

THREE DIMENSIONAL DATA PROCESSING METHODS AND SYSTEMS

TECHNICAL FIELD

- 5 The present disclosure relates to three dimensional data processing methods and systems and in particular to remeshing of three dimensional data.

BACKGROUND

- 10 As one kind of 3D data representation, triangular meshes have been successfully used in many commercial applications such as the film industry, intelligent city management, advertising, automated manufacturing, medical treatment, etc. Benefiting from the excellent geometric and physical properties, the high quality meshes can be used to simulate the real-world objects. Compared to point clouds and depth maps, triangular
- 15 meshes retain complete topological structure with better 2-manifold properties. Triangular meshes support special geometric feature analysis such as heat flow computation, Ricci flow, optimal transportation, functional map, geodesic curve extraction, etc. However, to obtain high-quality meshes that meet the requirements still requires a lot of manual operations. Constructing the meshes is time-consuming
- 20 and laborious. In addition, some artificially designed meshes by computer aided design (CAD) software have lots of low-quality triangular faces which affect geometric feature extraction. To solve this problem, remeshing is proposed.

- Remeshing methods are used to improve the mesh quality. Some researchers design
- 25 remeshing frameworks to achieve high-quality triangular faces for mesh representation. Such frameworks can be divided into two technical routes: edge reconnection and neighborhood optimization. The edge reconnection frameworks attempt to reconnect edges of the input mesh to achieve remeshing result. Such frameworks can achieve better isotropic property in remeshing result without complex
- 30 optimization. However, the frameworks are sensitive to original vertex distribution. For some meshes with extremely distorted triangular faces, the robustness of the frameworks is not stable and reliable. The neighborhood optimization frameworks optimize the adjacent structure of different vertexes directly. It includes two categories: centroidal Voronoi tessellation (CVT) based and particle-based frameworks. Such

frameworks are robust to original vertex distribution. The drawbacks of the frameworks are huge computation cost for adjacent structure optimization and poor convergence. For user-specified geometric feature keeping, the performance of the frameworks is not stable. It is sensitive to the curvature changes in the original mesh.

5

As mentioned before, related works can be broadly classified into three categories: Edge reconnect-based, CVT-based, and Particle-based. Edge reconnect-based methods rebuild the adjacency of vertices to implement remeshing. Some basic operations are used to guide the reconnect processing, including split, collapse, flip and valence optimization. The advantages of such methods include: 1). without complex optimization; 2). Lower computation cost; 3). convenient to implement. However, for adaptively isotropic remeshing, the adaptive remeshing process is sensitive to the point distribution of original mesh. The quality of remeshing result is not stable in some regions with sharp curvature changes.

15

CVT-based methods use the Voronoi Diagram or its dual processing, Delaunay triangulation, to achieve the remeshing result. Such methods optimize the mesh structure by Voronoi cell programming. In summary, the CVT-based methods can achieve high quality remeshing result with Voronoi Diagram optimization. It is robust to original vertex distribution. The drawback of the CVT-based methods is that they are sensitive to regions with sharp curvature changes. Using the intrinsic control can reduce such influence. However, this solution reduces the computational efficiency inevitably.

20

Particle-based methods attempt to find a more direct remeshing method by vertex neighbor structure optimization. Comparing to CVT-based methods, practice-based methods do not compute the Voronoi Diagram and this improves the computation efficiency. Basically, such methods are widely used in dynamic remeshing task with higher performance. However, the methods are sensitive to sharp curvature and have higher probability to trap into local optimum.

25

30

SUMMARY

The present disclosure provides an adaptive isotropic remeshing method which is based on curvature smoothed field (CSF). The CSF is a continuous and smooth field assigned by the discrete curvature values. In CSF, the remeshing can be implemented by an edge reconnection strategy to balance curvature sensitive and isotropic properties.

According to a first aspect of the present disclosure a method of processing three dimensional mesh data is provided. The method comprises: receiving input three dimensional mesh data comprising a plurality of vertexes and edges joining the vertexes; calculating a set of initial curvature values, the set of initial curvature values comprising a curvature value for each vertex of mesh data corresponding to the input three dimensional mesh data; forming an initial curvature field from the set of initial curvature values; smoothing the initial curvature field to obtain a set of smoothed curvature values; calculating an average edge length for the input three dimensional mesh data; determining a multiple value for each vertex of the input three dimensional mesh data; iteratively reconnecting vertexes of the input three dimensional mesh data to generate reconnected three dimensional mesh data by: splitting edges with a length greater than a first threshold depending on the multiple value corresponding to a vertex associated with the respective edge and the average edge length; collapsing edges with a length less than a second threshold into a middle point, the second threshold depending on the multiple value corresponding to a vertex associated with the respective edge and the average edge length; flipping edges to implement valance optimization; and tangent smoothing edges; and outputting the reconnected three dimensional mesh data.

