

APPROXIMATE INTRINSIC VOXEL STRUCTURES FOR POINT CLOUD PROCESSING

PRIORITY CLAIM

[0001] This application claims priority from Singapore Patent Application No. 10202011637Y filed on 23 November 2020.

TECHNICAL FIELD

[0002] The present invention generally relates to voxel structure point cloud processing, and more particularly relates to methods and systems for approximate intrinsic voxel structures for point cloud processing, including resampling, simplification and mesh reconstruction.

BACKGROUND OF THE DISCLOSURE

[0003] With the rapid development of 3D scanning technology, using 3D point clouds to represent real-world objects is becoming increasingly popular. A point cloud comprises a set of points that carry geometry and possible attribute information such as point position, color, and texture. The point cloud data structure is versatile and flexible in creating and representing 3D models and thus is being widely adopted in a wide range of applications including 3D object recognition, scene reconstruction, industrial modelling.

[0004] Although point cloud data structures exhibit unique advantages, there have been several challenges in practice. First, to accurately represent 3D objects, point clouds are generally made up of high-density points. As a result, point clouds can be huge in terms of data volume, which could significantly increase the cost of storage and transmission. Secondly, a dense point cloud includes a fair amount of redundant

information. This information is usually not only dispensable in many cases, but greatly unfavorable for efficiently performing specific tasks such as visualization, geometric feature training, and localization. Finally, limited by the accuracy of scanning devices, the density distributions of raw point clouds are non-uniform thereby affecting the utilization of point clouds.

[0005] To tackle the aforementioned challenges, some point cloud processing methods have been proposed, including point cloud compression, point cloud simplification and point cloud resampling. Although these three approaches share similar ultimate goals, their mechanisms and application scenarios are quite different. Point cloud compression aims to reduce the storage of a point cloud by exploiting the spatial correlation among adjacent points. However, this approach does not lead to a change of the number of points of the original point cloud in most cases. Even in some lossy compressions, the number of points will be changed, but the range should be controlled.

[0006] Point cloud simplification reduces the information redundancy of a given point cloud by removing an amount of non-essential points without significantly affecting the performance of target tasks by changing the point cloud into a new one with fewer points. Point cloud resampling obtains a new resampling point cloud from the original one while keeping the geometric consistency. The resampling point number can be controlled to be larger or smaller than the original point cloud and a point cloud with non-uniform density can be adjusted into a uniform point cloud.

[0007] A number of effective point cloud resampling schemes have been proposed and they can be broadly divided into two categories based on different targets. A first category of resampling schemes focuses on retaining important local regions in a point cloud and may be referred to as local resampling. Local resampling methods are

proposed to improve the effectiveness and retain the local geometric features such as DSO feature, sharp-feature keeping, Gaussian curvature, and saliency detection. The resampling results may be achieved by summing up the results from different subsets. The local resampling methods can provide a more flexible framework for geometric feature keeping with lower computational cost. While local resampling may retain local regions such as keypoints and visually sensitive areas, other geometric features of the point cloud may be lost and some global features such as intrinsic and isotropic properties may be lost. In addition, for extremely anisotropic regions in a point cloud, the methods without density adjusting will produce error resampling results with a high probability.

[0008] A second category of resampling schemes constructs a resampling result from an original point cloud, while keeping the quality of both a simplified point cloud and important geometric features. This second category may be referred to as global resampling and effective and efficient global resampling should satisfy four requirements: correct local region, uniform density, point number control, and geometric feature keeping.

[0009] According to moving least-square (MLS) surface theory, the surface of a 3D object can be combined by local regions, which are represented by neighbor relationships of points in a point cloud. The correct local regions improve the resampling quality and, based on the correct local regions, the point density determines the accuracy of MLS surface representation. A uniform resampling result will reduce MLS fitting errors and the point cloud number should be controlled by user-specification. MLS supports some applications which require the same point number of point clouds. Global resampling methods are constructed using different global resampling strategies. Classical methods include geometric error optimization,

Voronoi diagram, global clustering, and Laplace graphic. Although more accurate resampling can be achieved, these methods require high computational complexity and are generally incapable of providing flexible local resampling solutions.

[0010] The state of art resampling methods include Centroidal Voronoi Tessellation (CVT), Laplace graphic and particle-based resampling. Such methods resample the raw point cloud into an isotropic one with certain point number control in local tangent space. Such resampling methods also support point cloud-based mesh reconstruction. However, such methods smooth the raw point cloud in a local tangent space and some geometric features will be lost. The point positions are changed by such methods which cannot be used in point cloud simplification with strict point position restriction. Furthermore, Laplace graphic and Centroidal Voronoi Tessellation (CVT) cannot satisfy the four requirements for effective and efficient global resampling at the same time.

[0011] Thus, there is a need for methods for point cloud resampling which overcomes the drawbacks of conventional point cloud resampling methods and provide correct local region, uniform density, point number control, and geometric feature keeping. Furthermore, other desirable features and characteristics will become apparent from the subsequent detailed description and the appended claims, taken in conjunction with the accompanying drawings and this background of the disclosure.

SUMMARY

[0012] According to at least one aspect of the present embodiments, a method for voxel-structure-based point cloud resampling is provided. The method includes dividing a point cloud into a plurality of voxel boxes with adjacency, each of the plurality of voxel boxes defining a local region in a three-dimensional space. The

method also includes point cloud resampling the plurality of points based on an approximate intrinsic voxel structure to generate a resampled point cloud, wherein the point cloud resampling is implemented differently in at least two of the plurality of voxel boxes.

[0013] According to other aspects of the present embodiments, a method for point cloud simplification and a method for voxel-structure-based mesh reconstruction are provided which also include the dividing the point cloud into a plurality of voxel boxes with adjacency and point cloud resampling the plurality of points based on an approximate intrinsic voxel structure of the method for voxel-structure-based point cloud resampling.

BRIEF DESCRIPTION OF THE DRAWINGS

[0014] The accompanying figures, where like reference numerals refer to identical or functionally similar elements throughout the separate views and which together with the detailed description below are incorporated in and form part of the specification, serve to illustrate various embodiments and to explain various principles and advantages in accordance with present embodiments.

[0015] FIG. 1 depicts an illustration of a pipeline for point cloud processing including AIVS-based point cloud resampling in accordance with the present embodiments.

[0016] FIG. 2 depicts an illustration of a smoothing process for a 2D point cloud with noise in accordance with the present embodiment.

