplified using the novel method. Left column: the original point sets (triangulated). Among them, the dragon model is non-uniformly distributed. Middle column: the simplified point cloud sets (triangulated). Right column: The zoom-in views of the corresponding sampling results.

NT . How the normal vector tolerance NT affects the final results has been discussed above. Fig. 11 shows the influence of the user-defined space interval DT to the simplification error.

4. Experimental results and discussion

The novel method was implemented using C++, MFC (Microsoft Fundamental Classes), and OpenGL. The program for the proposed method was run on an AMD Athlon 64 X2 dual core processor 4400++ PC. The Stanford dragon model and bunny model tested in this paper were developed by Greg Turk and Marc Levoy in 1994 at Stanford University [43]. The fandisk model and Moeller model were constructed by Hoppe [44]. The other models were scanned by the structured light scanner XJTUOM system [45]. For better visualization, the simplified point cloud sets were triangulated in the reverse software geomagic studio [26].

4.1. Results of our sampling methods

The novel method can produce feature-sensitive and well-balanced sampling points. Fig. 12 shows three examples where the dragon model was reduced to 7.6% of its original number, the bunny model was reduced to 18.7% and the Moeller model was reduced to 6.9%. However, the small features were well maintained in the simplified point cloud sets. In addition, the original dragon model was non-uniformly distributed, which indicates that the novel method can be adapted to simplify non-uniformly distributed point sets.

Fig. 13 shows three actual part models simplified by using the novel method. These models have sharp edges. The car door model was reduced to 25%, the mobile phone shell model was reduced to 10% and the lamp bracket model was reduced to 12.8%. The experimental results indicate that the novel method can preserve the sharp edges. In addition, the original boundaries were well maintained, which again illustrate the superiority of the novel method.

The novel method can produce any sparser level-of-detail point sets whilst preserving the small features and the sharp edges. In

Fig. 14, the sharp edges of the twist drill model can be clearly seen when the point set was reduced from 51k to 2k. In Fig. 15, the small features of the Fuwa Huanhuan model were well maintained though it was reduced to about 1% of its original number. These examples demonstrate the good performance of the proposed method.

Table 2 shows the time statistic for different example models. In case of the dragon model, the original number is 425 545. During the k -means clustering simplifying, the cluster initialization step takes 2.765 s; the boundary cluster detection step takes 0.141 s and the recursive subdivision and refinement step takes 1.969 s. Finally 32 925 points are preserved. This table illustrates the efficiency of the novel method.

4.2. Comparison with other point sampling methods

The 3D Grid method [18], the Curvature-Aware method [19] and the MLS based method [11] were employed for a comparative study. For the Curvature-Aware method, after conducting the iterative mean-shift clustering scheme, a standard hierarchical clustering program was executed to simplify redundant data. The executable package of the MLS based method [11] was provided by Dr. Carlos E. Scheidegger. The simplification results were triangulated with the software geomagic studio [26]. In Fig. 16, the famous fandisk model was simplified. Since there was no redundant data in the original model (vertices 6475, faces 12 946), we increased the vertexes with the geomagic studio [26]. Finally, the number of vertexes was 103 570. As shown in the figure, the novel method gives better results: uniformly distributed sparse sampling points in the flat areas and necessary dense points in the high curvature regions. The sharp edges of the fandisk model are well maintained. The 3D Grid method [18] can also preserve sharp edges but too many sampling points are assigned to the sharp edges and fewer in the flat areas, which lead to unbalance (the maximum edge deviation was obtained). When the simplified point set was triangulated in the reverse engineering software geomagic studio [26], holes were generated. Furthermore, some less obvious edges were blurred since the standard normal vector deviation is used as threshold. Due to the filtering property of the