In an embodiment, the method further comprises: generating sub-remeshed input three dimensional mesh data by dividing edges of the input three dimensional mesh data having a length greater than a sub-remeshing threshold into two edges by adding additional vertexes, wherein the mesh data corresponding to the input three dimensional mesh data is the sub-remeshed input three dimensional mesh data.

In an embodiment, the method further comprises: computing an average global edge length of the input three dimensional mesh data; and determining the sub-remeshing threshold from average global edge length of the input three dimensional mesh data.

In an embodiment, the process of iteratively reconnecting vertexes of the input three dimensional mesh data to generated reconnected three dimensional mesh data is carried out for a fixed number of iterations.

5

In an embodiment, the process of iteratively reconnecting vertexes of the input three dimensional mesh data to generate reconnected three dimensional mesh data is carried out until the edge lengths are within a threshold range of the average edge length.

10

In an embodiment, the method further comprises: constructing a histogram corresponding to the statistical distribution of curvature values of the smoothed curvature field, wherein the multiple value for each vertex of the input three dimensional mesh data is determined from the histogram.

15

In an embodiment, constructing the histogram comprises dividing the curvature values into a plurality of groups according to a ratio.

In an embodiment, smoothing the initial curvature field to obtain a set of smoothed curvature values comprises applying a Laplacian operator by calculating cotangent weights.

20

In an embodiment, calculating the set of initial curvature values comprises calculating inverse cosine values are computed from neighbor points in a Voronoi cell and calculating the initial curvature values as the maximum inverse cosine value for a point.

25

According to a second aspect of the present disclosure a computer readable carrier medium carrying processor executable instructions which when executed on a processor cause the processor to carry out a method as set out above is provided.

30

According to a third aspect of the present disclosure a three dimensional data processing system for processing three dimensional mesh data is provided. The three dimensional data processing system comprising: a processor and a data storage device storing computer program instructions operable to cause the processor to:

receive input three dimensional mesh data comprising a plurality of vertexes and edges joining the vertexes; calculate a set of initial curvature values, the set of initial curvature values comprising a curvature value for each vertex of mesh data corresponding to the input three dimensional mesh data; form an initial curvature field
5 from the set of initial curvature values; smooth the initial curvature field to obtain a set of smoothed curvature values; calculate an average edge length for the input three dimensional mesh data; determine a multiple value for each vertex of the input three dimensional mesh data; iteratively reconnect vertexes of the input three dimensional mesh data to generate reconnected three dimensional mesh data by: splitting edges
10 with a length greater than a first threshold depending on the multiple value corresponding to a vertex associated with the respective edge and the average edge length; collapsing edges with a length less than a second threshold into a middle point, the second threshold depending on the multiple value corresponding to a vertex associated with the respective edge and the average edge length; flipping edges to
15 implement valance optimization; and tangent smoothing edges; and output the reconnected three dimensional mesh data.

In an embodiment, the data storage device further stores computer program instructions operable to cause the processor to: generate sub-remeshed input three
20 dimensional mesh data by dividing edges of the input three dimensional mesh data having a length greater than a sub-remeshing threshold into two edges by adding additional vertexes, wherein the mesh data corresponding to the input three dimensional mesh data is the sub-remeshed input three dimensional mesh data.

25 In an embodiment, the data storage device further stores computer program instructions operable to cause the processor to: compute an average global edge length of the input three dimensional mesh data; and determine the sub-remeshing threshold from average global edge length of the input three dimensional mesh data.

30 In an embodiment, the data storage device further stores computer program instructions operable to cause the processor to: carry out the process of iteratively reconnecting vertexes of the input three dimensional mesh data to generated reconnected three dimensional mesh data for a fixed number of iterations.

In an embodiment, the data storage device further stores computer program instructions operable to cause the processor to: carry out the process of iteratively reconnecting vertexes of the input three dimensional mesh data to generate reconnected three dimensional mesh data until the edge lengths are within a threshold
5 range of the average edge length.

In an embodiment, the data storage device further stores computer program instructions operable to cause the processor to: construct a histogram corresponding to the statistical distribution of curvature values of the smoothed curvature field,
10 wherein the multiple value for each vertex of the input three dimensional mesh data is determined from the histogram.