[0017] FIG. 3, comprising FIGs. 3A, 3B and 3C, depicts illustrations of point cloud down-sampling by octree in accordance with the present embodiments, wherein FIG. 3A depicts octree down-sampling, FIG. 3B depicts an original point cloud rendering of

an image of a statue, and FIG. 3C depicts an octree down-sampled point cloud rendering of the image of the statue of FIG. 3B.

[0018] FIG. 4 depicts an illustration of up-sampling processing based on Delaunay interpolation in accordance with the present embodiments.

[0019] FIG. 5, comprising FIGs. 5A and 5B, depicts an illustrations of using Euclidean distance to approximate geodesic distance between points in a voxel box in accordance with present embodiments, wherein FIG. 5A depicts a 3D voxel box with points geodesically arranged and FIG. 5B depicts approximating the geodesic distances between points using Euclidean distances.

[0020] FIG. 6, comprising FIGs. 6A, 6B and 6C, depicts an instance of point resampling in a voxel box in accordance with the present embodiments, wherein FIG. 6A depicts the plane of points in the voxel box used for Euclidean distance approximation, FIG. 6B depicts a Voronoi diagram of a portion of the plane of FIG. 6A partitioned into regions close to each point in the voxel box, and FIG. 6C depicts a magnification of a portion of the Voronoi diagram of FIG. 6B.

[0021] FIG. 7 depicts an illustration of approximate intrinsic control of AIVS in accordance with the present embodiments.

[0022] FIG. 8, comprising FIGs. 8A, 8B and 8C, depicts illustrations of an example of resampling in different parallel computation steps in accordance with the present embodiments, wherein FIG. 8A depicts a 3D representation of voxel boxes and FIGs. 8B and 8C depict parallel computing steps Pa1 and Pa2.

[0023] FIG. 9, comprising FIGs. 9A and 9B, depicts an illustration of an exemplary isotropic resampling in accordance with the present embodiments, wherein FIG. 9A depicts non-optimized voxel box point distances and FIG. 9B depicts optimization by isotropic resampling.

[0024] FIG. 10 depicts an illustration of an example of a point cropping process in accordance with the present embodiments.

[0025] FIG. 11 depicts an illustration of an example of sharp-feature keeping AIVS-based flexible resampling in accordance with the present embodiments.

[0026] FIG. 12, comprising FIGs. 12A, 12B and 12C, depicts an example of reconstructed mesh with different user-specified point numbers in accordance with the present embodiments, wherein FIG. 12A depicts a reconstructed mesh with a 5,000 point number, FIG. 12B depicts a reconstructed mesh with a 10,000 point number, and FIG. 12A depicts a reconstructed mesh with a 20,000 point number.

[0027] FIG. 13, comprising FIGs. 13A and 13B, depicts an illustration of internal boundary cropping in accordance with the present embodiments, wherein FIG. 13A depicts illegal triangles (i.e., triangles crossing internal boundaries) and FIG. 13B depicts removal of the illegal triangles.

[0028] FIG. 14, comprising FIGs. 14A and 14B, depicts a first example of mesh optimization in accordance with the present embodiments, wherein FIG. 14A depicts an entire point cloud and FIG. 14B depicts a magnification of a portion of the point cloud of FIG. 14A showing sharp feature keeping.

[0029] FIG. 15, comprising FIGs. 15A and 15B, depicts a second example of mesh optimization in accordance with the present embodiments, wherein FIG. 15A depicts an entire point cloud and FIG. 15B depicts a magnification of a portion of the point cloud of FIG. 15A showing sharp feature keeping.

[0030] FIG. 16 depicts an illustration comparing simplification results from different methods including AIVS-based simplification in accordance with the present embodiments for a 10,000 simplification point model.

[0031] FIG. 17 depicts an illustration comparing sharp-feature keeping simplification results from different methods including AIVS-based simplification in accordance with the present embodiments for both a 5,000 simplification point model (upper row) and a 10,000 simplification point model (lower row).

[0032] FIG. 18 depicts a visual comparison of mesh reconstruction results by different methods for three different models, each having a 10,000 point number.

[0033] FIG. 19, comprising FIGs. 19A and 19B, depicts minimum angle statistic histograms of mesh reconstruction of two models using Scale Space-based mesh reconstruction optimization, wherein FIG. 19A depicts a minimum angle statistic histogram for a first model and FIG. 19B depicts a minimum angle statistic histogram for a second model.

[0034] FIG. 20, comprising FIGs. 20A and 20B, depicts minimum angle statistic histograms of mesh reconstruction of two models using Screened Poisson-based mesh reconstruction optimization, wherein FIG. 20A depicts a minimum angle statistic histogram for a first model and FIG. 20B depicts a minimum angle statistic histogram for a second model.

[0035] FIG. 21, comprising FIGs. 21A and 21B, depicts minimum angle statistic histograms of mesh reconstruction of two models using Delaunay Strategy-based mesh reconstruction optimization, wherein FIG. 21A depicts a minimum angle statistic histogram for a first model and FIG. 21B depicts a minimum angle statistic histogram for a second model.

[0036] FIG. 22, comprising FIGs. 22A and 22B, depicts minimum angle statistic histograms of mesh reconstruction of two models using Parallel CVT-based mesh reconstruction optimization, wherein FIG. 22A depicts a minimum angle statistic

histogram for a first model and FIG. 22B depicts a minimum angle statistic histogram for a second model.

[0037] FIG. 23, comprising FIGs. 23A and 23B, depicts minimum angle statistic histograms of mesh reconstruction of two models using Gaussian kernel-based mesh reconstruction optimization, wherein FIG. 23A depicts a minimum angle statistic histogram for a first model and FIG. 23B depicts a minimum angle statistic histogram for a second model.

[0038] FIG. 24, comprising FIGs. 24A and 24B, depicts minimum angle statistic histograms of mesh reconstruction of two models using AIVS-based mesh reconstruction optimization in accordance with the present embodiments, wherein FIG. 24A depicts a minimum angle statistic histogram for a first model and FIG. 24B depicts a minimum angle statistic histogram for a second model.

[0039] And FIG. 25, comprising FIGs. 25A and 25B, depict graphs of computation time for conventional resampling methods as compared to the AIVS-based resampling method in accordance with the present embodiments, wherein FIG. 25A depicts a graph of computation time for 10,000 simplification points and FIG. 25B depicts computation time for 0.1 simplification rate.

[0040] Skilled artisans will appreciate that elements in the figures are illustrated for simplicity and clarity and have not necessarily been depicted to scale.