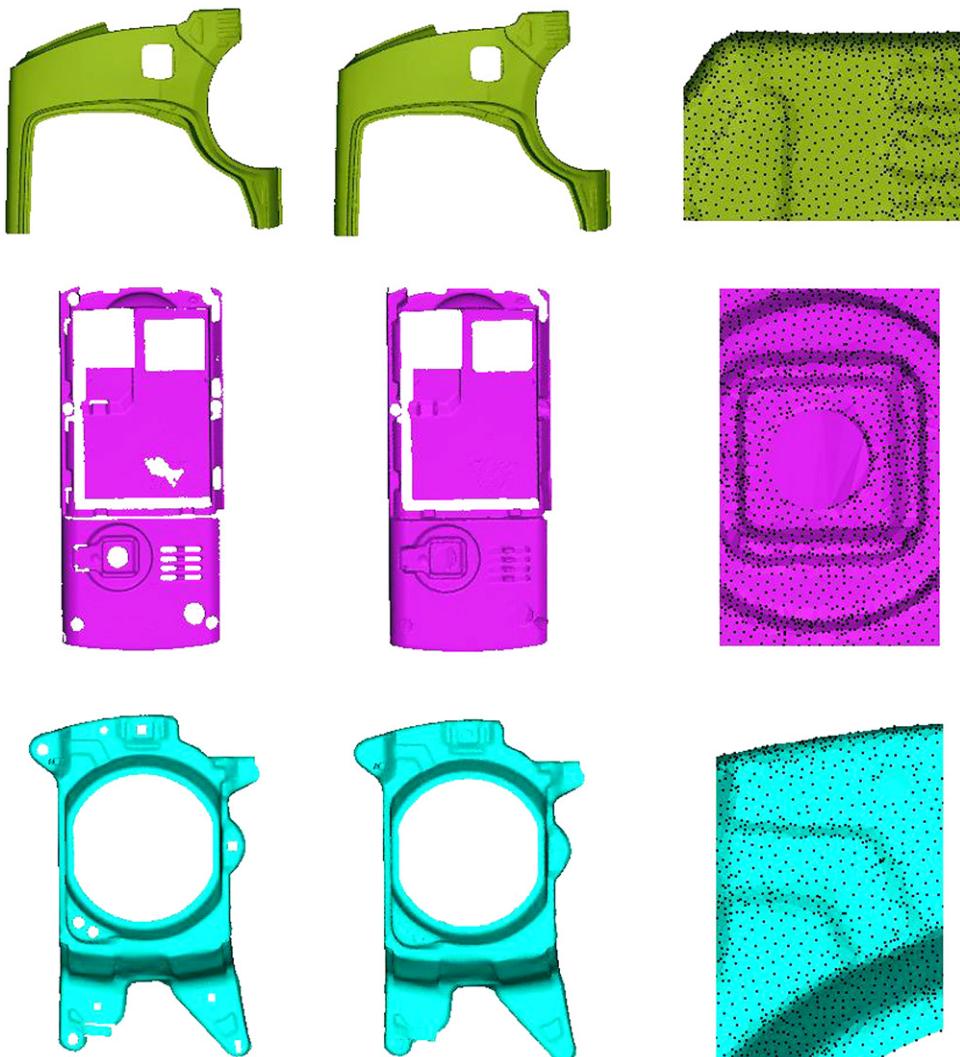


Fig. 13. Work pieces simplified using the novel method. Left column: the original point sets (triangulated). Middle column: the simplified point cloud sets (triangulated). Right column: The zoom-in views of the corresponding sampling results.

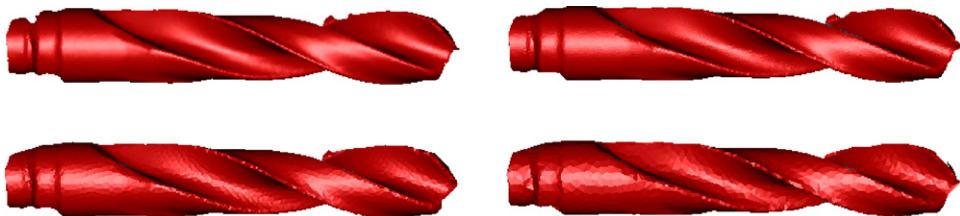


Fig. 14. Simplification of the twist drill model at different levels-of-detail (triangulated). From left to right, first row: 51k (original), 10k; second row: 4k and 2k points.