In an embodiment, the data storage device further stores computer program instructions operable to cause the processor to: construct the histogram by dividing
15 the curvature values into a plurality of groups according to a ratio.

In an embodiment, the data storage device further stores computer program instructions operable to cause the processor to: smooth the initial curvature field to obtain a set of smoothed curvature values by applying a Laplacian operator by
20 calculating cotangent weights.

In an embodiment, the data storage device further stores computer program instructions operable to cause the processor to: calculate the set of initial curvature values by calculating inverse cosine values computed from neighbor points in a
25 Voronoi cell and to calculate the initial curvature values as the maximum inverse cosine value for a point.

The present disclosure provides curvature smoothed field (CSF) for adaptively isotropic remeshing. The field smooths the curvature in the original mesh. It
30 establishes a continuous curvature distribution which reduces the distortion of edge length during the remeshing.

Based on the CSF, we design the histogram-based edge reconnection scheme to automatically optimize edge length in the remeshing. The scheme statistics the

curvature distribution in the CSF and optimizes edge lengths. It obtains the consistency between vertex density and curvature sensitive which balances geometric feature keeping and mesh quality improvement.

- 5 We provide a sub-remeshing as a pre-processing for the mesh with extremely distorted triangular faces. As a supplementary function, it improves the robustness and efficiency of our adaptively isotropic remeshing.

10 Our adaptively isotropic remeshing method can be classified into first category, while utilizing the CSF to control the edge length during the remeshing. It improves the robustness for different vertex distributions and keeps the advantages of edge reconnect-based methods. With a histogram-based edge reconnection scheme, our method achieves balance between geometric feature keeping and mesh quality improvement. It can also optimize handling models by sub-remeshing.

15

BRIEF DESCRIPTION OF THE DRAWINGS

In the following, embodiments of the present invention will be described as non-limiting examples with reference to the accompanying drawings in which:

20

FIG.1 illustrates an overview of a three dimensional data remeshing method according to an embodiment of the present invention;

25 FIG.2 is a block diagram showing a three dimensional data processing system according to an embodiment of the present invention;

FIG.3 is a flow chart showing a three dimensional data processing method according to an embodiment of the present invention;

30 FIG.4A and FIG.4B illustrate the process of sub-remeshing which is carried out in embodiments of the present invention;

FIG.5A to FIG.5C show an example of sub-remeshing and adaptive isotropic remeshing;

FIG.6 shows the angles used in calculation of a curvature smoothed field in embodiments of the present invention;

5 FIG.7A shows an example of an initial curvature scalar field and FIG.7B shows a corresponding curvature smoothed field;

FIG.8 shows an example of using a histogram based method for determination of a multiple value for each vertex of a mesh;

10

FIG.9 is a flow chart showing a method of iteratively reconnecting vertexes to form output mesh data used in embodiments of the present invention;

15 FIG.10 shows an example of the iterative reconnection process used in embodiments of the present invention;

FIG.11 and FIG.12 show a comparison of different remeshing methods; and

20 FIG.13 and FIG.14 are tables showing Hausdorff distances corresponding to the remeshing results shown in FIG.11 and FIG.12 respectively.

DETAILED DESCRIPTION

25 The present disclosure provides a curvature smoothed field-based adaptively isotropic remeshing method. It is to optimize an original mesh with the balance between geometric feature keeping and mesh quality improvement. To improve the robustness for various vertex distributions, we use the proposed curvature smoothed field (CSF) to control the edge reconnection. The CSF provides a smooth and stable scalar field based on curvature. Based on the CSF, a histogram-based edge reconnection
30 process is carried out to optimize the edge length of the original mesh. The process can balance geometric feature keeping and mesh quality improvement. Triangular faces which cross the border of regions with different curvatures can be controlled to avoid unstable face distortion while keeping the curvature sensitive. The point density

is proportional to curvature. For some meshes with extremely distorted triangular faces, a sub-remeshing for mesh pre-processing is used. The function can be regarded as a mesh subdivision for the distorted triangular faces. Combining these functions, the adaptively isotropic remeshing method can be used in various low-quality, artificially designed meshes.