DETAILED DESCRIPTION

[0041] The following detailed description is merely exemplary in nature and is not intended to limit the invention or the application and uses of the invention. Furthermore, there is no intention to be bound by any theory presented in the preceding background of the invention or the following detailed description. It is the intent of present

embodiments to present an Approximate Intrinsic Voxel Structure (AIVS) which enables an efficient point cloud processing pipeline combining resampling, simplification and mesh reconstruction.

[0042] Specifically, the present embodiments include a point cloud resampling method that samples the raw point cloud into an isotropic one or an anisotropic one with certain point number and intrinsic control. The intrinsic control means that the local region in a point cloud is defined by the geodesic distance, thereby improving the accuracy of local region detection in a point cloud. The isotropic property means that the distances between each point to its first-ring neighbors are approximately equal in a global view, guaranteeing that the point cloud after resampling is a global uniform density one.

[0043] For example, in accordance with the present embodiment, the point cloud can be divided into a plurality of voxel boxes based on adjacency, where the term “adjacency” refers to whether the boxes can be judged as adjacent or not. Thus, as a rule for point selection in simplification the approximate intrinsic voxel structure may be constructed based on geodesic distance where “intrinsic” may be regarded as geodesic distance-based.

[0044] For anisotropic resampling, the local uniform density can be kept while simultaneously keeping the important geometric features. Based on the methods in accordance with the present embodiments, two applications are proposed: point cloud simplification and point cloud-based mesh reconstruction. While point cloud simplification is not new, using the resampling method in accordance with the present embodiments to simplify a point cloud can meet all requirements of conventional point cloud simplification.

[0045] Point cloud-based mesh reconstruction constructs a polygon mesh from an input point cloud. The quality of the reconstructed mesh depends on local region detection and isotropic property. Benefiting from the advantages of the resampling method in accordance with the present embodiments, both point cloud simplification and point cloud-based mesh reconstruction can be implemented with high precision.

[0046] Point cloud resampling in accordance with the present embodiments is based on Approximate Intrinsic Voxel Structure (AIVS), which is an efficient structure for point cloud organization and enables resampling to be implemented in different voxel boxes independently. AIVS allows independent resampling from different voxel boxes while providing intrinsic control with a certain point number facilitating resampling of a raw point cloud into an isotropic one. The certain point number can be specified by a user without any range limitation. The efficiently flexible resampling scheme based on AIVS in accordance with the present embodiments classifies the points in the point cloud into different categories according to a user-specified classification strategy. With different resampling rates for the different categories, important geometric features such as sharp-edges can be maintained.

[0047] Parallel computation for AIVS-based resampling is also enabled in accordance with the present embodiments, thereby accelerating the speed of resampling while keeping geometric consistency with the original point cloud.

[0048] AIVS satisfies all four requirements for resampling tasks, namely correct local region, uniform density, point number control, and geometric feature keeping, and can achieve isotropic property in the resampled point cloud. AIVS also provides a more flexible scheme to meet user-specified anisotropic resampling, such as sharp-feature keeping, while keeping a locally uniform density. Based on the functions of AIVS, it can be used to implement two applications in accordance with the present

embodiments: point cloud simplification and point cloud-based mesh reconstruction, which can be implemented with high precision.

[0049] Compared to conventional resampling methods, point cloud processing methods in accordance with the present embodiments provide a balance between global and local resampling schemes while avoiding the drawbacks of both resampling schemes. The methods in accordance with the present embodiments obtain an isotropic result with intrinsic control. They also provide an anisotropic resampling for geometric features keeping with local uniform density. The methods in accordance with the present embodiments can advantageously be accelerated by parallel computing and support applications such as simplification and mesh reconstruction.

[0050] Referring to FIG. 1, an illustration 100 depicts a pipeline for point cloud processing in accordance with present embodiments. The pipeline includes inputting a raw point cloud 110, AIVS-based resampling 120 in accordance with the present embodiments, and then either simplification 150 of the resampled point cloud or point cloud-based mesh reconstruction 160 of the resampled point cloud.

[0051] The raw point cloud 110 is composed of a plurality of points in a three-dimensional (3D) space and the AIVS-based point cloud resampling 120 is mainly composed of three components: point cloud pre-processing 130, AIVS construction 140 and AIVS-based resampling 145. The point cloud pre-processing 130 removes noise by denoising 132 the point cloud and then adjusts the non-uniform density from the denoised input point cloud by, for example, octree down-sampling 134 then up-sampling 136 the denoised point cloud.

[0052] The AIVS construction 140 provides the intrinsic control that improves the accuracy of the point distance measurement in accordance with the present embodiments by dividing the resampling task into sub-tasks in different voxel boxes.

The AIVS-based resampling 145 includes anisotropic or isotropic resampling by computing the resampling rates of points in different voxel boxes in accordance with user-specified parameters, where anisotropic resampling would include classifying the points into different categories in accordance with the user-specified category criteria before computing the resampling rates of the points in the different voxel boxes. The AIVS-based resampling 145 may include resampling the point cloud by parallel computation to accelerate the resampling. The AIVS-based resampling 145 further includes deleting the redundant points to fit an accurate point number specified by the user.

[0053] The two post-resampling applications include point cloud simplification 150 and point cloud-based mesh reconstruction 160. For the simplification 150, the pipeline is same as the resampling. The difference is that the points are selected from the original point cloud without new points inserted and position changes. For the mesh reconstruction 160, the pipeline is inputting the point cloud after AIVS-based resampling 145, constructing the mesh from the resampling 145 result by Delaunay triangulation, and optimizing the mesh.

Point Cloud Pre-Processing 130

[0054] The raw point clouds from 3D scanning devices cannot be used directly without first pre-processing 130. The pre-processing 130 includes denoising 132 and uniform density adjusting. For denoising 132, the theory of moving least-square (MLS) surface representation is utilized to smooth the point cloud 110 as formulated in Equation (1).

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

[0055] The denoising 132 is a "pulling back" process for points which are outliers or noise. The weight is determined by the distance between a target point p_i and its adjacent point p_j . The point set $\{p_j\}$ includes all adjacent points of p_i . The result of the smoothing process provides an updated direction Δp_i to smooth point p_i based on $\{p_j\}$. The weight function θ determines different point weights in $\{p_j\}$ and is fit into a Gaussian function in accordance with Equation (2).

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

[0056] Referring to FIG. 2, an illustration 200 depicts a smoothing process for a 2D point cloud with noise in accordance with the present embodiment. The smoothing process takes a point cloud 210, calculates 220 an MLS surface 225, and "pulls back" 230 outliers (e.g., points 232, 234) in accordance with Equations (1) and (2) to create a denoised point cloud result 240.