Table 2

Time statistic of the novel method for different point cloud sets. The time data was collected on a PC with an AMD Athlon 64 X2 dual core processor 4400++, 2.0 GB Memory.

Models	Original number	Timings for different steps			Preserved number
		Cluster initialization (s)	Boundary cluster detection (s)	Recursive subdivision and refinement (s)	
Dragon	435 545	2.765	0.141	1.969	32 925
Bunny	34 834	0.25	0.016	0.063	6500
Moeller	123 397	0.781	0.047	0.156	8494
Car door	111 843	0.703	0.125	0.094	28 250
Phone shell	379 578	2.062	0.204	0.39	38 168
Lamp bracket	123 581	0.75	0.062	0.266	15 809
Twist drill	51k	0.344	0.015	0.266	2k
Huanhuan	449k	3.875	0.063	4.000	10k



Fig. 15. Simplification of the Fuwa Huanhuan model at different levels-of-detail (triangulated). From left to right, first row: 449k (original), 44k and 21k points; second row: 10k and 5k points.

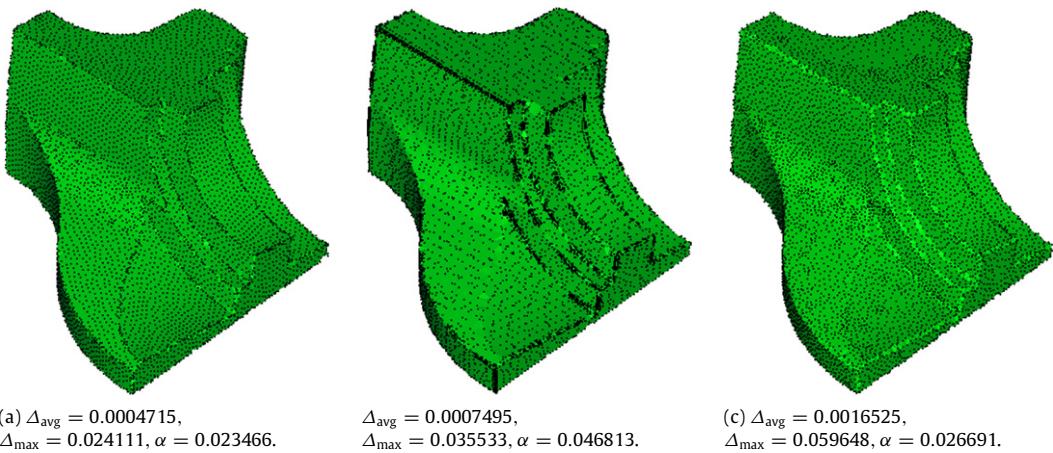


Fig. 16. Simplification results of the fandisk model (triangulated), the original number is 103570. (a) Simplified using the new method with parameters of space interval and normal vector deviation threshold as (0.076, 0.13) (9682); (b) simplified using the 3D Grid method [18] with parameters of space interval and standard normal vector deviation as (0.0696, 0.41) (9694); (c) simplified using the Curvature-Aware method [19] with parameter of k -neighborhood and angle threshold as (16, 15°) (9666).

mean shift clustering procedure, the Curvature-Aware method [18] seriously blurred the sharp edges. Since the MLS based method cannot handle sharp features [11], it was not employed to simplify the fandisk model.

Fig. 17 shows another example. The bowl model with small patterns was simplified. These small patterns are well maintained in the novel method. The 3D Grid method [18] blurred some of the small patterns. The boundary of the point set simplified with the Curvature-Aware [19] method contracted. The MLS

based method [11] produced the smoothest point set surface. The detailed distribution of the simplification errors is displayed in Fig. 18. The Curvature-Aware method [19] and the MLS based method [11] show higher simplification errors. The possible reason is that the two methods have the filtering property. Among the four feature sensitive methods, the novel method gives the best simplification result: low simplification error and low standard edge deviation. Once again, the experimental results demonstrated the good performance of the novel method.