FIG.1 illustrates an overview of a three dimensional data remeshing method according to an embodiment of the present invention. As shown in FIG.1, the method 100 takes input three dimensional data 110 as input. The input three dimensional data 110 indicates an input triangular mesh. In some embodiments, where the input three dimensional data indicates a distorted triangular mesh 112, the method 100 includes sub-remeshing 120 in which long mesh edges are divided and calculating a curvature smoothed field 130 on the mesh after sub-remeshing. In other embodiments, the method involves calculating the curvature smoothed field 130 from the original input mesh. The determination of whether to carry out the sub-remeshing may be based on determining whether any edges of the input mesh have a length which is greater than $2E_{ave}$ (average global edge length). If any edges have a length greater than $2E_{ave}$ then the sub-remeshing comprises dividing those edges by inserting points. Following the calculation of the curvature smoothed field 130, analysis 140 using the curvature smoothed field is carried out, and based on the analysis result, edge reconnection 150 carried out to provide an output reconnected mesh 152, 154.

FIG.2 shows a three dimensional data processing system according to an embodiment of the present invention. The three dimensional data processing system 200 is a computer system with memory that stores computer program modules which implement three dimensional data processing methods according to embodiments of the present invention.

The three dimensional data processing system 200 comprises a processor 210, a working memory 212, an input interface 214, an output interface 216, and program storage 220. The processor 210 may be implemented as one or more central processing unit (CPU) chips. The program storage 220 is a non-volatile storage device such as a hard disk drive which stores computer program modules. The computer program modules are loaded into the working memory 212 for execution by the

processor 210. The input interface 214 is an interface which allows data, three dimensional mesh data to be received by three dimensional data processing system 200. The input interface 214 may be a wireless network interface such as a Wi-Fi or Bluetooth interface, alternatively it may be a wired interface. The output interface 216
5 allows the three dimensional data processing system 200 to output reconnected three dimensional mesh data calculated as a result of the methods described in more detail below.

The program storage 220 stores a sub-remeshing module 222, a curvature calculation
10 module 224, a smoothed curvature field module 226, an analysis module 228 and an edge reconnection module 230. The computer program modules cause the processor 210 to execute various three dimensional data processing which is described in more detail below. The program storage 220 may be referred to in some contexts as computer readable storage media and/or non-transitory computer readable media. As
15 depicted in FIG.2, the computer program modules are distinct modules which perform respective functions implemented by the three dimensional data processing system 200. It will be appreciated that the boundaries between these modules are exemplary only, and that alternative embodiments may merge modules or impose an alternative decomposition of functionality of modules. For example, the modules discussed herein
20 may be decomposed into sub-modules to be executed as multiple computer processes, and, optionally, on multiple computers. Moreover, alternative embodiments may combine multiple instances of a particular module or sub-module. It will also be appreciated that, while a software implementation of the computer program modules is described herein, these may alternatively be implemented as one or more hardware
25 modules (such as field-programmable gate array(s) or application-specific integrated circuit(s)) comprising circuitry which implements equivalent functionality to that implemented in software.

In an embodiment, the functionality disclosed above may be provided by executing the
30 application and/or applications in a cloud computing environment. Cloud computing may comprise providing computing services via a network connection using dynamically scalable computing resources. A cloud computing environment may be established by an enterprise and/or may be hired on an as-needed basis from a third party provider.

FIG.3 is a flow chart showing a three dimensional data processing method according to an embodiment of the present invention. The method 300 shown in FIG.3 is carried out by the three dimensional data processing system shown in FIG.2

5

In step 302, the input interface 214 of the three dimensional data processing system 200 receives input three dimensional mesh data.

10

In step 303, the sub-remeshing module 222 is executed by the processor 210 of the three dimensional data processing system 200 to calculate a global average edge length $E_{\{ave\}}$. In some embodiments, it is determined whether any edges have a length greater than $2E_{\{ave\}}$ and if so, the method moves to step 304 in which sub-re-meshing is carried out. If no edges have a length greater than $2E_{\{ave\}}$, then the method moves to step 306.

15

In step 304, the sub-remeshing module 222 is executed by the processor 210 of the three dimensional data processing system 200 to carry out sub-remeshing to generate sub-remeshed input three dimensional mesh data.

20

As a pre-processing in adaptively isotropic remeshing, subremeshing is used to reduce long edges in the mesh. In some artificially designed meshes, there are a large number of the distorted triangular faces with uneven edge length distribution. Such faces affect the performance of adaptively isotropic remeshing and even produce incorrect remeshing result.

25

FIG.4A and FIG.4B illustrate the process of sub-remeshing which is carried out in embodiments of the present invention.

30

FIG.4A shows examples of triangles with uneven edge length distribution. The core problem produced by such distorted triangular faces is that the adjacent structure cannot be kept by the edges with abnormal length (like edge ac in the examples shown in FIG.4A).

To solve the problem, the edges with long length should be divided by sub-remeshing. Some instances are shown in FIG.4B.