[0057] To adjust the density of a point cloud, we carry out a down-sampling process using octree down-sampling. Octree down-sampling is widely used in point cloud compression. Distances between points in an octree are controlled by a voxel scale used as a constraint. Following the constraint, a point cloud with non-uniform regions is transferred into an approximately uniform point cloud. For some point clouds with high density, the octree-based down-sampling reduces their scale, thereby improving the efficiency of the whole pipeline. FIG. 3A depicts an illustration 300 of point cloud down-sampling by octree where an original point cloud 305 is processed in accordance with octree down-sampling to result in the down-sampled point cloud 310. Note that the more dense upper right portion 315 of the original point cloud 310 is divided into smaller voxel boxes during octree down-sampling than the remainder of the point cloud.

[0058] FIGs. 3B and 3C depict point cloud images 330, 360 of a statue 332, 362, respectively, where a front view 334 and bottom view 336 depict an original point cloud

rendering of the statue 332 and a front view 364 and bottom view 366 depict an octree down-sampled point cloud rendering of the statue 362.

[0059] In practice, the voxel scale is estimated based on a user-specified point number and the point number of the original point cloud. In Equation (3), a recommendation for octree voxel scale O_s is provided. The voxel scale O_s is the maximum value of d_k , which is the longest distance of point p_i to its k neighbor points. The value k is estimated by point number $|P|$ and resampling point number $|P_s|$. Parameter m is used to control the down-sampling rate, where $m \leq |P|/|P_s|$. Considering the candidate space for resampling, if $|P|/|P_s| > 5$, the default value is $m = 5$, otherwise $|P|/|P_s| \leq 5$ ignores m and $k = 1$.

$$O_s = \max\{d_k(p_i, p_j), j \leq k\}, k = \frac{|P|}{m * |P_s|} \quad (3)$$

[0060] Using the octree-based down-sampling 134, the point number of a point cloud will be reduced and the resampling number $|P_s|$ should be smaller than $|P|$. It can be regarded as a raw simplification. For resampling, $|P_s|$ should not be limited. In some cases (sparse point clouds or user-specified point number being similar to or larger than a given point cloud), it is desirable to keep or increase the point number while adjusting the point cloud density. Therefore, an up-sampling process is needed before the octree-based down-sampling and density adjustment. In accordance with the present embodiments, an up-sampling processing 136 (FIG. 1) based on Delaunay interpolation is proposed. FIG. 4 depicts an illustration 400 of up-sampling processing based on Delaunay interpolation. Delaunay interpolation builds a Delaunay triangulation region 410 for each point and, as shown in the image 420, inserts new points into the region 410. The black points 430 represent up-sampling points ($s=6$) added by the up-sampling and inserted into the triangle's face 450. The points 430 are proportional to the triangle's area. The light grey points 440 on a single edge ($s=3$) are computed by

s as shown in Equation (4). After the pre-processing 130 (FIG. 1), the point cloud can be used for next steps.

$$s_l = \left\lceil \sqrt{2s + \frac{1}{2}} - \frac{\sqrt{2}}{2} \right\rceil \quad (4)$$

Approximate Intrinsic Voxel Structure (AIVS) Construction 140

[0061] The Approximate Intrinsic Voxel Structure (AIVS) in accordance with the present embodiments combines global and local resampling strategies with intrinsic control. AIVS is a structure similar to voxelization and octree. Points of a given point cloud are divided in accordance with the present embodiments into different voxel boxes of AIVS and resampled independently. The “approximate intrinsic” in AIVS means that a Euclidean distance is used to approximate geodesic distance in a voxel box as shown in illustrations 500, 550 of FIGs. 5A and 5B. The voxel box limits curvature changes between points, which is regarded as an intrinsic control. In the illustration 550, the Euclidean distance between points is used to approximate the geodesic distance between the points in a voxel box 510 in the illustration 500.

[0062] A farthest point sampling (FPS) is used to resample the points in the voxel boxes independently to provide an intrinsic resampling result for the voxel boxes. Referring to FIGs. 6A, 6B and 6C, illustrations 600, 630, 660 depict an instance of point resampling in a voxel box in accordance with the present embodiments. The illustration 600 depicts the plane of points in the voxel box used for Euclidean distance approximation (e.g., as shown in the illustration 550). The illustration 630 depicts a local resampling result in a Voronoi diagram representation where the resampling partitions the plane into regions close to each point in the voxel box. The illustration 660 is a simplification result of the Voronoi diagram 630 showing that the resampling is an isotropic resampling .

[0063] The intrinsic control is provided based on voxel boxes. Referring to FIG. 7, an illustration 700 depicts intrinsic control of AIVS in accordance with the present embodiments. The point cloud 710 depicts modelling a 3D image in accordance with AVIS. A portion of this 3D image is depicted in a 2D arrangement 720 of voxel boxes. The distances between different points are controlled inside the voxel boxes. Intrinsic control 730 of AIVS in the 2D boxes is shown in illustrations 735, 740. The voxel boxes control the adjacent points in the local space (i.e., the black dotted squares in the illustration 740). The wrong adjacent points are removed from different voxel boxes. As the process of independent processing of the voxel boxes is computationally demanding, the AIVS processing can be accelerated in accordance with the present embodiments by parallel computation. For example, each voxel box may correspond to a parallel computing unit. Considering the adjacent limit points selection, the points from adjacent voxel boxes are not sampled in a parallel computing step as the parallel computing for resampling in one step is arranged to voxel boxes which are not adjacent.

[0064] Referring to FIGs. 8A, 8B and 8C, illustrations 800, 830, 860 depict an example of resampling in different parallel computation steps in accordance with the present embodiments. The illustration 800 depicts a 3D representation of voxel boxes where the box 810 is being simplified. In the illustration 830, the different parallel computing steps Pa1 835 and Pa2 840 are arranged for different voxel boxes and each parallel computing step 835, 840 has different processing threads as shown in the illustration 860. Once the resampling point number in each voxel box is determined, the AIVS-based resampling result is achieved. The point number is then proportional to the corresponding MLS surface area. The resampling rate R_v in each voxel box is obtained by ratio $|P_s|/|P|$ and the resampling point number $|P_{sv}|$ in a voxel box is

calculated by $|P_{sv}| = |P_v| \times R_v$, where $|P_v|$ is the point number of a voxel box before resampling.

Resampling in AIVS

[0065] Resampling can be classified into two categories: isotropic resampling and anisotropic resampling. Isotropic resampling means that the distances from a point to each of its neighbors is approximately equal. Isotropic resampling achieves global uniform density results and the weights of points are equal. Anisotropic resampling is an opposite operation where the point density in different regions and point weights are not same. Anisotropic resampling is used to satisfy special requirements such as curvature-sensitive sampling and sharp feature preservation.