Fig. 17. Simplification results of the bowl model (triangulated). (a) The original bowl model (63,104); (b) simplified using the new method with parameters (1.3, 0.135), finally preserved points 11,318, $\Delta_{\text{avg}} = 0.015606$, $\Delta_{\text{max}} = 0.256424$ and $\sigma = 0.512200$; (c) simplified using the 3D Grid method [18] with parameters (1.45, 0.10), finally preserved points 11,369, $\Delta_{\text{avg}} = 0.0178805$, $\Delta_{\text{max}} = 0.308435$ and $\sigma = 0.744606$; (d) simplified using the Curvature-Aware method [19] with parameters (16, 15°), finally preserved points 11,298, $\Delta_{\text{avg}} = 0.039012$, $\Delta_{\text{max}} = 0.947109$ and $\sigma = 0.868859$; (e) simplified using the MLS based method [11] with parameter of radius factor as (6.2), finally preserved points 11,696, $\Delta_{\text{avg}} = 0.0361295$, $\Delta_{\text{max}} = 0.277436$ and $\sigma = 0.636196$.

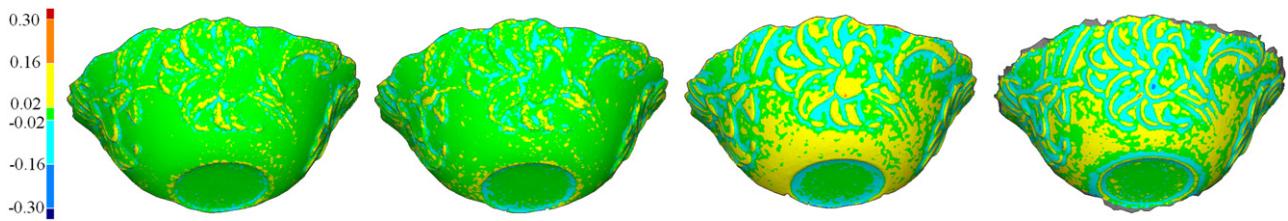


Fig. 18. Distribution of the simplification error of the simplified point sets in Fig. 17. From left to right: the new method, the 3D Grid method [18], the Curvature-Aware method [19] and the MLS based method [11].

5. Conclusions and further work

In this paper, the k -means clustering theory [21–23] which has been widely used in the pattern recognition and machine learning literatures has been extended to simplify 3D point sets. An automatic recursive subdivision scheme is designed to pick out representative points and remove redundant points. To maintain the integrity of the border, an automatic boundary cluster detection algorithm is developed. To avoid the final distribution of the simplified points to become locally greedy and unbalanced, a refinement algorithm is proposed. The new method is mainly impacted by the two factors: user-defined space interval and normal vector tolerance. By adjusting the two parameters, any level of details and amount can be obtained. Comprehensive examples and a comparative study illustrate the good performance of the novel method.

During the boundary cluster subdivision, the formula (13) is used as a judgment criterion to determine whether a boundary cluster is needed to be divided. However, we can also directly subdivide each detected boundary cluster to the required level without any judgment. Formula (14), formula (15) and formula (16) are derived based on analyzing the k -means clustering simplification process. It is difficult to prove these formulas with strict mathematical theory. In addition, since the maximum normal vector is employed, the proposed method is sensitive to noise. If the noise of the 3D point cloud is serious, effective noise filtering should be conducted before the simplification. The proposed method can also simplify multiple 3D point clouds. Our future research will concentrate on simplifying multiple 3D point sets simultaneously.

Acknowledgements

We would like to express our sincere thanks to the anonymous reviewers for their thoughtful comments that have improved the quality of the paper and Dr. Reetu Shrestha of Xi'an Jiaotong University for her careful proofreading that has improved the grammar of the paper. We would also like to express our sincere thanks to Dr. Carlos E. Scheidegger for providing us with the executable package of the MLS based method. This study is

supported by the Nature Science Foundation of China Grant No. 50975219.