Edge with long length means that it has a large difference in length to its adjacent edges (ab to ac in the first picture and ab to bc in FIG.4B). The subremeshing divides the edges and reconnect the new edges in different triangular faces. The sub-remeshing scheme includes three steps: 1). computing the global average edge length $E_{\{ave\}}$; 2). scanning all edges of input mesh, and inserting new points to divide the edge which is larger than $2E_{\{ave\}}$; 3). Reconnecting the new points in different triangular faces. The $E_{\{ave\}}$ is to control the edge length in distorted triangular faces. Benefiting from new points inserted into an edge with large length, the edges with long length can be reduced to be within an acceptable range. After sub-remeshing, the original distorted triangular faces are divided into a new set of triangular faces without extremely long length. Then the purpose of sub-remeshing is achieved.

FIG.5A to FIG.5C show an example of sub-remeshing and adaptive isotropic remeshing. FIG.5A shows the original input mesh, FIG.5B shows the sub-remeshing result and FIG.5C shows the adaptive isotropic remeshing result.

As shown in FIG.5B, in the sub-remeshing result the triangular faces are cut into small ones and long edges are divided.

Returning now to FIG.3, in step 306, the curvature calculation module 224 is executed by the processor 210 of the three dimensional data processing system 200 to calculate initial curvature values.

In step 308, the curvature calculation module 224 is executed by the processor 210 of the three dimensional data processing system 200 to form an initial curvature field.

The initial curvature field is calculated by computing a curvature value at each vertex, which forms the initial discrete curvature field S . Specifically, for vertex x_i we define a normal-based curvature value to be:

$$s(x_i) = \max_{x_j \in R(x_i)} \{\arccos\langle N(x_i) \cdot N(x_j) \rangle\}$$

where $s(x_i)$ represents the curvature value obtained from the maximum of inverse cosine values, and the inverse cosine values are computed from x_i to its neighbor points x_j in Voronoi cell $R(x_i)$. This curvature intuitively represents how much the surface bends in the local neighborhood of a vertex.

In step 310, the smoother curvature field module 226 is executed by the processor 210 of the three dimensional data processing system 200 to smooth the initial curvature field and thereby generate a smoothed curvature field.

The initial curvature field S is smoothed to generate a smooth scalar field for the mesh, which defines the relationships between each vertex to its neighbors. The curvature smoothed field (CSF) provides a smooth scalar field for the mesh and defines relationships between each vertex to its neighbors by cotangent weights. The computation can be regarded as the Laplace operator

$$\Delta s_i = \sum_{j \in N_i} \left(\frac{w_{ij}}{w_i} \cdot s_j \right) - s_i, w_i = \sum_{j \in N_i} w_{ij}$$

where S is a scalar field which is provided to assign a value to each vertex. The parameters s_i and s_j are assigned to vertexes x_i and x_j . The N_i represents the neighbor set of vertex i . The Δs_i represents the update value for s_i . Using the smooth processing, a discrete scalar field is smoothed to be the CSF. The cotangent weights are computed by

$$w_{ij} = (\cot \alpha + \cot \beta) / |x_i - x_j|$$

where x_i and x_j are vertexes in a mesh. The weight w_{ij} of the two vertexes are defined by the distance and related cotangent values. The angles α and β are shown in FIG.6.

FIG.6 shows the angles used in calculation of a curvature smoothed field in embodiments of the present invention.

The smooth processing is used to update the value of S . During enough iterations of the processing, Δs_i for each vertex is approach to zero which likes harmonic state. Then, the CSF is achieved. Such state is shown in the Equation below. The ΔS represents the smooth processing update vector combined by Δs . The Equation means that the scalar field S is smoothed:

$$\Delta S = 0$$

The geometric feature keeping task is based on the CSF. We compute the curvature values to be the initial discrete field S . We define a normal-based curvature value to assign the S shown as

$$s(p_i) = \max\{\cos^{-1}\langle N(p_i) \cdot N(p_j) \rangle, p_j \in R(p_i) \}$$

where $s(p_i)$ represents curvature value of point p_i achieved from the maximum value of the arc cosine values. The arc cosine values are computed from the p_i to its neighbor points in Voronoi cell $R(p_i)$. The point p_j belongs to $R(p_i)$. The defined curvature intuitively reflects the bending degree in the point. In practice, there is a special case should be noticed. The value of cotangent weight may be negative. It produces unstable values during the smooth processing. An example of this is shown in FIG.5C. To avoid such cases, we control the range of cotangent weight, $w_{ij} \in [0.1, 10]$. We also set the control rate λ to balance the smooth degree. The related update process for S is shown as

$$s'_i = s_i + (1 - \lambda) \cdot \Delta s_i$$

where s'_i is the new value of s_i after one iteration of the update process.