[0066] For isotropic resampling, the local resampling rates are equal in AIVS. The resampling results can be achieved from a combination of local resampling results in different voxel boxes. From local to global, it should be noticed that the points' distances are not optimized in the boundary of adjacent boxes and a non-uniform point density may occur near the boundary of the voxel boxes. Referring to FIGs. 9A and 9B, illustrations 900, 950 depict an exemplary isotropic resampling in accordance with the present embodiments. In the two voxel boxes 910, 920 in the illustration 900, the distances between the points in the two dotted grey boxes 925 are not optimized--the distances are too small. To solve the problem, points 955 that have been sampled from the adjacent voxel boxes have been added to limit the non-uniform sampling and achieve isotropic resampling. The added points 955 are called adjacent limit points and the FPS considers the adjacent limit points 955 around the boundary of the voxel box 920 when resampling is processed in the voxel box 920. Thus, non-uniform sampling can advantageously be avoided in the boundary of adjacent boxes and isotropic

resampling is achieved by adjacent limit points near the voxel boxes' boundary in accordance with the present embodiments.

[0067] Combining the local resampling points from different voxel boxes, a final resampling result is obtained. The resampling point number from voxel boxes may be larger than the accurate resampling point number specified by the user (i.e., the user-specified accurate resampling point number). The reason is the resampling rate R_v for each box produces a remainder. An accurate point cropping process is therefore required. Following the isotropic property, a point should be removed which has a closer distance to its adjacent points than other points. In an illustration 1000 of FIG. 10, P_1 and P_2 are adjacent points as shown in the point diagram 1010. The distance (sum of dotted lines 1035) of adjacent points to P_2 in the point diagram 1030 is smaller than the distance (sum of dotted lines 1025) of adjacent points P_1 in the point diagram 1020. Thus, as compared to P_1 , the point P_2 satisfies the conditions of redundant point. Accordingly, the point P_2 is removed as shown in the point diagram 1040 by the point cropping process in accordance with present embodiments.

[0068] The cropping process in accordance with present embodiments deletes the redundant points iteratively until the point number is equal to the user-specified resampling point number. In Equation (5), the point cropping process in accordance with present embodiments is shown for deleting redundant points. Point p_j is the adjacent point of p_i . The adjacent point set of p_i is $R(p_i)$ and, in the exemplary cropping process shown in the illustration 1000, $|R(p_i)| = 2$.

$$p_d = \{p_i | \min \sum_{p_j \in R(p_i)} \|p_i, p_j\|^2\} \quad (5)$$

[0069] For anisotropic resampling, a flexible resampling scheme is proposed in accordance with the present embodiments. The density of the resampling points in a voxel box is decided by the resampling rate R_v . In Equation (6), a flexible resampling

scheme in accordance with the present embodiments is shown. The points are classified into c categories and the number of points which belong to category c_i is $|P_i|$. For each category, a user-specified rate a_i is used to control the resampling rate $|R_{vi}|$. Following the different resampling rates, a flexible resampling result is achieved. Considering that point numbers from different categories are not equal, the resampling rates $\{|R_{vi}|\}$ should be limited in a range ($R_{vi} \in [0.1, 1]$) to avoid sampling errors. In addition, the resampling number should be smaller than existing points in the voxel box and larger than zero. Based on the limitation of R_{vi} , the different rates of a_i are specified by users.

$$P_s = \sum_{i=1}^c R_{vi} \times P_i, R_{vi} = \beta \times a_i$$

$$\beta = \frac{|P_s|}{\sum_{i=1}^k a_i \times |P_i|} \quad (6)$$

[0070] Based on Equation (6), a type of anisotropic resampling for sharp-feature keeping is introduced. The points are classified into two categories: edge point set P_b and ordinary point set P_{nb} . An edge point means a corresponding MLS area of the point crosses an edge region and edge regions typically include sharp curvature transfer. Edge points can be detected using a Voronoi covariance measure (VCM). Once the resampling rate for edge points is larger than the rate for ordinary points, the edges in a point cloud can be labeled in visualization. Referring to FIG. 11, an illustration 1100 depicts an example of sharp-feature keeping AIVS-based flexible resampling in accordance with the present embodiments. In the illustration 1100, edges 1105a, 1105b are shown in a Fandisk point cloud 1110, the points at the edges 1105a, 1105b being shown in greater detail in the magnified portion view 1150. In summary, AIVS provides a general resampling framework for isotropic and anisotropic resampling without complex computation.

Resampling-based Applications

[0071] As shown in the illustration 100 (FIG. 1), the resampling method in accordance with the present embodiments supports two applications: point cloud simplification 150 and point cloud-based mesh reconstruction 160. For point cloud simplification 150, the resampling method in accordance with the present embodiments as described hereinabove can be used directly to simplify a point cloud by remove the up-sampling step 136 from the pre-processing 130 to achieve the simplification result. The points from the simplification result are selected from the original point cloud 110 without position change. Point cloud simplification is an approximately isotropic process. For special simplification tasks, the resampling method in accordance with the present embodiments also provides anisotropic resampling functions as discussed hereinabove.

[0072] For point cloud-based mesh reconstruction 160, the resampling method provides the input point cloud 110 with correct local regions and uniform density. The point cloud-based mesh reconstruction in accordance with the present embodiments is regarded as a distance field optimization which improves the accuracy of mesh reconstruction. Based on the resampling result, a Delaunay triangulation method is used to reconstruct mesh. Since the distance field is optimized with intrinsic control, the reconstructed mesh is more accurate. FIGs. 12A, 12B and 12C, depict illustrations 1200, 1210, 1220 of reconstructed mesh with different user-specified point numbers in accordance with the present embodiments. The illustration 1200 depicts a reconstructed mesh with a 5,000 point number, the illustration 1210 depicts a reconstructed mesh with a 10,000 point number, and the illustration 1220 depicts a reconstructed mesh with a 20,000 point number.

[0073] To further improve the quality of the reconstructed mesh, a mesh optimization process is provided in accordance with the present embodiments which is composed of internal boundary rebuilding and isotropic remeshing. The internal boundary is an

important geometric feature. Benefiting from the intrinsic control in AIVS, the internal boundary can be rebuilt from the reconstructed mesh. Triangles that cross regions which are not adjacent are illegal, meaning that regions covered by illegal triangles are inside an internal boundary. The internal boundary can be rebuilt by deleting the illegal triangles. In the illustrations 1300, 1350 of FIGs. 13A and 13B, internal boundary cropping in accordance with the present embodiments is depicted. In the illustration 1300, illegal triangles 1310a, 1310b, 1310c, 1310d cross internal boundaries. In accordance with the internal boundary cropping process, the illegal triangles 1310a, 1310b, 1310c, 1310d are removed as shown in the illustration 1350 to rebuild the internal boundaries represented by the dashed line 1360.