References

- [1] Varady T, Martin RR, Cox J. Reverse engineering of geometric models—an introduction. Comput-Aided Des 1997;29(4):255–68.
- [2] Shi BQ, Liang J, Liu Q, Xiao ZZ. Precision inspection of point cloud and CAD model based on constraint search sphere. Comput Integr Manuf Syst 2010; 16(5):929–34.
- [3] Lee SJ, Chang DY. A laser sensor with multiple detectors for freeform surface digitization. Int J Adv Manuf Technol 2006;31:474–82.
- [4] Valkenburg RJ, Mcivor AM. Accurate 3D measurement using a structured light system. Image Vis Comput 1998;16:99–110.
- [5] Lee KH, Woo H, Suk T. Data reduction methods for reverse engineering. Int J Adv Manuf Technol 2001;17:735–43.
- [6] Son S, Park H, Lee KH. Automated laser scanning system for reverse engineering and inspection. Int J Mach Tools Manuf 2002;42:889–97.
- [7] Luebke DP. A developer's survey of polygonal simplification algorithms. IEEE Comput Graph Appl 2001;21(3):24–35.
- [8] Cignoni P, Montani C, Scopigno R. A comparison of mesh simplification algorithms. Comput Graph 1998;22(1):37–54.
- [9] Martin RR, Stroud IA, Marshall AD. Data reduction for reverse engineering. RECCAD. Deliverable document 1 COPERNICUS project. No. 1068. Computer and Automation Institute of Hungarian Academy of Science. January 1996.
- [10] Alexa M, Behr J, Cohen-Or D, Fleishman S, Levin D, Silva CT. Computing and rendering point set surfaces. IEEE Trans Vis Comput Graphics 2003;9(1):3–15.
- [11] Scheidegger CE, Fleishman S, Silva CT. Triangulating point set surfaces with bounded error. In: Eurographics symposium on geometry processing. 2005. p. 63–72.
- [12] Moenning C, Dodgson NA. Intrinsic point cloud simplification. In: Proceedings of the 14th international conference on computer graphics and vision. GraphiCon. 2004.
- [13] Pauly M, Gross M, Kobbelt LP. Efficient simplification of point-sampled surfaces. In: Proceedings of the 13th IEEE visualization conference. 2002. p. 163–70.
- [14] Wu J, Kobbelt L. Optimized sub-sampling of point sets for surface splatting. Comput Graph Forum 2004;23(3):643–52.
- [15] Song H, Feng HY. A global clustering approach to point cloud simplification with a specified data reduction ratio. Comput-Aided Des 2008;40(3):281–92.
- [16] Kalaiyah A, Varshey A. Modeling and rendering of points with local geometry. IEEE Trans Vis Comput Graphics 2003;9(1):30–42.
- [17] Linsen L. Point cloud representation. Technical report. Germany: Faculty of Informatics, University of Karlsruhe; 2001.
- [18] Lee KH, Woo H, Suk T. Point data reduction using 3D grids. Int J Adv Manuf Technol 2001;18(3):201–10.
- [19] Miao Y, Pajarola R, Feng J. Curvature-aware adaptive re-sampling for point-sampled geometry. Comput-Aided Des 2009;41(6):395–403.
- [20] Kang EC, Kim DB, Lee KH. Balanced feature-sensitive point sampling for 3D model generation. Int J Adv Manuf Technol 2008;38:130–42.