FIG.7A shows an example of an initial curvature scalar field and FIG.7B shows a corresponding curvature smoothed field.

In FIG.7A and FIG.7B, we compare the original curvature scalar field and the CSF to explain the effect of our smooth processing. The default value of λ is set to 0.5 which averages the curvatures during iterate. It is clear that after smooth processing, the curvature-based scalar field is changed to be smoother and more continuous.

Returning again to FIG.3, in step 312, the analysis module 228 is executed by the processor 210 of the three dimensional data processing system to calculate an average edge length.

In step 312, the average triangle edge length l of the mesh is calculated.

In step 314, the analysis module 228 is executed by the processor 210 of the three dimensional data processing system to determine a multiple value for each vertex.

In step 314, the corresponding multiple value m to l for each point according to curvature values processed by CSF is assigned.

We provide the histogram-based curvature analysis to generate m . The corresponding is shown as

$$m(p_i) = \begin{cases} m_g, & m_g \in \{m_1, \dots, m_{base}, \dots, m_k\} \\ & s(p_i) \in [s_{g-1}, s_g] \end{cases}$$

where $m(p_i)$ is the m value for point p_i . It is decided by the curvature value $s(p_i)$. The curvature values of a point cloud are classified into k groups, each group corresponds to a value of $\{m\}$. To establish the m value for each point, we design a histogram-based curvature analysis for all points to generate statistical result.

FIG.8 shows an example of using a histogram based method for determination of a multiple value for each vertex of a mesh.

Firstly, we compute the histogram for curvature from all points and select the base value (the curvature with the maximum value in the histogram, the bold black line in FIG.8), which is corresponding to m_{base} . Next, we divide k groups according to the base value. The points with larger curvature value than the base one are divided into
5 $k/2$ groups and each group have same point number, same to the points with smaller curvature value. Finally, we collect the curvature ranges $\{s_g\}$ from the k groups.

According to the relationship between inner angles and edge lengths in a triangle, the ratio of the longer edge and shorter one should be less than $\sqrt{2}:1$ to reduce the
10 number of obtuse triangles. For the middle group of k with the base value, $m_{base} = 1$, which is consistent to the statistical result. Following the mentioned ratio, we set other values in $\{m\} = \{1.8, 1.4, 1, 0.8, 0.6\}$ which achieves balance between geometric feature keeping and mesh quality improvement. To avoid the extreme longer or shorter edge than then base one, the value of k is set to 5 by default.

15 In the example shown in FIG.8, $\{m_1, \dots, m_5\} = \{1.8, 1.4, 1.0, 0.8, 0.6\}$, X axis: curvature; Y axis: point number. The points belong to m_1 , m_2 and left half of m_3 are equal and same to other side.

20 Returning again to FIG.3, in step 316, the edge reconnection module 230 is executed by the processor 210 of the three dimensional data processing system 200 to iteratively reconnect vertexes to generate reconnected three dimensional mesh data. The iterative process of reconnecting vertexes is shown in FIG.9

25 FIG.9 is a flow chart showing a method of iteratively reconnecting vertexes to form output mesh data used in embodiments of the present invention. The method 900 shown in FIG.9 is carried out by the edge reconnection module 230 is executed by the processor 210 of the three dimensional data processing system 200.

30 In step 902, triangle edges are split if its length is larger than $\frac{5}{3}l \times m$ (one edge have two vertexes, may correspond to different values for m , choose the smaller one).

In step 904, triangle edges are collapsed into the middle point if its length is smaller than $\frac{4}{5}l \times m$ (choose the larger m for two vertexes).

In step 906, triangle edges are flipped to implement valence optimization (If the flip generates obtuse angles or the flip triangle edge includes two external edge points, the flip should not be processed).

In step 908, if the point is not the external edge point, smooth point to center of the neighbor points on tangent space.

The iterative process may end once a specified number of iterations has been process (for example five), alternatively, the iterative process may end once the edge lengths are not changed according to a range defined by specified average length.

FIG.10 shows an example of the iterative reconnection process used in embodiments of the present invention. FIG.10 shows the input mesh, the mesh following splitting, the mesh following collapse, the mesh following flipping and finally the mesh following tangent smoothing. The steps form a basic iteration of the edge reconnection. The iteration includes split, collapse, flip, and tangent smoothing. Prior to the iteration, the parameter m should be defined for each point. It influences the edge length in the reconnection. When m corresponding to each point is equal, the adaptively isotropic remeshing degenerates to isotropic one. To achieve the reasonable representation of curvature changes, the m should correspond to the curvature distribution.