[0074] As is well-known to those skilled in the art, an isotropic mesh is better than a mesh without isotropic property. Thus, in accordance with the present embodiments, an isotropic remeshing method is proposed to transfer the reconstructed mesh into an isotropic one. Basically, isotropic remeshing method in accordance with the present embodiments is based on a conventional isotropic remeshing method with improvements to address obtuse angles and a certain point number control for sharp-feature keeping. The remeshing steps are summarized as: (a) compute an average triangle edge length l of the mesh, (b) split the edge if its length is larger than $\frac{4}{3}l$, (c) count the split number i_s , (d) collapse the edge into the middle point if its length is smaller than $\frac{4}{5}l$ unless the collapse edge includes a sharp edge point with more than two sharp edge neighbors, (e) stop if the collapse number is larger than i_s , and (f) if the collapse number is not larger than i_s , then flip edges to optimize the valence unless the flip generates obtuse angles or the flip edge includes two sharp edge points followed by smoothing points in tangent space.

[0075] The details of typical valence optimization and tangent space smoothing are known to those skilled in the art. Based on the mesh optimization, the mesh is transferred into an isotropic mesh with a certain point number wherein the sharp edges are kept in the mesh. A first example of mesh optimization is shown in the illustrations 1400, 1450 of FIGs. 14A and 14B and a second example of mesh optimization is shown in the illustrations 1500, 1550 of FIGs. 15A and 15B. The point cloud modelling of a rabbit in the illustration 1400 shows that mesh optimization preserves the rounded edges 1410, 1420 which is better seen in the magnification view in the illustration 1450. Similarly, the point cloud modelling in the illustration 1500 shows that mesh optimization preserves the sharp edges 1510, 1520 which is better seen in the magnification view in the illustration 1550.

[0076] The resampling performance by simplification and mesh reconstruction in accordance with the present embodiments is evaluated by comparing the resampling performance to resampling performance of conventional methodologies. For the evaluation of simplification, geometric fitting errors are computed for the AIVS method in accordance with the present embodiments and is compared with geometric fitting errors of conventional simplification methods, including MLS error optimization, Hierarchical simplification, Wlop, and Laplace graphic. TABLES I and II compare the geometric error analysis results for the different methods as applied to different point cloud models, where TABLE I computes geometric errors for models having 10,000 simplification points and TABLE II computes geometric errors for the models at a 0.1 simplification rate. The models are Stanford Models including a bunny model (e.g., FIG. 14A), a horse model, an armadillo model, an angel model (e.g., FIGs. 12A, 12B and 12C), a dragon model, a buddha model (e.g., FIGs. 3B and 3C), and an Asian dragon model.

Methods	MLS Error		Hierarchy		Wlop		Laplace graph		AIVS	
Models	Δ Max	Δ Avg	Δ Max	Δ Avg	Δ Max	Δ Avg	Δ Max	Δ Avg	Δ Max	Δ Avg
Bunny	0.02541	0.00329	0.06283	0.00319	0.02761	0.00298	0.02483	0.00297	0.02082	0.00258
Horse	0.03356	0.00221	0.03207	0.00175	0.03513	0.00179	0.01704	0.00184	0.01780	0.00164
Armadillo	0.09539	0.00508	0.04834	0.00505	0.02702	0.00337	0.02072	0.00330	0.02401	0.00329
Angel	0.05130	0.00474	0.05674	0.00448	0.04446	0.00252	0.02345	0.00243	0.01605	0.00244
Dragon	0.07126	0.00539	0.07059	0.00848	0.03178	0.00361	0.02502	0.00346	0.02499	0.00344
Buddha	0.09334	0.00725	0.00828	0.00740	0.07465	0.00415	0.02439	0.00391	0.02198	0.00381
Asian Dragon	-----	-----	0.02083	0.00326	0.03270	0.00302	-----	-----	0.02597	0.00289

TABLE I
Geometric Errors for the Standford Models by The Different Methods
(10,000 Simplification Points)

Methods	MLS Error		Hierarchy		Wlop		Laplace graph		AIVS	
Models	Δ Max	Δ Avg	Δ Max	Δ Avg	Δ Max	Δ Avg	Δ Max	Δ Avg	Δ Max	Δ Avg
Bunny	0.09732	0.00815	0.07272	0.008858	0.04204	0.00646	0.03703	0.00611	0.03665	0.00603
Horse	0.07611	0.00480	0.03582	0.00497	0.03223	0.00302	0.02559	0.00313	0.02242	0.00291
Armadillo	0.07491	0.00355	0.03502	0.0032	0.02234	0.00245	0.01607	0.00231	0.01439	0.00229
Angel	0.05621	0.00233	0.05135	0.002401	0.04329	0.00134	0.02602	0.00136	0.01330	0.00129
Dragon	0.02759	0.00173	0.03492	0.00181	0.02455	0.00123	0.01779	0.00134	0.00981	0.00114
Buddha	0.04376	0.00184	0.06175	0.00272	0.03371	0.00124	0.02631	0.00129	0.01357	0.00121
Asian Dragon	-----	-----	0.01221	0.00065	0.00725	0.00026	-----	-----	0.00564	0.00027

TABLE II
Geometric Errors for the Standford Models by The Different Methods
(0.1 Simplification Rate)

[0077] The results show that the AIVS-based simplification method has good performance for the MLS surface fitting as compared to the conventional methods. Referring to FIG. 16, an illustration 1600 shows a comparison of simplification results of the armadillo model for the MLS error optimization method, the hierarchical simplification method, the Wlop method, the Laplace graphic method and the AIVS-based simplification method in accordance with the present embodiments. Similarly, an illustration 1700 in FIG. 17 shows a comparison of sharp-feature keeping simplification results of the Fandisk model for the MLS error optimization method, the hierarchical simplification method, the Wlop method, the Laplace graphic method and the AIVS-based simplification method in accordance with the present embodiments, where an upper row 1710 includes point clouds having 5,000 simplification points and a lower row 1720 includes point clouds having 10,000 simplification points. It can be seen from the illustration 1600 that the simplification results by the AIVS-based

resampling method in accordance with the present embodiments achieved better isotropic properties and, from the illustration 1700, that using the anisotropic resampling in accordance with the present embodiments enables preservation of the sharp edges in a point cloud model.