- [21] Agarwal PK, Procopiuc CM. Exact and approximation algorithms for clustering. In: Proc. ninth ann. ACM-SIAM symp. discrete algorithms. 1998. p. 658–67.
- [22] Alsabti K, Ranka S, Singh V. An efficient k -means clustering algorithm. In: Proc. first workshop high performance data mining. 1998.
- [23] Arora S, Raghavan P, Rao S. Approximation schemes for Euclidean k -median and related problems. In: Proc. 30th ann. ACM symp. theory of computing. 1998. p. 106–13.
- [24] Zhou H, Liu YH. Accurate integration of multi-view range images using k -means clustering. *Pattern Recognit* 2008;41:152–75.
- [25] Cignoni P, Rocchini C, Scopigno R. Metro: measuring error on simplified surfaces. *Comput Graph Forum* 1998;17(2):167–74.
- [26] Geomagic Studio User's Guide. Geomagic studio. 2008.
- [27] Hoppe H, DeRose T, Duchamp T, McDonald J, Stuetzle W. Surface reconstruction from unorganized points. *Comput Graph* 1992;26(2):71–8.
- [28] Mitra NJ, Nguyen A. Estimating surface normals in noisy point cloud data. In: Proceedings of the nineteenth annual symposium on computational geometry. 2003. p. 08–10.
- [29] Hu G, Xu J, Miao L, Peng Q. Bilateral estimation of vertex normal for point-sampled models. In: Proc. computational science and its applications. ICCSA 2005. 2005. p. 758–68.
- [30] OuYang D, Feng HY. On the normal vector estimation for point cloud data from smooth surfaces. *Comput-Aided Des* 2005;37:1071–9.
- [31] Thurmer G, Wuthrich C. Computing vertex normals from polygonal facets. *J Graph Tools* 1998;3(1):43–6.
- [32] Max N. Weights for computing vertex normals from facet normals. *J Graph Tools* 1999;4(2):1–6.
- [33] Klasing K, Althoff D, Wollherr D, Buss M. Comparison of surface normal estimation methods for range sensing applications. In: IEEE international conference on robotics and automation. 2009. p. 12–17.
- [34] Taubin G. A signal processing approach to fair surface design. In: Proceedings of ACM SIGGRAPH 1995. 1995. p. 351–8.
- [35] Vollmer J, Mencl R, Muller H. Improved Laplacian smoothing of noisy surface meshes. *Comput Graph Forum* 1999;18(3):131–8.
- [36] Fleishman S, Drori I, Cohen-Or D. Bilateral mesh denoising. In: Proceedings of SIGGRAPH03. 2003. p. 950–3.
- [37] Desbrun M, Meyer M, Schroder P, Barr A. Implicit fairing of irregular meshes using diffusion and curvature flow. In: Proceedings of SIGGRAPH99. 1999. p. 317–24.
- [38] Hu G, Peng Q, Forrest AR. Mean shift denoising of point-sampled surfaces. *Vis Comput* 2006;22:147–57.
- [39] Zaiane O. Principles of knowledge discovery in databases. Data clustering, lecturing slides for CMPUT, vol. 690. University of Alberta; 1999 [Chapter 8].
- [40] Bentley JL. Multidimensional binary search trees used for associative searching. *Commun ACM* 1975;18:509–17.
- [41] Brodsky D, Watson B. Model simplification through refinement. In: Proceedings of graphics interface 2000. 2000. p. 221–8.
- [42] Horowitz E, Sahni S. Fundamentals of data structure. Pitmen Publishing Limited; 1975.
- [43] <http://graphics.stanford.edu/data/3Dscanrep/>.
- [44] <http://www.cs.caltech.edu/~njlitke/meshes/collections/hhoppe/>.
- [45] <http://www.xjtuom.com>.



Bao-Quan Shi is a Ph.D. student in the School of Mechanical Engineering, Xi'an Jiaotong University in China. He received his B.S. and M.S. degree in mechanical engineering from Xi'an Jiaotong University. His main interests are computer vision, image processing, strain measurement and photogrammetry.



Jin Liang is a professor in the School of Mechanical Engineering, Xi'an Jiaotong University in China. His research interests are computer vision and optical surveying. He obtained his B.S., M.S. and Ph.D. degree from Xi'an Jiaotong University and has been working as a teacher in the university after graduation.



Qing Liu is a resident doctor in the Xi'an Jiaotong University Stomatology Hospital. She received her B.S. and M.S. degree from Xi'an Jiaotong University and is working on medical image processing.