Returning finally to FIG.3, in step 318, the reconnected three dimensional mesh is output by the output interface of the three dimensional data processing system 200.

FIG.11 and FIG.12 show a comparison of different remeshing methods. We show the performance of our adaptively isotropic remeshing in this section. Two indexes are used to estimate the quality of remeshing, including geometric consistency evaluation and mesh quality measurement.

Hausdorff distance can be utilized to measure the geometric consistency. The quality of remeshing result can be evaluated by triangle quality measurement $Q(t) \in [0,1]$. Based on the indexes, we provide a quantitative analysis for different remeshing methods, including anisotropic remeshing, centroidal Voronoi tessellation (CVT), particle-based resampling, and Delaunay-GPU.

For geometric consistency evaluation, we specify a certain range (9,000~13,000 points) of point number in remeshing results. Different methods should output the mesh with points in the range, which ensures the fairness and reasonable of the measurement. In FIG.11 and FIG.12, we show remeshing results by different methods.

FIG.13 and FIG.14 are tables showing Hausdorff distances corresponding to the remeshing results shown in FIG.11 and FIG.12 respectively.

According to the distances, our method obtains better geometric consistency with similar points. For mesh quality measurement, We compute average Q_{ave} for different remeshing results. Since there is no smooth field like CSF of our method, the anisotropic method produces error edges in regions with sharp curvature changes. For some models, the remeshing process of anisotropic may crash. For CVT and Delaunay-GPU remeshing methods, Voronoi cells are optimized and the mesh quality is improved theoretically. However, such methods cannot achieve the stable result in some mesh models with sharp curvature changes and extreme obtuse triangles. Such situations produce errors for Voronoi cell optimization in local tangent space. Our method provides a feasible solution. It achieves balance between isotropic property and geometric feature keeping.

3D meshes are widely used to support the commercial applications such as AR/VR systems, shape retrieval from a 3D model database, city scale visualization and reconstruction, autopilot, unmanned aircraft control, and human body mesh based activity recognition. Our remeshing method improves the quality of the mesh which saves labor costs and improve efficiency. The advantages include simple to implement without complicated optimization, robust to various meshes, and balance isotropic property and curvature sensitive.

Whilst the foregoing description has described exemplary embodiments, it will be understood by those skilled in the art that many variations of the embodiments can be made within the scope and spirit of the present invention.

CLAIMS

1. A method of processing three dimensional mesh data, the method comprising:
 - receiving input three dimensional mesh data comprising a plurality of vertexes and edges joining the vertexes;
 - calculating a set of initial curvature values, the set of initial curvature values comprising a curvature value for each vertex of mesh data corresponding to the input three dimensional mesh data;
 - forming an initial curvature field from the set of initial curvature values;
 - smoothing the initial curvature field to obtain a set of smoothed curvature values;
 - calculating an average edge length for the input three dimensional mesh data;
 - determining a multiple value for each vertex of the input three dimensional mesh data;
 - iteratively reconnecting vertexes of the input three dimensional mesh data to generate reconnected three dimensional mesh data by:
 - splitting edges with a length greater than a first threshold depending on the multiple value corresponding to a vertex associated with the respective edge and the average edge length;
 - collapsing edges with a length less than a second threshold into a middle point, the second threshold depending on the multiple value corresponding to a vertex associated with the respective edge and the average edge length;
 - flipping edges to implement valance optimization; and
 - tangent smoothing edges; and
 - outputting the reconnected three dimensional mesh data.
2. The method according to claim 1, further comprising:
 - generating sub-remeshed input three dimensional mesh data by dividing edges of the input three dimensional mesh data having a length greater than a sub-remeshing threshold into two edges by adding additional vertexes, wherein the mesh data corresponding to the input three dimensional mesh data is the sub-remeshed input three dimensional mesh data.