[0078] To evaluate the resampling performance for mesh reconstruction in accordance with the present embodiments, different reconstruct methods are compared by triangle quality measurement and minimum angle statistics. The methods include Scale Space, Screened Poisson, Delaunay Strategy, Parallel CVT, and Gaussian kernel based optimization methods, and mesh reconstruction in accordance with the present embodiments. In TABLE III, the triangle quality measurement results of the different methods are compared for different models having different point cloud numbers.

Method Model (Point Number)	Scale Space		Screened Poisson		Delaunay Strategy		Parallel CVT		Gaussian kernel		AIVS	
	<i>Q_{min}</i>	<i>Q_{avg}</i>	<i>Q_{min}</i>	<i>Q_{avg}</i>	<i>Q_{min}</i>	<i>Q_{avg}</i>	<i>Q_{min}</i>	<i>Q_{avg}</i>	<i>Q_{min}</i>	<i>Q_{avg}</i>	<i>Q_{min}</i>	<i>Q_{avg}</i>
Bunny(10,000)	Nan	0.66	Nan	0.61	0.11	0.70	0.19	0.86	Nan	0.77	0.59	0.91
Horse(10,000)	Nan	0.67	Nan	0.61	0.12	0.69	0.20	0.86	Nan	0.72	0.62	0.91
Armadillo(10,000)	Nan	0.62	Nan	0.61	0.14	0.65	0.26	0.86	Nan	0.73	0.47	0.91
Dragon(20,000)	Nan	0.62	Nan	0.61	Nan	0.66	Nan	0.86	Nan	0.72	0.40	0.91
Asian Dragon(20,000)	Nan	0.56	Nan	0.61	0.11	0.66	Nan	0.85	Nan	0.75	0.19	0.91
Angel(10,000)	Nan	0.60	Nan	0.60	Nan	0.64	0.19	0.85	Nan	0.69	0.26	0.91
Buddha(20,000)	Nan	0.61	Nan	0.62	Nan	0.65	Nan	0.84	Nan	0.72	0.28	0.91
Lucy(20,000)	Nan	0.61	Nan	0.62	Nan	0.68	0.16	0.85	Nan	0.69	0.22	0.91
Thai Statue(20,000)	Nan	0.58	Nan	0.61	Nan	0.66	0.16	0.84	Nan	0.75	0.22	0.90
Ant(10,000)	Nan	0.65	Nan	0.61	0.13	0.71	0.45	0.88	Nan	0.74	0.56	0.91
Centuar(10,000)	Nan	0.65	Nan	0.61	0.11	0.72	0.42	0.87	Nan	0.71	0.59	0.91
Giraffe(10,000)	Nan	0.64	Nan	0.62	0.13	0.70	0.33	0.88	Nan	0.71	0.64	0.91
Hand(10,000)	Nan	0.68	Nan	0.61	Nan	0.72	0.38	0.88	Nan	0.78	0.57	0.91
Nunchakus(10,000)	Nan	0.64	Nan	0.62	0.18	0.70	0.32	0.87	Nan	0.74	0.18	0.90
WoodMan(10,000)	Nan	0.62	Nan	0.61	Nan	0.68	0.22	0.88	Nan	0.72	0.54	0.91
Block(10,000)	Nan	0.69	Nan	0.63	0.22	0.75	0.18	0.85	Nan	0.65	0.18	0.89
Fandisk(10,000)	Nan	0.68	Nan	0.65	0.27	0.76	0.38	0.87	Nan	0.72	0.17	0.89
Joint(10,000)	Nan	0.68	Nan	0.64	0.23	0.72	0.44	0.86	Nan	0.73	0.28	0.90

TABLE III
Comparisons of Triangle Quality Measurement by the Different Methods

[0079] According to the evaluation results, our method has good performance for mesh reconstruction. This can also be seen in the illustration 1800 of FIG. 18 which

visually compares the reconstructed meshes from different methods for three different models having a 10,000 point number.

[0080] In addition, minimum angle statistics for the different methods are compared using minimum angle statistic histograms. FIGs. 19A and 19B depict minimum angle statistic histograms 1900, 1950 of mesh reconstruction of two models using Scale Space-based mesh reconstruction optimization. FIGs. 20A and 20B depict minimum angle statistic histograms 2000, 2050 of mesh reconstruction of two models using Screened Poisson-based mesh reconstruction optimization. FIGs. 21A and 21B depict minimum angle statistic histograms 2100, 2150 of mesh reconstruction of two models using Delaunay Strategy-based mesh reconstruction optimization. FIGs. 22A and 22B depict minimum angle statistic histograms 2200, 2250 of mesh reconstruction of two models using Parallel CVT-based mesh reconstruction optimization. FIGs. 23A and 23B depict minimum angle statistic histograms 2300, 2350 of mesh reconstruction of two models using Gaussian kernel-based mesh reconstruction optimization. And FIGs. 24A and 24B depict minimum angle statistic histograms 2400, 2450 of mesh reconstruction of two models using AIVS-based mesh reconstruction optimization in accordance with the present embodiments. As seen from the histograms 2400, 2450, AIVS-based mesh reconstruction optimization in accordance with the present embodiments provides a tighter minimum angle distribution which is more consistent across both models as compared to the other methods.

[0081] The time complexity of AIVS-based resampling without parallel computation is $O(nm \log(m))$, n is the voxel box number, and m is the average point number in each voxel box. For each voxel box, the resampling time complexity can be regarded as the local FPS, which is equal to sorting a set of points by FPS. Using parallel computation in accordance with the present embodiments, the time complexity of AIVS

can be reduced to $O(8m \log(m))$. The neighborhood size is 8 for each voxel, which means the number of parallel computation steps is 8. In the framework in accordance with the present embodiments, OpenMP is used as the parallel computation structure. In TABLES IV and V, computation time reports of different methods are compared for different point clouds.

Methods	MLS	Hierarchy	Wlop	Laplace	AIVS
Bunny	192.42	6.06	207.51	217.91	2.86
Horse	301.69	7.65	285.56	316.51	4.71
Armadillo	2225.32	18.23	206.19	1152.36	14.45
Angel	3699.47	25.06	342.70	1882.06	21.20
Dragon	11112.92	42.88	325.82	2712.82	28.69
Buddha	16458.72	53.81	367.94	4321.31	31.26
Asian-Dragon	-----	359.01	4271.37	-----	60.23

TABLE IV
Time Cost by Different Methods
(Seconds for 10,000 Simplification Points)

Methods	MLS	Hierarchy	Wlop	Laplace	AIVS
Bunny	231.81	3.91	62.20	206.48	2.67
Horse	334.21	5.51	113.59	314.94	4.08
Armadillo	2150.22	22.61	349.19	1174.19	14.66
Angel	3521.07	30.29	460.70	1895.78	22.21
Dragon	10192.72	60.35	1384.23	2742.64	28.51
Buddha	15394.71	75.72	1875.78	4441.88	35.64
Asian-Dragon	-----	548.39	15878.43	-----	61.23

TABLE V
Time Cost by Different Methods
(Seconds for 0.1 Simplification Rate)

[0082] Referring to FIGs. 25A and 25B, graphs 2500, 2550 depict the statistics of time cost from the methods as shown in TABLES IV and V, respectively, where the number of points in an original point cloud is plotted along the x-axis and the time cost in seconds is plotted along the y-axis. It can be seen from the graphs 2500, 2550 that the computation time provided by the AVIS-based resampling in accordance with the present embodiments is less than the computation time for any of the conventional resampling methods.

[0083] Thus, it can be seen that the present embodiments provide methods for point cloud processing which advantageously delivers correct local region, uniform density, point number control, and geometric feature keeping and provide reduced geometric errors, increased triangle quality and reduced computational time as compared to conventional point cloud processing methods. In accordance with the present embodiments, point cloud processing including AIVS-based resampling enables applications for point cloud simplification and/or point cloud-based mesh reconstruction.

[0084] Following the popularity of 3D scanning devices, 3D point clouds are widely used to support commercial applications such as augmented reality and virtual reality systems, shape retrieval from a point cloud-based database, city scale point clouds visualization and reconstruction, and autopilot and unmanned aircraft control. The resampling method in accordance with the present embodiments improves the efficiency of point cloud usage. In addition, the simplification in accordance with the present embodiments reduces the cost of the storage and the mesh reconstruction in accordance with the present embodiments provides high quality meshes for accurate geometric analysis. The advantages of the methods in accordance with the present embodiments include simplicity of implementation without complicated optimization, customization/personalization of method performance by enabling defining different user-specified requirements, and time cost accelerated by parallel computation. The low computational complexity and reduced storage requirements enable the methods in accordance with the present embodiments to be integrated on portable devices such as mobile phone and tablet computer.

[0085] While exemplary embodiments have been presented in the foregoing detailed description of the present embodiments, it should be appreciated that a vast number of

variations exist. It should further be appreciated that the exemplary embodiments are only examples, and are not intended to limit the scope, applicability, operation, or configuration of the invention in any way. Rather, the foregoing detailed description will provide those skilled in the art with a convenient road map for implementing exemplary embodiments of the invention, it being understood that various changes may be made in the function and arrangement of steps and method of operation described in the exemplary embodiments without departing from the scope of the invention as set forth in the appended claims.

CLAIMS

What is claimed is:

1. A method for voxel-structure-based point cloud resampling comprising:
dividing a point cloud into a plurality of voxel boxes with adjacency, and wherein each of the plurality of voxel boxes defines a local region in a three-dimensional space;
and
point cloud resampling the plurality of points based on an approximate intrinsic voxel structure to generate a resampled point cloud, wherein the point cloud resampling is implemented differently in at least two of the plurality of voxel boxes.
2. The method in accordance with Claim 1, wherein the point cloud resampling implemented differently in at least two of the plurality of voxel boxes comprises implementing isotropic resampling to generate an isotropic point cloud comprising a number of points.
3. The method in accordance with Claim 2, wherein generating the isotropic point cloud comprises specifying a point number, the point number specifying the number of points of the isotropic point cloud.
4. The method in accordance with Claim 3, wherein generating the isotropic point cloud further comprises generating a globally uniform dense isotropic point cloud, wherein generating the globally uniform dense isotropic point cloud comprises defining each of the number of points of the isotropic point cloud to be equidistant in local regions of the three-dimensional space.

5. The method in accordance with any of Claims 2 to 4, wherein the point cloud resampling implemented differently in at least two of the plurality of voxel boxes further comprises implementing anisotropic resampling of ones of the plurality of points in a second one of the at least two of the plurality of voxel boxes, wherein implementing anisotropic resampling comprises maintaining local uniform density in local region of the at least two of the plurality of voxel boxes.

6. The method in accordance with any of the preceding claims, further comprising before point cloud resampling the step of classifying the plurality of points into categories based on a classification strategy, and wherein the point cloud resampling comprises point cloud resampling ones of the plurality of points at one of a plurality of different resampling rates.

7. The method in accordance with Claim 6, wherein classifying the plurality of points comprises classifying the plurality of points into the categories based on a user-specified classification strategy.

8. The method in accordance with either Claim 6 or Claim 7, wherein the plurality of different resampling rates are user defined.

9. The method in accordance with either Claim 6 or Claim 7, wherein the plurality of different resampling rates are determined in response to one of the categories into which the one of the plurality of points being resampled has been classified

10. The method in accordance with any of the preceding claims, further comprising before point cloud resampling the steps of:

pre-processing the point cloud; and

constructing the approximate intrinsic voxel structure of the plurality of points of the point cloud.

11. The method in accordance with Claim 10, wherein pre-processing the point cloud comprises:

denoising the plurality of points of the point cloud to generate a denoised plurality of points;

octree down-sampling of the denoised plurality of points to generate a down-sampled plurality of points; and

up-sampling the down-sampled plurality of points to generate an up-sampled plurality of points of the point cloud with certain point number and uniform density, wherein constructing the approximate intrinsic voxel structure comprises constructing the approximate intrinsic voxel structure of the up-sampled plurality of points of the point cloud.

12. The method in accordance with any of the preceding claims, wherein the point cloud resampling implemented differently in the at least two of the plurality of voxel boxes comprises parallelly processing the point cloud resampling in the at least two of the plurality of voxel boxes.

13. The method in accordance with any of the preceding claims, further comprising deleting redundant points of the resampled point cloud until the plurality of points within the resampled point cloud are equal to a user-specified number of points.

14. A method for point cloud simplification comprising the method for voxel-structure-based point cloud resampling of any of the Claims 1 to 13.

15. A method for voxel-structure-based mesh reconstruction comprising the method for voxel-structure-based point cloud resampling of any of the Claims 1 to 13.

16. The method in accordance with Claim 15, further comprising:
constructing a mesh based on the resampled point cloud by Delaunay triangulation; and
optimizing the mesh.

17. The method in accordance with Claim 16, wherein optimizing the mesh comprises internal edge rebuilding of the mesh and isotropic remeshing while maintaining geometric features of the mesh.