3. The method according to claim 2, further comprising:
computing an average global edge length of the input three dimensional mesh data; and
determining the sub-remeshing threshold from average global edge length of the input three dimensional mesh data.
4. The method according to any preceding claim, wherein the process of iteratively reconnecting vertexes of the input three dimensional mesh data to generated reconnected three dimensional mesh data is carried out for a fixed number of iterations.
5. The method according to any one of claims 1 to 3, wherein the process of iteratively reconnecting vertexes of the input three dimensional mesh data to generate reconnected three dimensional mesh data is carried out until the edge lengths are within a threshold range of the average edge length.
6. The method according to any preceding claim, further comprising:
constructing a histogram corresponding to the statistical distribution of curvature values of the smoothed curvature field, wherein the multiple value for each vertex of the input three dimensional mesh data is determined from the histogram.
7. The method according to claim 6, wherein constructing the histogram comprises dividing the curvature values into a plurality of groups according to a ratio.
8. The method according to any preceding claim, wherein smoothing the initial curvature field to obtain a set of smoothed curvature values comprises applying a Laplacian operator by calculating cotangent weights.
9. The method according to any preceding claim, wherein calculating the set of initial curvature values comprises calculating inverse cosine values are computed from neighbor points in a Voronoi cell and calculating the initial curvature values as the maximum inverse cosine value for a point.

10. A computer readable carrier medium carrying processor executable instructions which when executed on a processor cause the processor to carry out a method according to any one of claims 1 to 9.

11. A three dimensional data processing system for processing three dimensional mesh data, the three dimensional data processing system comprising: a processor and a data storage device storing computer program instructions operable to cause the processor to:

- receive input three dimensional mesh data comprising a plurality of vertexes and edges joining the vertexes;

- calculate a set of initial curvature values, the set of initial curvature values comprising a curvature value for each vertex of mesh data corresponding to the input three dimensional mesh data;

- form an initial curvature field from the set of initial curvature values;

- smooth the initial curvature field to obtain a set of smoothed curvature values;

- calculate an average edge length for the input three dimensional mesh data;

- determine a multiple value for each vertex of the input three dimensional mesh data;

- iteratively reconnect vertexes of the input three dimensional mesh data to generate reconnected three dimensional mesh data by:

- splitting edges with a length greater than a first threshold

- depending on the multiple value corresponding to a vertex associated with the respective edge and the average edge length;

- collapsing edges with a length less than a second threshold into a middle point, the second threshold depending on the multiple value corresponding to a vertex associated with the respective edge and the average edge length;

- flipping edges to implement valance optimization; and

- tangent smoothing edges; and

- output the reconnected three dimensional mesh data.

12. The three dimensional data processing system according to claim 11, wherein the data storage device further stores computer program instructions operable to cause the processor to: generate sub-remeshed input three dimensional mesh data

by dividing edges of the input three dimensional mesh data having a length greater than a sub-remeshing threshold into two edges by adding additional vertexes, wherein the mesh data corresponding to the input three dimensional mesh data is the sub-remeshed input three dimensional mesh data.

13. The three dimensional data processing system according to claim 12, wherein the data storage device further stores computer program instructions operable to cause the processor to:

compute an average global edge length of the input three dimensional mesh data; and

determine the sub-remeshing threshold from average global edge length of the input three dimensional mesh data.

14. The three dimensional data processing system according to any one of claims 11 to 13, wherein the data storage device further stores computer program instructions operable to cause the processor to: carry out the process of iteratively reconnecting vertexes of the input three dimensional mesh data to generated reconnected three dimensional mesh data for a fixed number of iterations.

15. The three dimensional data processing system according to any one of claims 11 to 13, wherein the data storage device further stores computer program instructions operable to cause the processor to: carry out the process of iteratively reconnecting vertexes of the input three dimensional mesh data to generate reconnected three dimensional mesh data until the edge lengths are within a threshold range of the average edge length.

16. The three dimensional data processing system according to any one of claims 11 to 15, wherein the data storage device further stores computer program instructions operable to cause the processor to:

construct a histogram corresponding to the statistical distribution of curvature values of the smoothed curvature field, wherein the multiple value for each vertex of the input three dimensional mesh data is determined from the histogram.

17. The three dimensional data processing system according to any one of claims 11 to 16, wherein the data storage device further stores computer program instructions operable to cause the processor to: construct the histogram by dividing the curvature values into a plurality of groups according to a ratio.

18. The three dimensional data processing system according to any one of claims 11 to 17, wherein the data storage device further stores computer program instructions operable to cause the processor to: smooth the initial curvature field to obtain a set of smoothed curvature values by applying a Laplacian operator by calculating cotangent weights.

19. The three dimensional data processing system according to any one of claims 11 to 18, wherein the data storage device further stores computer program instructions operable to cause the processor to: calculate the set of initial curvature values by calculating inverse cosine values computed from neighbor points in a Voronoi cell and to calculate the initial curvature values as the maximum inverse cosine value for a point.